An acceleration procedure for optimal first-order methods

Michel Baes*, Michael Bürgisser†

July 16, 2012

Abstract

We introduce in this paper an optimal first-order method that allows an easy and cheap evaluation of the local Lipschitz constant of the objective's gradient. This constant must ideally be chosen at every iteration as small as possible, while serving in an indispensable upper bound for the value of the objective function. In the previously existing variants of optimal first-order methods, this upper bound inequality was constructed from points computed during the current iteration. It was thus not possible to select the optimal value for this Lipschitz constant at the beginning of the iteration.

In our variant, the upper bound inequality is constructed from points available before the current iteration, offering us the possibility to set the Lipschitz constant to its optimal value at once. This procedure, even if efficient in practice, presents a higher worse-case complexity than standard optimal first-order methods. We propose an alternative strategy that retains the practical efficiency of this procedure, while having an optimal worse-case complexity. We show how our generic scheme can be adapted for smoothing techniques, and perform numerical experiments on large scale eigenvalue minimization problems. As compared with standard optimal first-order methods, our schemes allows us to divide computation times by two to three orders of magnitude for the largest problems we considered.

Keywords: Convex Optimization, First-Order Methods, Eigenvalue Optimization.

1 Introduction

With a few notable exceptions [GG05], first-order methods constitute the main family of algorithms able to deal with very large-scale convex optimization problems [Peñ08, Nes10, RT11, Nes12]. Among them, optimal first-order methods play a distinguished role: they are practically as cheap as a first-order method can be, with a complexity per iteration growing as a moderate polynomial of the problem's size, while the worst-case number of iterations they require is provably optimal for *smooth* instances [NY83, Nes83]. Their scope of applicability is restricted to optimization problems with a differentiable convex objective function f, whose gradient is globally Lipschitz continuous. Nevertheless, Nesterov introduced a systematic procedure, applicable to many nonsmooth convex

*Corr. author. Institute for Operations Research, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland, michel.baes@ifor.math.ethz.ch.

†Institute for Operations Research, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland, michael.buergisser@ifor.math.ethz.ch.

functions, for building a smooth approximation to which one can apply an optimal first-order minimization algorithm citenesterov:coreDP12/2003. His construction can easily be specified to realize the optimal compromise between the smoothness of the substitute objective function and how accurately it approximates the original objective. Smoothing techniques extended dramatically the scope of optimal first-order methods, and many variants of the original scheme developed in [Nes83] have been studied since then (see e.g. [Tse08, BT09, d'A08, LLM]).

Critically, optimal first-order methods need an estimation of the corresponding Lipschitz constant with respect to an appropriate norm. Originally, this bound is used to build an approximation of the epigraph of the objective function. The larger the bound, the worse this approximation is, and the more steps the method is likely to take. Some strategies have been proposed to re-actualize at every step this bound [Nes07a, BCG11]. These strategies are based on the fact that the Lipschitz constant is used at a particular iteration to satisfy a single inequality rather than as a global property. If this inequality is verified, it suffices to reduce this constant, redo the iteration with the new value, and recheck the inequality, until it is no longer satisfied. If the inequality is not verified, we simply multiply the constant by an appropriate value and re-perform the iteration as long as the inequality does not hold. This strategy yielded a significant increase in practical efficiency. However, the cost of a single iteration has to be multiplied by a number ranging between two and possibly a few dozen.

We show in this paper how a slight modification of these methods allows us to choose inexpensively the smallest possible approximation that guarantees the global convergence of the method. In particular, we avoid redoing several times the work needed for one iteration. The practical effect of such a procedure is appreciable, and is documented at the end of this paper. On the theoretical side, we show that our re-evaluation of the Lipschitz constant, if applied systematically, gives an algorithm which requires at worse $\mathcal{O}((LD)/\epsilon)$ iterations, where L is the global Lipschitz of the objective's gradient, D measures the diameter of the feasible set, and $\epsilon > 0$ is the desired absolute accuracy on the objective's value. In comparison with the vanilla optimal first-order method, which has a complexity of $\mathcal{O}(\sqrt{(LD)/\epsilon})$ iterations, this algorithm is clearly worse. We propose a mixed strategy that presents simultaneously the practical efficiency of our systematic method for very large- scale problems and, up to a constant that we can take as close to 1 as desired, the theoretical efficiency of optimal methods.

When we apply to smoothing techniques, our mixed strategy suggests a different choice of the smoothness parameter than the standard one. This fact should not be too surprising: as our method is precisely designed to fit appropriate local estimates of the gradient's Lipschitz constant, it allows us to be slightly sloppier in our request for global smoothness.

In order to validate our general scheme, we consider a well-known application of smoothing techniques to the problem of minimizing the largest eigenvalue of a convex combination of given symmetric matrices [Nes07b]. This problem has many applications, and a large variety of methods have been devised to solve it [HR00, AK07], some of which are adaptations of optimal first-order methods [NJLS09, BBN11, dK12]. To the best of our knowledge, these methods improve the complexity of each step of optimal first-order methods, but are not trying to decrease the number of these steps. With respect to original smoothing techniques, our method allows us to *divide* by hundreds the practical number of iterations for large-scale instances, that is, when we have 100 matrices of size larger than 200. Our methods even allowed us to deal with a problem involving 10% sparse matrices of dimension 12,800 within 9 hours, while standard optimal first-order methods

would have taken more than one year if it were to perform all the iterations predicted by the worse-case analysis. It appears in practice that the standard optimal method needs about two third of these iterations: about eight months would be needed to solve that problem.

The paper is organized as follows. We outline our method in Section 2. First, we analyze its complexity for smooth convex problems and particularize our result to the two variants mentioned above. Then, we describe how the algorithm and its variants can be particularized to smoothed problems. In Section 3, we apply these methods to the eigenvalue minimization problem and present some numerical experiments. We have relegated the rather technical proof of the main theorem of the paper in the appendix.

2 An accelerated optimal first-order method

In this section, we introduce an accelerated version of Nesterov's optimal first-order method that is presented in [Nes05] and discuss its application in smoothing techniques.

2.1 General algorithm

We start by considering the following optimization problem:

$$f^* = \min_{x \in Q} f(x), \tag{1}$$

where Q is a closed and convex subset of $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ is a function, which is supposed to attain its minimum on the set Q. In addition, we assume that f is convex and differentiable with a Lipschitz continuous gradient on Q.

We consider $\mathbb{R}^n$ with the standard Euclidean scalar product, which is denoted by $\langle \cdot, \cdot \rangle$. The space $\mathbb{R}^n$ is equipped with a norm $\|\cdot\|_{\mathbb{R}^n}$, which may differ from the norm that is induced by the scalar product. We write $\|\cdot\|_{\mathbb{R}^n,*}$ for the dual norm to $\|\cdot\|_{\mathbb{R}^n}$:

$$\|u\|_{\mathbb{R}^n,*} := \max_{x \in \mathbb{R}^n} \left\{ \langle u, x \rangle : \|x\|_{\mathbb{R}^n} = 1 \right\}, \qquad u \in \mathbb{R}^n.$$

As f has a Lipschitz continuous gradient on Q, there exists a constant $L = L(Q) > 0$ which satisfies the inequality:

$$\|\nabla f(x) - \nabla f(y)\|_{\mathbb{R}^n,*} \leq L \|x - y\|_{\mathbb{R}^n} \qquad \forall\, x, y \in Q. \tag{2}$$

Nesterov developed a first-order method (see Equations (5.6) in [Nes05]) that allows us to compute approximate solutions to Problem (1). This optimal first-order method has a convergence rate of $\mathcal{O}\left(L/T^2\right)$, which outperforms the rate of convergence of common subgradient methods by two orders of magnitude. We quickly recall that common subgradient methods converge with the order $\mathcal{O}(1/T^{0.5})$; see for instance [NY83].

At every step of Nesterov's optimal first-order method, the Lipschitz constant L is used to update the iterates; see [Nes05] for the details. However, the constant L is a global parameter of the function f, as L needs to satisfy Condition (2) on the whole set Q. In this subsection, we

introduce a refined version of Nesterov's optimal first-order method, where we replace the global parameter L by local estimates.

