ℓ_1 Minimization via Randomized First Order Algorithms

Anatoli Juditsky[*] Fatma Kılınç Karzan[†] Arkadi Nemirovski[‡]

June 5, 2011

Abstract

In this paper we propose randomized first-order algorithms for solving bilinear saddle points problems. Our developments are motivated by the need for sublinear time algorithms to solve large-scale *parametric* bilinear saddle point problems where cheap online assessment of solution quality is crucial. We present the theoretical efficiency estimates of our algorithms and discuss a number of applications, primarily to the problem of ℓ_1 minimization arising in sparsity-oriented Signal Processing. We