This algorithm requires the following basic notions. We say that $d_Q : Q \to \mathbb{R}_{\geq 0}$ is a *distance-generating function for the set* Q if it complies with the following requirements:

1. d_Q is continuous on Q;
2. d_Q is strongly convex with modulus 1 on Q:
$$d_Q(\lambda x + [1-\lambda]y) + \frac{\lambda[1-\lambda]}{2}\|x-y\|^2_{\mathbb{R}^n} \leq \lambda d_Q(x) + [1-\lambda]d_Q(y) \qquad \forall\, x, y \in Q;$$
3. given the set $Q^o(d_Q) := \{x \in Q : \partial d_Q(x) \neq \emptyset\}$, the subdifferential ∂d_Q gives rise to a continuous selection d'_Q on the set Q^o. If there is no possibility for confusion, we write Q^o instead of $Q^o(d_Q)$.

Let d_Q be a distance-generating function for the set Q and choose $z \in Q^o$. We write
$$V_z^{d_Q}(x) = d_Q(x) - d_Q(z) - \left\langle d'_Q(z), x - z\right\rangle \in \mathbb{R}_{\geq 0}$$
for the *Bregman distance of* $x \in Q$ *with respect to* $z \in Q^o$. Nesterov's optimal first-order method and its accelerated version that we present in this paper utilize a *prox-mapping*, that is, a mapping of the form:
$$\mathrm{Prox}_{Q,z}^{d_Q} : \mathbb{R}^n \to Q^o : s \mapsto \arg\min_{x \in Q} \left\{\langle s, x - z\rangle + V_z^{d_Q}(x)\right\}, \qquad z \in Q^o. \tag{3}$$
If there is no possibility for confusion, we abbreviate $V_z^{d_Q}$ and $\mathrm{Prox}_{Q,z}^{d_Q}$ into V_z and $\mathrm{Prox}_{Q,z}$, respectively. Given $s \in \mathbb{R}^n$ and $z \in Q^o$, the value $\mathrm{Prox}_{Q,z}(s)$ can be rewritten as
$$\mathrm{Prox}_{Q,z}(s) = \arg\min_{x \in Q} \left\{\left\langle s - d'_Q(z), x\right\rangle + d_Q(x)\right\}.$$
It can be easily verified that this optimization problem has indeed a unique minimizer (Note that the objective function $x \mapsto \left\langle s - d'_Q(z), x\right\rangle + d_Q(x)$ is continuous and strongly convex. It remains to apply Lemma 6 from [Nes09].) and that this minimizer belongs to Q^o. For the reminder of this paper, we assume that this minimizer can be computed easily (Ideally, we can write it in a closed form.). The unique element
$$c(d_Q) := \arg\min_{x \in Q} \{d_Q(x)\} \in Q^o$$
is called the d_Q-center (Note that $c(d_Q) = \mathrm{Prox}_{Q,z}(d'_Q(z))$ for any $z \in Q^o$.). Without loss of generality, we may assume that d_Q vanishes at the point $c(d_Q)$. Then, Lemma 6 in [Nes09] can be used to justify the following inequality:
$$d_Q(x) \geq \frac{1}{2}\|x - c(d_Q)\|^2_{\mathbb{R}^n} \qquad \forall\, x \in Q. \tag{4}$$

We discuss now the analytical complexity of the accelerated optimal first-order method displayed in Algorithm 1. We choose $T \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and assume that the sequences $(x_t)_{t=0}^{T+1}$, $(u_t)_{t=0}^{T+1}$,

Algorithm 1 Accelerated optimal first-order method

Choose $T \in \mathbb{N}_0$.
Choose $(\gamma_t)_{t=0}^{T+1}$ with $\gamma_0 \in (0,1]$, $\gamma_t \geq 0$, and $\gamma_t^2 \leq \Gamma_t := \sum_{k=0}^{t} \gamma_k$ for any $0 \leq t \leq T+1$.
Set $L_0 = L$ and $x_0 = c(d_Q)$.
Compute $u_0 := \arg\min_{x \in Q} \{\gamma_0 (f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle) + L_0 d_Q(x)\}$.
Set $z_0 = u_0$, $\tau_0 = \gamma_1/\Gamma_1$, and $x_1 = \tau_0 z_0 + (1-\tau_0)u_0 = z_0$.
Define $\hat{x}_1 := \mathrm{Prox}_{Q,z}(\gamma_1 \nabla f(x_1)/L_0)$.
Set $u_1 = \tau_0 \hat{x}_1 + (1-\tau_0)u_0$.
for $1 \leq t \leq T$ **do**
  Choose $0 < L_t \leq L$ such that:

$$f(u_t) \leq f(x_t) + \langle \nabla f(x_t), u_t - x_t \rangle + \frac{L_t}{2} \|u_t - x_t\|_{\mathbb{R}^n}^2 . \tag{5}$$

  Set $z_t = \arg\min_{x \in Q} \left\{ \sum_{k=0}^{t} \gamma_k (f(x_k) + \langle \nabla f(x_k), x - x_k \rangle) + L_t d_Q(x) \right\}$.
  Set $\tau_t = \gamma_{t+1}/\Gamma_{t+1}$ and $x_{t+1} = \tau_t z_t + (1-\tau_t)u_t$.
  Compute $\hat{x}_{t+1} := \mathrm{Prox}_{Q,z_t}(\gamma_{t+1} \nabla f(x_{t+1})/L_t)$.
  Set $u_{t+1} = \tau_t \hat{x}_{t+1} + (1-\tau_t)u_t$.
end for

$(z_t)_{t=0}^{T}$, $(\hat{x}_t)_{t=1}^{T+1}$, $(\gamma_t)_{t=0}^{T+1}$, $(\Gamma_t)_{t=0}^{T+1}$, $(\tau_t)_{t=0}^{T}$, and $(L_t)_{t=0}^{T}$ are generated by Algorithm 1. Given $0 \leq t \leq T$, we say that *Inequality* $(\mathcal{I}_t)$ *holds* if

$$\Gamma_t f(u_t) + \sum_{k=0}^{t-1} (L_{k+1} - L_k) \left(d_Q(z_{k+1}) - \frac{1}{2} \|z_k - \hat{x}_{k+1}\|_{\mathbb{R}^n}^2 \right) \leq \psi_t, \tag{$\mathcal{I}_t$}$$

where

$$\psi_t := \min_{x \in Q} \left\{ \sum_{k=0}^{t} \gamma_k (f(x_k) + \langle \nabla f(x_k), x - x_k \rangle) + L_t d_Q(x) \right\}.$$

As the proof of the following result is rather long and technical, we give it in the Appendix A.

Theorem 2.1 *Inequality* $(\mathcal{I}_t)$ *holds for any* $0 \leq t \leq T$.

For the reminder of this subsection, we refer to $x^* \in Q$ as an optimal solution to the optimization problem $f^* = \min_{x \in Q} f(x)$.

Theorem 2.2 *For any* $T \in \mathbb{N}_0$*, we have:*

$$f(u_T) - f^* \leq \frac{1}{\Gamma_T} \left[L_T d_Q(x^*) + \sum_{t=0}^{T-1} (L_t - L_{t+1}) \left(d_Q(z_{t+1}) - \frac{1}{2} \|z_t - \hat{x}_{t+1}\|_{\mathbb{R}^n}^2 \right) \right].$$

Proof: Let $0 \leq t \leq T$. The convexity of the function f and the definition of Γ_t imply

$$\psi_t := \min_{x \in Q} \left\{ L_t d_Q(x) + \sum_{k=0}^{t} \gamma_k (f(x_k) + \langle \nabla f(x_k), x - x_k \rangle) \right\}$$

$$
\begin{aligned}
&\leq L_t d_Q(x^*) + \sum_{k=0}^{t} \gamma_k \left(f(x_k) + \langle \nabla f(x_k), x^* - x_k \rangle \right) \\
&\leq L_t d_Q(x^*) + \sum_{k=0}^{t} \gamma_k f(x^*) \\
&= L_t d_Q(x^*) + \Gamma_t f(x^*).
\end{aligned}
$$

It remains to combine this inequality with Theorem 2.1. ■

Nesterov [Nes05] suggests to choose the sequence $(\gamma_t)_{t=0}^{T+1}$ as

$$
\gamma_t := \frac{t+1}{2} \qquad \forall\ 0 \leq t \leq T+1. \tag{6}
$$

Lemma 2 of [Nes05] shows that we have the following equations for this choice of the sequence $(\gamma_t)_{t=0}^{T+1}$:

$$
\tau_t = \frac{2}{t+3} \qquad \forall\ 0 \leq t \leq T
$$

and

$$
\Gamma_t = \frac{(t+1)(t+2)}{4}, \qquad \gamma_t^2 \leq \Gamma_t \qquad \forall\ 0 \leq t \leq T+1.
$$

As an immediate consequence of Theorem 2.2, we obtain the following result for our accelerated optimal first-order method.

Corollary 2.1 *Let us choose the sequence $(\gamma_t)_{t=0}^{T+1}$ in Algorithm 1 as described in (6). Then, we have for any $T \in \mathbb{N}_0$:*

$$
f(u_T) - f^* \leq \frac{4 L_T d_Q(x^*)}{(T+1)(T+2)} + \sum_{t=0}^{T-1} \frac{4\left(L_t - L_{t+1}\right)}{(T+1)(T+2)} \left(d_Q(z_{t+1}) - \frac{1}{2} \left\| z_t - \hat{x}_{t+1} \right\|_{\mathbb{R}^n}^2 \right). \tag{7}
$$

■

There exist different strategies for updating the sequence $(L_t)_{t=0}^{T}$ in Algorithm 1. When $\alpha := 0$, we recover the complexity results of Nesterov's optimal first-order method (see e.g. Subsection 5.3 of [Nes05]), for which Inequality (7) can be rewritten as $f(u_T) - f^* \leq \frac{4 L d_Q(x^*)}{(T+1)(T+2)}$.

Alternative 1: (most aggressive adaptive setting) Fix $0 < \kappa \ll 1$ and let $1 \leq t \leq T$. The most aggressive choice for the constant L_t corresponds to

$$
L_t := \max\left\{ \bar{L}_t, \kappa L \right\} \in [\kappa L, L], \qquad \bar{L}_t := \frac{2\left[f(u_t) - f(x_t) - \langle \nabla f(x_t), u_t - x_t \rangle \right]}{\left\| u_t - x_t \right\|_{\mathbb{R}^n}^2} \leq L. \tag{8}
$$

The computation of the constant L_t requires the entities u_t, x_t, and $\nabla f(x_t)$. In sharp contrast with the methods proposed so far [Nes07a, BCG11], all these entities are known from the previous step $t-1$, implying that the constant L_t can be determined immediately.

Independent of the choice of the L_t's, we can always derive the following trivial convergence result for Algorithm 1 from Inequality (7):

$$
f(u_T) - f^* \leq \frac{4 L \sup_{x \in Q} d_Q(x)}{(T+1)(T+2)} + \frac{20 L T \sup_{x \in Q} d_Q(x)}{(T+1)(T+2)} \leq \frac{20 L \sup_{x \in Q} d_Q(x)}{T+2},
$$

as $L_T d(x^*) \leq L \sup_{x \in Q} d_Q(x)$ and

$$\begin{aligned}\sum_{t=0}^{T-1} (L_t - L_{t+1}) \left(d_Q(z_{t+1}) - \frac{1}{2} \|z_t - \hat{x}_{t+1}\|^2_{\mathbb{R}^n} \right) &\leq \sum_{t=0}^{T-1} |L_t - L_{t+1}| \left(d_Q(z_{t+1}) + \frac{1}{2} \|z_t - \hat{x}_{t+1}\|^2_{\mathbb{R}^n} \right) \\ &\leq \sum_{t=0}^{T-1} L \left(\sup_{x \in Q} d_Q(x) + 4 \sup_{x \in Q} d_Q(x) \right) \\ &= 5LT \sup_{x \in Q} d_Q(x).\end{aligned}$$

Note that the last inequality holds due to (4).

Thus, Algorithm 1 equipped with the most aggressive update strategy, which is described in (8), needs at most

$$T = \left\lceil 20L \sup_{x \in Q} d(x)/\epsilon - 2 \right\rceil$$

iterations to find a feasible ϵ-solution, provided that $\sup_{x \in Q} d(x)$ is finite.

Alternative 2: (hybrid setting) Finally, we can combine the two settings that are presented above. We choose a number $\alpha \geq 0$ and denote by $1 \leq t \leq T$ the current iteration. As long as

$$\sum_{k=0}^{\bar{t}-1} (L_k - L_{k+1}) \left(d_Q(z_{k+1}) - \frac{1}{2} \|z_k - \hat{x}_{k+1}\|^2_{\mathbb{R}^n} \right) \leq \alpha L d_Q(x^*) \qquad \forall\ 1 \leq \bar{t} \leq t, \tag{9}$$

we use the update strategy that is described in (8). When Condition (9) is not satisfied for the first time, we set $L_{\bar{t}} := L$ for any $\bar{t} \geq t$ and recompute the point z_t.

With the just specified setting, Inequality (7) results in the bound

$$f(u_T) - f^* \leq \frac{4(1+\alpha) L d_Q(x^*)}{(T+1)(T+2)}.$$

That is, we need to perform at most

$$T = \left\lceil 2\sqrt{(1+\alpha) L d_Q(x^*)/\epsilon} - 1 \right\rceil$$

iterations of Algorithm 1 to find a point $x \in Q$ with $f(x) - f^* \leq \epsilon$, where $\epsilon > 0$. This complexity result deviates by a factor of $(1+\alpha)^{0.5}$ from the efficiency estimate of the non-adaptive method. With $\alpha = 5(T+1) \sup_{x \in Q} d(x) - 1$, the setting coincides with Alternative 1.

2.2 The accelerated optimal first-order method in smoothing techniques

Smoothing techniques [Nes05] constitute a two-stage procedure that can be applied to non-smooth optimization problems with a very particular structure. In a first step, a smooth approximation of the non-smooth objective function is formed, so that Nesterov's optimal first-order method can be applied afterwards. In this section, we study the effects of replacing Nesterov's original optimal first-order method by its accelerated version in smoothing techniques.

We assume that the sets $Q_1 \subset \mathbb{R}^n$ and $Q_2 \subset \mathbb{R}^m$ are both compact and convex. In addition, we endow the spaces $\mathbb{R}^n$ and $\mathbb{R}^m$ with two (maybe different) norms. We denote by $\|\cdot\|_{\mathbb{R}^n}$ and $\|\cdot\|_{\mathbb{R}^m}$ the norm of the spaces $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. Nesterov considers convex optimization problems of the form:

$$\min_{x \in Q_1} \max_{y \in Q_2} \phi(x, y), \qquad \phi(x, y) := f_1(x) + \langle \mathcal{A}(x), y \rangle - f_2(y), \tag{10}$$

where $f_1 : \mathbb{R}^n \to \mathbb{R}$, $f_2 : \mathbb{R}^m \to \mathbb{R}$ are smooth and convex, and $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator. With a slight abuse of notation, we write $\langle \cdot, \cdot \rangle$ for the Euclidean scalar product in both spaces $\mathbb{R}^n$ and $\mathbb{R}^m$.

According to the standard MiniMax Theorem in Convex Analysis (see Corollary 37.3.2 in [Roc70]), we have, due to the compactness and convexity of the sets Q_1 and Q_2, the following pair of primal-dual convex optimization problems:

$$\min_{x \in Q_1} \left\{ \overline{\phi}(x) := \max_{y \in Q_2} \phi(x, y) \right\} = \max_{y \in Q_2} \left\{ \underline{\phi}(y) := \min_{x \in Q_1} \phi(x, y) \right\}.$$

The operator $\mathcal{A}$ comes with an adjoint operator $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{R}^n$, which is defined by the relation:

$$\langle \mathcal{A}(x), y \rangle = \langle x, \mathcal{A}^*(y) \rangle \qquad \forall\ (x, y) \in \mathbb{R}^n \times \mathbb{R}^m.$$

The analysis of Nesterov's smoothing techniques requires a norm of the operator $\mathcal{A}$. This norm is constructed as follows:

$$\|\mathcal{A}\|_{\mathbb{R}^n, \mathbb{R}^m} := \max_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} \left\{ \langle \mathcal{A}(x), y \rangle : \|x\|_{\mathbb{R}^n} = 1,\ \|y\|_{\mathbb{R}^m} = 1 \right\}.$$

We are ready to form a smooth approximation of $\overline{\phi}$ to which we can apply Algorithm 1. We choose a distance-generating function $d_{Q_2} : Q_2 \to \mathbb{R}_{\geq 0}$ for the set Q_2 and consider the auxiliary function

$$\overline{\phi}_\mu : \mathbb{R}^n \to \mathbb{R} : x \mapsto \max_{y \in Q_2} \left\{ f_1(x) + \langle \mathcal{A}(x), y \rangle - f_2(y) - \mu d_{Q_2}(y) \right\},$$

where $\mu > 0$ is a positive smoothness parameter. This function defines a uniform approximation of $\overline{\phi}$, as

$$\overline{\phi}_\mu(x) \leq \overline{\phi}(x) \leq \overline{\phi}_\mu(x) + \mu \max_{z \in Q_2} d_{Q_2}(z) \qquad \forall\ x \in Q_1; \tag{11}$$

see Inequality (2.7) in [Nes05]. The function $y \mapsto \langle \mathcal{A}(x), y \rangle - f_2(y) - \mu d_{Q_2}(y)$ is strongly concave for any $x \in Q_1$, as the distance-generating function d_{Q_2} is strongly convex by its definition. Hence, the function $y \mapsto \langle \mathcal{A}(x), y \rangle - f_2(y) - \mu d_{Q_2}(y)$ has a unique maximizer on Q_2. We denote this maximizer by $y_*(x)$.

Nesterov showed that $\overline{\phi}_\mu$ is differentiable with a Lipschitz continuous gradient. We write $M > 0$ for the Lipschitz constant of the gradient of f_1.

Theorem 2.3 (Theorem 1 in [Nes05]) *The function $\overline{\phi}_\mu$ is well-defined, continuously differentiable, and convex on $\mathbb{R}^n$. The gradient of $\overline{\phi}_\mu$ takes the form*

$$\nabla \overline{\phi}_\mu(x) = \nabla f_1(x) + \mathcal{A}^*(y_*(x)),$$

and is Lipschitz continuous with the constant $L_\mu := M + \|\mathcal{A}\|^2_{\mathbb{R}^n, \mathbb{R}^m} / \mu$.

Algorithm 2 Algorithm 1 (with $\gamma_t = (t+1)/2$) applied to Problem (12)

Choose $T \in \mathbb{N}_0$.
Choose a smoothness parameter $\mu > 0$ and a distance-generating function $d_{Q_1} : Q_1 \to \mathbb{R}$ for the set Q_1.
Set $L_0 = L_\mu = M + \|\mathcal{A}\|^2_{\mathbb{R}^n,\mathbb{R}^m}/\mu$ and $x_0 = c(d_{Q_1})$.
Set $u_0 = \arg\min_{x\in Q_1} \left\{\frac{1}{2}\left(\overline{\phi}_\mu(x_0) + \langle \nabla\overline{\phi}_\mu(x_0), x - x_0\rangle\right) + L_0 d_{Q_1}(x)\right\}$.
Set $z_0 = u_0$, $\tau_0 = \frac{2}{3}$, and $x_1 = \tau_0 z_0 + (1-\tau_0)u_0 = z_0$.
Define $\hat{x}_1 := \mathrm{Prox}_{Q_1,z}\left(\nabla\overline{\phi}_\mu(x_1)/L_0\right)$.
Set $u_1 = \tau_0\hat{x}_1 + (1-\tau_0)u_0$.
for $1 \le t \le T$ **do**
  Choose $0 < L_t \le L_\mu$ such that:

$$\overline{\phi}_\mu(u_t) \le \overline{\phi}_\mu(x_t) + \left\langle \nabla\overline{\phi}_\mu(x_t), u_t - x_t\right\rangle + \frac{L_t}{2}\|u_t - x_t\|^2_{\mathbb{R}^n}.$$

  Set

$$z_t = \arg\min_{x\in Q_1}\left\{\sum_{k=0}^{t}\frac{k+1}{2}\left(\overline{\phi}_\mu(x_k) + \langle\nabla\overline{\phi}_\mu(x_k), x - x_k\rangle\right) + L_t d_{Q_1}(x)\right\}.$$

  Set $\tau_t = \frac{2}{t+3}$ and $x_{t+1} = \tau_t z_t + (1-\tau_t)u_t$.
  Compute $\hat{x}_{t+1} := \mathrm{Prox}_{Q_1,z_t}\left(\frac{t+2}{2}\nabla\overline{\phi}_\mu(x_{t+1})/L_t\right)$.
  Set $u_{t+1} = \tau_t\hat{x}_{t+1} + (1-\tau_t)u_t$.
end for

As an immediate consequence, we can apply Algorithm 1 to the problem:

$$\min_{x\in Q_1}\overline{\phi}_\mu(x). \tag{12}$$

Algorithm 2 corresponds to Algorithm 1 when we apply this method with step-sizes as described in (6) to Problem (12). A slight adaptation of the proof of Theorem 3 in [Nes05] yields to the following result, for which we need the definitions:

$$D_1 := \max_{x\in Q_1} d_{Q_1}(x) \qquad \text{and} \qquad D_2 := \max_{y\in Q_2} d_{Q_2}(y).$$

Theorem 2.4 *Fix $T \in \mathbb{N}_0$ and assume that the sequences $(x_t)_{t=0}^{T+1}$, $(u_t)_{t=0}^{T+1}$, $(z_t)_{t=0}^{T}$, $(\hat{x}_t)_{t=1}^{T+1}$, and $(L_t)_{t=0}^{T}$ are generated by Algorithm 2 with the smoothness parameter $\mu > 0$. For*

$$\bar{x} := u_T \in Q_1 \qquad \textit{and} \qquad \bar{y} := \sum_{t=0}^{T}\frac{2(t+1)}{(T+1)(T+2)}y_*(x_t) \in Q_2,$$

we have:

$$\overline{\phi}(\bar{x}) - \underline{\phi}(\bar{y}) \le \frac{4\left(D_1\|\mathcal{A}\|^2_{\mathbb{R}^n,\mathbb{R}^m}/\mu + D_1 M - \chi_T\right)}{(T+1)^2} + \mu D_2, \tag{13}$$

where

$$\chi_T := \sum_{t=0}^{T-1} (L_{t+1} - L_t) \left(d_{Q_1}(z_{t+1}) - \frac{1}{2} \|z_t - \hat{x}_{t+1}\|^2_{\mathbb{R}^n} \right).$$

For remainder of this section, we use the notations of Algorithm 1 and Theorem 2.4.

Proof: In accordance to Theorem 2.1 and to the step-size choice (6), we have the inequality:

$$\overline{\phi}_\mu(\bar{x}) = \overline{\phi}_\mu(u_T) \leq \frac{4(L_T D_1 - \chi_T)}{(T+1)(T+2)} + \min_{x \in Q_1} \frac{2\beta_T(x)}{(T+1)(T+2)}, \tag{14}$$

where

$$\beta_T(x) := \sum_{t=0}^{T} (t+1) \left(\overline{\phi}_\mu(x_t) + \left\langle \nabla \overline{\phi}_\mu(x_t), x - x_t \right\rangle \right) \qquad \forall\, x \in Q_1.$$

Let $x \in Q_1$. Using Theorem 2.3 and the convexity of f_1 and f_2, we can write:

$$\begin{aligned}
\beta_T(x) &= \sum_{t=0}^{T} (t+1) \left(f_1(x) + \langle \mathcal{A}(x), y_*(x_t) \rangle - f_2(y_*(x_t)) - \mu d_{Q_2}(y_*(x_t)) \right) \\
&\leq \sum_{t=0}^{T} (t+1) \left(f_1(x) - \langle \mathcal{A}(x), y_*(x_t) \rangle - f_2(y_*(x_t)) \right) \\
&\leq \frac{(T+1)(T+2)}{2} \left(f_1(x) + \langle \mathcal{A}(x), \bar{y} \rangle - f(\bar{y}) \right).
\end{aligned}$$

The above inequality implies:

$$\min_{x \in Q_1} \beta_T(x) \leq \frac{(T+1)(T+2)}{2} \underline{\phi}(\bar{y}). \tag{15}$$

Recall that we have $L_T \leq L_\mu = \|\mathcal{A}\|^2_{\mathbb{R}^n, \mathbb{R}^m} / \mu + M$ by construction. We use Inequalities (14), (15), and (11) to justify the following inequalities:

$$\frac{4(D_1 \|\mathcal{A}\|^2_{\mathbb{R}^n, \mathbb{R}^m} / \mu - \chi_T)}{(T+1)^2} \geq \frac{4(L_T D_1 - \chi_T)}{(T+1)(T+2)} \geq \overline{\phi}_\mu(\bar{x}) - \underline{\phi}(\bar{y}) \geq \overline{\phi}(\bar{x}) - \underline{\phi}(\bar{y}) - \mu D_2.$$

■

We conclude this section by discussing different strategies for choosing the sequence $(L_t)_{t=0}^T$ and the smoothness parameter μ.

Alternative 1: (most aggressive adaptive setting) We can always give the following upper bound for the quantity $(-\chi_T)$ in Theorem 2.4:

$$-\chi_T \leq 5 L_\mu D_1 T = 5 D_1 T \left(\|\mathcal{A}\|^2_{\mathbb{R}^n, \mathbb{R}^m} / \mu + M \right),$$

which allows us to reformulate (13) as

$$\overline{\phi}(\bar{x}) - \underline{\phi}(\bar{y}) \leq \frac{20 D_1 \left(\|\mathcal{A}\|^2_{\mathbb{R}^n, \mathbb{R}^m} / \mu + M \right)}{T+1} + \mu D_2.$$

Minimizing the right-hand side of the above inequality with respect to μ, that is, setting μ to

$$\mu_2^* := 2\,\|\mathcal{A}\|_{\mathbb{R}^n,\mathbb{R}^m}\sqrt{\frac{5D_1}{(T+1)D_2}},$$

we obtain:

$$\overline{\phi}(\bar{x}) - \underline{\phi}(\bar{y}) \leq 4\,\|\mathcal{A}\|_{\mathbb{R}^n,\mathbb{R}^m}\sqrt{\frac{5D_1D_2}{T+1}} + \frac{20D_1M}{T+1}.$$

As this bound is independent of the choice the L_t's, it is valid also for the most aggressive setting, that is, for

$$L_t := \max\left\{\bar{L}_t, \kappa L_\mu\right\} \in [\kappa L_\mu, L_\mu], \qquad \bar{L}_t := \frac{2\left[\overline{\phi}(u_t) - \overline{\phi}(x_t) - \left\langle \nabla\overline{\phi}(x_t), u_t - x_t\right\rangle\right]}{\|u_t - x_t\|^2_{\mathbb{R}^n}} \leq L_\mu, \quad (16)$$

where $1 \leq t \leq T$ and $0 < \kappa \ll 1$ is fixed.

Alternative 2: (hybrid setting) Let $\alpha \geq 0$. We follow the setting described in (16) for all $1 \leq t \leq T$ as long as

$$-\chi_{\bar{t}} \leq \alpha D_1\left(\|\mathcal{A}\|^2_{\mathbb{R}^n,\mathbb{R}^m}/\mu + M\right)$$

is satisfied for any $1 \leq \bar{t} \leq t$. When this condition fails for the first time, say for $t = t'$, we set L_t to L_μ for any $t \geq t'$ and recompute the point $z_{t'}$. In this hybrid setting, Inequality (13) yields to

$$\overline{\phi}(\bar{x}) - \underline{\phi}(\bar{y}) \leq \frac{4(1+\alpha)D_1\left(\|\mathcal{A}\|^2_{\mathbb{R}^n,\mathbb{R}^m}/\mu + M\right)}{(T+1)^2} + \mu D_2.$$

We choose μ such that the right-hand side of the above inequality is minimized, that is, we fix μ to

$$\mu_3^* := \frac{2\,\|\mathcal{A}\|_{\mathbb{R}^n,\mathbb{R}^m}}{T+1}\sqrt{\frac{(1+\alpha)D_1}{D_2}},$$

and end up with the following bound:

$$\overline{\phi}(\bar{x}) - \underline{\phi}(\bar{y}) \leq \frac{4\,\|\mathcal{A}\|_{\mathbb{R}^n,\mathbb{R}^m}\sqrt{(1+\alpha)D_1D_2}}{T+1} + \frac{4(1+\alpha)D_1M}{(T+1)^2}.$$

Note that Alternative 2 coincides with Alternative 1 if $\alpha = 5(T+1) - 1$.

3 An application in large-scale eigenvalue optimization

In this section, we study the practical behavior of accelerated smoothing techniques. We apply them to the problem of finding a convex combination of given symmetric matrices such that the maximal eigenvalue of the resulting matrix is minimal.

3.1 Problem description

Let

$$\Delta_m := \left\{ x \in \mathbb{R}^m_{\geq 0} : \sum_{j=1}^m x_j = 1 \right\} \subset \mathbb{R}^m$$

be the $(m-1)$-dimensional probability simplex. Denoting by $\mathcal{S}_n$ the space of symmetric real $(n \times n)$-matrices, we write $Y \succeq 0$ if $Y \in \mathcal{S}_n$ is positive semidefinite and $\mathrm{Tr}(Y) := \sum_{i=1}^n Y_{ii}$ for the trace of Y. We refer to

$$\Delta_n^M := \{Y \succeq 0 : \mathrm{Tr}(Y) = 1\} \subset \mathcal{S}_n$$

as the simplex in matrix form. Finally, we denote by

$$\lambda_n(Y) \geq \ldots \geq \lambda_1(Y)$$

the eigenvalues of the symmetric matrix $Y \succeq 0$ and assume that they are ordered decreasingly. Throughout this section, we consider the following problem:

$$\min_{x \in \Delta_m} \lambda_n \left(\sum_{j=1}^m x_j A_j \right) = \min_{x \in \Delta_m} \left\{ \overline{\phi}(x) := \max_{Y \in \Delta_n^M} \sum_{j=1}^m x_j \langle A_j, Y \rangle_F \right\}, \tag{17}$$

where $A_1, \ldots, A_m \in \mathcal{S}_n$ and $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius scalar product.

3.2 Applying accelerated smoothing techniques

3.2.1 Smoothing the objective function

We equip $\mathcal{S}_n$ with the induced 1-norm, that is, with $\|Y\|_{(1)} := \sum_{i=1}^n |\lambda_i(Y)|$, where $Y \in \mathcal{S}_n$. The dual norm corresponds to the induced ∞-norm, that is, to the norm $\|W\|_{(\infty)} := \max_{1 \leq i \leq n} |\lambda_i(W)|$ with $W \in \mathcal{S}_n$. We choose

$$d_{\Delta_n^M}(Y) := \ln(n) + \sum_{i=1}^n \lambda_i(Y) \ln(\lambda_i(Y)), \qquad Y \in \Delta_n^M,$$

as distance-generating function for the set Δ_n^M, for which we have $d_{\Delta_n^M}(Y) \leq \ln(n)$ for any $Y \in \Delta_n^M$; see for instance [Nes07b] for a proof that $d_{\Delta_n^M}$ is a distance-generating function for Δ_n^M. We obtain the following smooth objective function as an approximation to $\overline{\phi}$:

$$\overline{\phi}_\mu(x) := \max_{Y \in \Delta_n^M} \left\{ \sum_{j=1}^m x_j \langle A_j, Y \rangle_F - \mu d_{\Delta_n^M}(Y) \right\} = \mu \ln \left(\sum_{i=1}^n \exp \left[\lambda_i \left(\frac{\sum_{j=1}^m x_j A_j}{\mu} \right) \right] \right) - \mu \ln(n),$$

where $x \in \Delta_m$ and $\mu > 0$ denotes the smoothness parameter. The approximation quality depends on the smoothness parameter:

$$\overline{\phi}_\mu(x) \leq \overline{\phi}(x) \leq \overline{\phi}_\mu(x) + \mu \ln(n) \qquad \forall\, x \in \Delta_m.$$

Finally, the gradient of $\overline{\phi}_\mu$ is given by

$$\left[\nabla\overline{\phi}_\mu(x)\right]_j = \langle A_j, Y_*(x)\rangle_F \qquad \forall\ 1 \le j \le m,$$

where $x \in \Delta_m$ and $Y_*(x)$ denotes the unique maximizer of $Y \mapsto \sum_{j=1}^n x_j \langle A_j, Y\rangle_F - \mu d_{\Delta_n^M}(Y)$ over Δ_n^M. Theorem 2.3 implies that the gradient is Lipschitz continuous with a Lipschitz constant of $L_\mu := \max_{1\le j\le m} \|A_j\|_{(\infty)} / \mu$.

3.2.2 Applying the accelerated optimal first-order method with hybrid setting

Let the space $\mathbb{R}^m$ be equipped with the 1-norm. We use

$$d_{\Delta_m}(x) := \ln(m) + \sum_{j=1}^m x_j \ln(x_j), \qquad x \in \Delta_m,$$

as distance-generating function for the set Δ_m. Note that $d_{\Delta_m}(x) \le \ln(m)$ for any $x \in \Delta_m$.

We run Algorithm 2 with the hybrid setting that is described in Alternative 2 in Section 2.2. Let us fix the accuracy $\epsilon > 0$ and the parameter $\alpha \ge 0$ that defines when to switch back to the non-adaptive setting. The smoothness parameter is set as follows:

$$\mu := \frac{\epsilon}{2\ln(n)}.$$

Note that the smoothness parameter does not depend on α. According to Theorem 2.4, we need to perform at most

$$T = \left\lceil \frac{4\max_{1\le j\le m}\|A_j\|_{(\infty)}\sqrt{(1+\alpha)\ln(m)\ln(n)}}{\epsilon} - 1 \right\rceil \tag{18}$$

iterations of Algorithm 2 in order to find a tuple $(\bar{x}, \bar{Y}) \in \Delta_m \times \Delta_n^M$ such that

$$\max_{Y\in\Delta_n^M} \sum_{j=1}^m \bar{x}_j \langle A_j, Y\rangle_F - \min_{x\in\Delta_m} \sum_{j=1}^m x_j \left\langle A_j, \bar{Y}\right\rangle_F \le \epsilon. \tag{19}$$

3.3 Numerical results

We consider randomly generated instances of Problem (10), where we fix m to 100 and where the symmetric $(n \times n)$-matrices $A_1, \ldots, A_m$ have a joint sparsity structure, each of them with about $n^2/10$ non-zero entries. We approximate the parameter

$$\mathcal{L} := \max_{1\le j\le m} \|A_j\|_{(\infty)}$$

by applying the Power method to the matrices A_j and taking the maximum, which we denote by $\mathcal{L}'$, of the computed values afterwards. We solve the randomly generated instances of Problem (10) up to a relative accuracy of $\epsilon\mathcal{L}'$ with $\epsilon := 0.002$.

All numerical results that we present in this section are averaged over ten runs and obtained on a computer with 24 processors, each of them with 2.67 GHz, and with 96 GB of RAM. The methods are implemented in Matlab (version R2012a). Matrix exponentials are computed through the Matlab built-in function `expm()`.

Average CPU time [sec]

n	100	200	400	800
Original smoothing techniques	139	366	1′406	5′961
Accelerated smoothing techniques	116	3	9	32
Acceleration	16.55%	99.18%	99.36%	99.46%

Average # of iterations that are required in practice

n	100	200	400	800
Original smoothing techniques	6′180	6′690	7′150	7′520
Accelerated smoothing techniques	4′918	18	14	13
Reduction	20.42%	99.73%	99.80%	99.83%

Average # of iterations that are required in theory

n	100	200	400	800
Original smoothing techniques	9′210	9′879	10′505	11′096
Accelerated smoothing techniques	18′420	19′758	21′011	22′193
Reduction	−100.00%	−100.00%	−100.01%	−100.01%

Table 1: Average CPU time and number of iterations (in practice and in theory) that are required by original and accelerated smoothing techniques for finding an approximate solution to randomly generated instances of Problem (10) (with fixed accuracy $0.002\mathcal{L}'$ and with $m = 100$).

3.3.1 Comparing the practical behavior of different methods

In Table 1, we present numerical results for the following two methods:

- ⋄ Original smoothing techniques: This implementation corresponds to Algorithm 2 with constant $L_t = L_\mu$ for any $0 \leq t \leq T$. That is, we set $\alpha = 0$ in Alternative 2 in Section 2.2.
- ⋄ Accelerated smoothing techniques: We equip Algorithm 2 with the hybrid setting described in Alternative 2 in Section 2.2, where we choose $\alpha := 3$ and $\kappa := 10^{-12}$. With this setting, we need to perform twice as many iterations as with original smoothing techniques with respect to the worst-case bounds; see (18).

For both methods, we check the duality gap (19) at every 100-th iteration. Additionally for the later method, we also verify this condition at every of the first hundred iterations. The maximal eigenvalue that corresponds to the first term in (19) is computed through the Matlab built-in functions `max()` and `eig()`.

We observe that accelerated smoothing techniques require significantly less CPU time and iterations in practice than original smoothing techniques; see Table 1. For problems involving matrices of size 200×200 up to size 800×800, we can reduce the number of iterations in practice and the CPU time by more than 99%. Interestingly, the number of iterations that are required by accelerated smoothing techniques in practice is even decaying when the matrix size n is getting larger.

Note that there exists a gap in the average CPU time and number of iterations that are required by accelerated smoothing techniques in practice for solving the instances of size 100×100 and the

instances of size 200×200. In Figure 1, we plot the values

$$\beta_t := \frac{-\chi_t}{\ln(m) L_0} = \frac{-\sum_{t'=0}^{t-1} (L_{t'+1} - L_{t'}) \left(d_{\Delta_m}(z_{t'+1}) - \frac{1}{2} \|z_{t'} - \hat{x}_{t'+1}\|_1^2 \right)}{D_1 L_0} \qquad \forall\, t \geq 1. \tag{20}$$

In contrast to the cases $n = 200$, $n = 400$, or $n = 800$, where these values remain small (that is, below 0.25), we have considerably large values β_t for $n = 100$. However, the values are still below 3, as we switch back to a non-adaptive setting as soon as β_t would be larger than 3. This behavior is in full accordance with the gap mentioned in the beginning of this paragraph. The non-smooth patterns at the end of the plots in Figure 1 are due to the averaging over the different runs (We may need a different number of iterations in the different runs.).

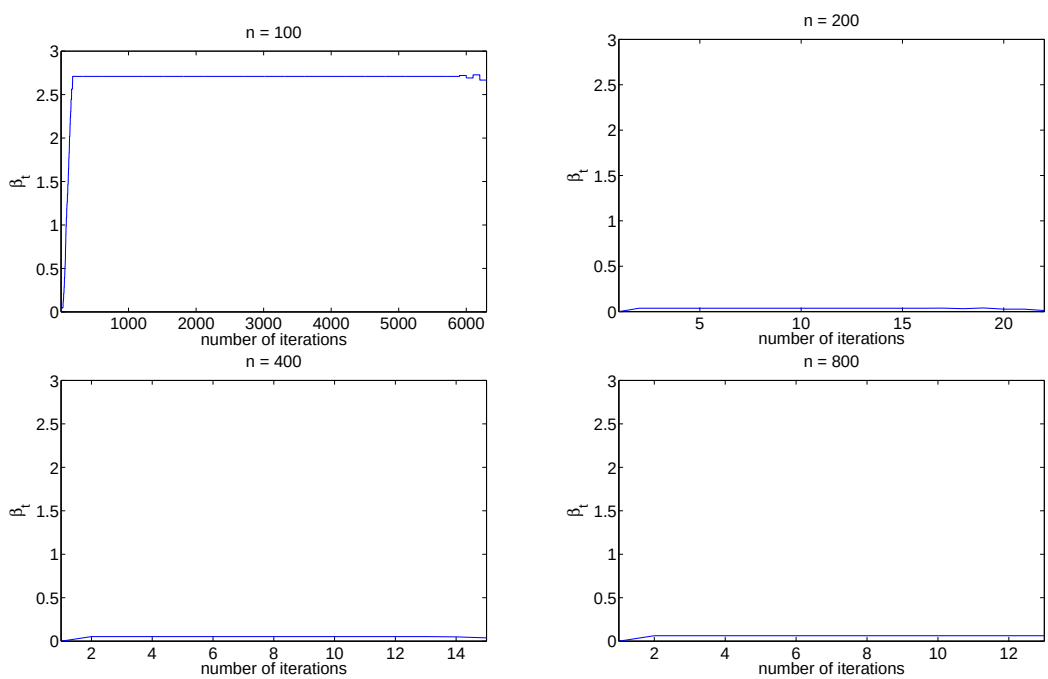

Figure 1: Ratios β_t; see (20) for the definition of these ratios.

3.3.2 Solving problems of very large scale

In Table 2, we show numerical results for accelerated smoothing techniques (with $\alpha = 3$, $\kappa = 10^{-12}$, and the same duality gap checking procedure as above) when applied to randomly generated instances of (17) that are of very large scale. Using accelerated smoothing techniques, we are able to solve approximately instances of (17) involving matrices of size $12'800 \times 12'800$ in about 8 hours and 40 minutes on average. Clearly, this performance would be out of reach for original smoothing techniques.

Accelerated smoothing techniques applied to large-scale instances of (17)

n	$1'600$	$3'200$	$6'400$	$12'800$
CPU time [sec]	158	791	$4'566$	$31'240$
Average # of iterations that are required in practice	13	13	13	13
Average # of iterations that are required in theory	$23'315$	$24'386$	$25'411$	$26'397$

Table 2: Average CPU time and number of iterations (in practice and in theory) that are required by accelerated smoothing techniques for finding an approximate solution to randomly generated large-scale instances of Problem (10) (with fixed accuracy $0.002\mathcal{L}'$ and with $m = 100$).

Acknowledgments

We gratefully thank Yurii Nesterov and Hans-Jakob Lüthi for many helpful discussions. This research is partially funded by the Swiss National Fund.

A Proof of Theorem 2.1

Choose $T \in \mathbb{N}_0$ and let the sequences $(x_t)_{t=0}^{T+1}$, $(u_t)_{t=0}^{T+1}$, $(z_t)_{t=0}^{T}$, $(\hat{x}_t)_{t=1}^{T+1}$, $(\gamma_t)_{t=0}^{T+1}$, $(\Gamma_t)_{t=0}^{T+1}$, $(\tau_t)_{t=0}^{T}$, and $(L_t)_{t=0}^{T}$ be generated by Algorithm 1. Recall that Inequality $(\mathcal{I}_t)$ holds for $0 \leq t \leq T$ if

$$\Gamma_t f(u_t) + \sum_{k=0}^{t-1} (L_{k+1} - L_k) \left(d_Q(z_{k+1}) - \frac{1}{2} \|z_k - \hat{x}_{k+1}\|_{\mathbb{R}^n}^2 \right) \leq \psi_t, \tag{$\mathcal{I}_t$}$$

where

$$\psi_t := \min_{x \in Q} \left\{ \sum_{k=0}^{t} \gamma_k \left(f(x_k) + \langle \nabla f(x_k), x - x_k \rangle \right) + L_t d_Q(x) \right\}.$$

By its definition (see Algorithm 1), the element $z_t \in Q$ is the minimizer to the above optimization problem, which allows us to rewrite ψ_t as:

$$\psi_t = \sum_{k=0}^{t} \gamma_k \left(f(x_k) + \langle \nabla f(x_k), z_t - x_k \rangle \right) + L_t d_Q(z_t).$$

We show by induction that Inequality $(\mathcal{I}_t)$ holds for any $0 \leq t \leq T$.

Lemma A.1 *Inequality $(\mathcal{I}_0)$ holds, that is, we have $\gamma_0 f(u_0) \leq \psi_0$.*

Proof: We apply the definition of u_0 (see Algorithm 1), Inequality (4), the condition on γ_0 saying that $\gamma_0 \in (0, 1]$, and Theorem 2.1.5 in [Nes03] in order to justify the following relations:

$$\begin{aligned} \psi_0 &:= \min_{x \in Q} \left\{ \gamma_0 \left(f(x_0) + \langle \nabla f(x_0, x - x_0 \rangle \right) + L_0 d_Q(x) \right\} \\ &= \gamma_0 \left(f(x_0) + \langle \nabla f(x_0), u_0 - x_0 \rangle \right) + L_0 d_Q(u_0) \end{aligned}$$

$$
\begin{aligned}
&\geq \gamma_0 \left(f(x_0) + \langle \nabla f(x_0), u_0 - x_0 \rangle\right) + \frac{L_0}{2} \|u_0 - x_0\|^2_{\mathbb{R}^n} \\
&\geq \gamma_0 \left(f(x_0) + \langle \nabla f(x_0), u_0 - x_0 \rangle + \frac{L_0}{2} \|u_0 - x_0\|^2_{\mathbb{R}^n}\right) \\
&\geq \gamma_0 f(u_0).
\end{aligned}
$$

■

Let us verify the inductive step.

Lemma A.2 *Let $0 \leq t \leq T-1$. If Inequality $(\mathcal{I}_t)$ holds, also $(\mathcal{I}_{t+1})$ is true.*

Proof: Let $0 \leq t \leq T-1$ and assume that $(\mathcal{I}_t)$ holds. We make the following two definitions:

$$
\begin{aligned}
\chi_t &:= \sum_{k=0}^{t-1} (L_{k+1} - L_k) \left(d_Q(z_{k+1}) - \frac{1}{2} \|z_k - \hat{x}_{k+1}\|^2_{\mathbb{R}^n}\right) \in \mathbb{R}, \\
s_t &:= \sum_{k=0}^{t} \gamma_k \nabla f(x_k) \in \mathbb{R}^n.
\end{aligned}
$$

In addition, we define the linear function:

$$
l_t : Q \to \mathbb{R} : x \mapsto l_t(x) = \sum_{k=0}^{t} \gamma_k \left(f(x_k) + \langle \nabla f(x_k), x - x_k \rangle\right).
$$

Choose $x \in Q$. The definition of z_t implies:

$$
0 \leq \left\langle L_t \nabla d_Q(z_t) + \sum_{k=0}^{t} \gamma_k \nabla f(x_k), x - z_t \right\rangle = \langle L_t \nabla d_Q(z_t) + s_t, x - z_t \rangle. \tag{21}
$$

As the Inequality $(\mathcal{I}_t)$ holds and as the function f is convex, we have:

$$
\psi_t \geq \Gamma_t f(u_t) + \chi_t \geq \Gamma_t \left(f(x_{t+1}) + \langle \nabla f(x_{t+1}), u_t - x_{t+1} \rangle\right) + \chi_t.
$$

This implies:

$$
\psi_t + \gamma_{t+1} \left(f(x_{t+1}) + \langle \nabla f(x_{t+1}), x - x_{t+1} \rangle\right) \geq \Gamma_{t+1} f(x_{t+1}) + \gamma_{t+1} \langle \nabla f(x_{t+1}), x - z_t \rangle + \chi_t,
$$

where we use the relations $\Gamma_{t+1} = \Gamma_t + \gamma_{t+1}$ and

$$
\begin{aligned}
\Gamma_t(u_t - x_{t+1}) + \gamma_{t+1}(x - x_{t+1}) &= \Gamma_t u_t - \Gamma_{t+1} x_{t+1} + \gamma_{t+1} x \\
&= \Gamma_t u_t - \Gamma_{t+1} \left(\tau_t z_t + (1 - \tau_t) u_t\right) + \gamma_{t+1} x \\
&= \Gamma_t u_t - \Gamma_{t+1} \left(\frac{\gamma_{t+1}}{\Gamma_{t+1}} z_t + \frac{\Gamma_t}{\Gamma_{t+1}} u_t\right) + \gamma_{t+1} x \\
&= \gamma_{t+1}(x - z_t).
\end{aligned}
$$

Combining the above inequality with the fact that $\psi_t = L_t d_Q(z_t) + l_t(z_t)$ and with (21), we observe:

$$
L_t d_Q(x) + l_{t+1}(x)
$$

$$
\begin{aligned}
&= L_t d_Q(x) + l_t(x) + \gamma_{t+1}\left(f(x_{t+1}) + \langle \nabla f(x_{t+1}), x - x_{t+1}\rangle\right)\\
&= L_t V_{z_t}(x) + \psi_t + \langle L_t \nabla d_Q(z_t) + s_t, x - z_t\rangle + \gamma_{t+1}\left(\langle \nabla f(x_{t+1}), x - x_{t+1}\rangle + f(x_{t+1})\right)\\
&\geq L_t V_{z_t}(x) + \psi_t + \gamma_{t+1}\left(f(x_{t+1}) + \langle \nabla f(x_{t+1}), x - x_{t+1}\rangle\right)\\
&\geq L_t V_{z_t}(x) + \Gamma_{t+1} f(x_{t+1}) + \gamma_{t+1}\langle \nabla f(x_{t+1}), x - z_t\rangle + \chi_t.
\end{aligned}
$$

With $\vartheta_t^{(1)} := (L_{t+1} - L_t)\, d_Q(z_{t+1})$, we thus get:

$$
\begin{aligned}
\psi_{t+1} &:= \min_{x\in Q}\left\{L_{t+1} d_Q(x) + l_{t+1}(x)\right\}\\
&= L_{t+1} d_Q(z_{t+1}) + l_{t+1}(z_{t+1})\\
&= \vartheta_t^{(1)} + L_t d_Q(z_{t+1}) + l_{t+1}(z_{t+1})\\
&\geq \vartheta_t^{(1)} + \min_{x\in Q}\left\{L_t d_Q(x) + l_{t+1}(x)\right\}\\
&\geq \vartheta_t^{(1)} + \min_{x\in Q}\left\{L_t V_{z_t}(x) + \Gamma_{t+1} f(x_{t+1}) + \gamma_{t+1}\langle \nabla f(x_{t+1}), x - z_t\rangle + \chi_t\right\}.
\end{aligned}
$$

Let $\vartheta_t^{(2)} := \frac{1}{2}(L_t - L_{t+1})\,\|z_t - \hat{x}_{t+1}\|_{\mathbb{R}^n}^2$. Using the construction rule for $\hat{x}_{t+1}$ and the fact that the inequality $V_z(x) \geq \|x - z\|_{\mathbb{R}^n}^2/2$ holds for any $x \in Q$ and $z \in Q^o$ (this relation follows from the strong convexity of d_Q), we obtain:

$$
\begin{aligned}
\psi_{t+1} &\geq \vartheta_t^{(1)} + L_t V_{z_t}(\hat{x}_{t+1}) + \Gamma_{t+1} f(x_{t+1}) + \gamma_{t+1}\langle \nabla f(x_{t+1}), \hat{x}_{t+1} - z_t\rangle + \chi_t\\
&\geq \vartheta_t^{(1)} + \frac{L_t}{2}\|z_t - \hat{x}_{t+1}\|_{\mathbb{R}^n}^2 + \Gamma_{t+1} f(x_{t+1}) + \gamma_{t+1}\langle \nabla f(x_{t+1}), \hat{x}_{t+1} - z_t\rangle + \chi_t\\
&= \vartheta_t^{(1)} + \vartheta_t^{(2)} + \frac{L_{t+1}}{2}\|z_t - \hat{x}_{t+1}\|_{\mathbb{R}^n}^2 + \Gamma_{t+1} f(x_{t+1}) + \gamma_{t+1}\langle \nabla f(x_{t+1}), \hat{x}_{t+1} - z_t\rangle + \chi_t\\
&= \vartheta_t^{(1)} + \vartheta_t^{(2)} + \chi_t + \Gamma_{t+1}\left(\frac{L_{t+1}}{2\Gamma_{t+1}}\|z_t - \hat{x}_{t+1}\|_{\mathbb{R}^n}^2 + f(x_{t+1}) + \tau_t\langle \nabla f(x_{t+1}), \hat{x}_{t+1} - z_t\rangle\right).
\end{aligned}
$$

As $\tau_t^2 \leq \Gamma_{t+1}^{-1}$ and as $x_{t+1} - \tau_t z_t = (1 - \tau_t)u_t = u_{t+1} - \tau_t \hat{x}_{t+1}$, this inequality yields to:

$$
\begin{aligned}
\psi_{t+1} &\geq \vartheta_t^{(1)} + \vartheta_t^{(2)} + \chi_t + \Gamma_{t+1}\left(\frac{L_{t+1}\tau_t^2}{2}\|z_t - \hat{x}_{t+1}\|_{\mathbb{R}^n}^2 + f(x_{t+1}) + \tau_t\langle \nabla f(x_{t+1}), \hat{x}_{t+1} - z_t\rangle\right)\\
&= \vartheta_t^{(1)} + \vartheta_t^{(2)} + \chi_t + \Gamma_{t+1}\left(\frac{L_{t+1}}{2}\|u_{t+1} - x_{t+1}\|_{\mathbb{R}^n}^2 + f(x_{t+1}) + \langle \nabla f(x_{t+1}), u_{t+1} - x_{t+1}\rangle\right).
\end{aligned}
$$

It remains to apply (5):

$$
\begin{aligned}
\psi_{t+1} &\geq \vartheta_t^{(1)} + \vartheta_t^{(2)} + \Gamma_{t+1} f(u_{t+1}) + \chi_t\\
&= \sum_{k=0}^{t}(L_{k+1} - L_k)\left(d_Q(z_{k+1}) - \frac{1}{2}\|z_k - \hat{x}_{k+1}\|_{\mathbb{R}^n}^2\right) + \Gamma_{t+1} f(u_{t+1}).
\end{aligned}
$$

■

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.

[BBN11] M. Baes, M. Bürgisser, and A. Nemirovski, *A randomized Mirror-Prox method for solving matrix saddle-point problems*, Tech. report, ETH Zurich / Georgia Institute of Technology, 2011, (available at http://arxiv.org/abs/1112.1274).

[BCG11] S. Becker, E. Candès, and M. Grant, *Templates for convex cone problems with applications to sparse signal recovery*, Mathematical Programming Computation **3** (2011), 165–218.

[BT09] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences archive **2** (2009), 183–202.

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

[dK12] A. d'Aspremont and N. El Karoui, *A stochastic smoothing algorithm for semidefinite programming*, Tech. report, CMAP, Ecole Polytechnique, Paris, 2012.

[GG05] J. Gondzio and A. Grothey, *Direct Solution of Linear Systems of Size 10^9 Arising in Optimization with Interior Point Methods*, Parallel Processing and Applied Mathematics, PPAM 2005 (R. Wyrzykowski, J. Dongarra, N. Meyer, and J. Wasniewski, eds.), Lecture Notes in Computer Science, vol. 3911, Springer, 2005, pp. 513–525.

[HR00] C. Helmberg and F. Rendl, *A Spectral Bundle Method for Semidefinite Programming*, SIAM Journal on Optimization **10** (2000), no. 3, 673–696.

[LLM] G. Lan, Z. Lu, and R. Monteiro, *Primal-dual First-order Methods with $\mathcal{O}(1/\epsilon)$ Iteration-complexity for Cone Programming*, To appear in Mathematical Programming.

[Nes83] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$*, Doklady AN SSSR (translated as Soviet Math. Dokladi) **3** (1983), 543–547.

[Nes03] ———, *Introductory lectures on convex optimization: a basic course*, Applied Optimization, vol. 87, Kluwer Academic Publishers, 2003.

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

[Nes07a] ———, *Gradient methods for minimizing composite objective function*, CORE Discussion Paper 76, Center for Operation Research and Econometrics, Université catholique de Louvain, 2007.

[Nes07b] ———, *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.

[Nes10] ———, *Efficiency of coordinate descent methods on huge-scale optimization problems*, CORE Discussion Paper 10, Center for Operation Research and Econometrics, Université catholique de Louvain, 2010.

[Nes12] ———, *Subgradient methods for huge-scale optimization*, CORE Discussion Paper 16, Center for Operation Research and Econometrics, Université catholique de Louvain, 2012.

[NJLS09] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization **19** (2009), no. 4, 1574–1609.

[NY83] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, John Wiley, 1983.

[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.

[RT11] P. Richtárik and M. Takáč, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, available on optimization online, School of Mathematics, University of Edinburgh, 2011.

[Tse08] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, Tech. report, Department of Mathematics, University of Washington, 2008.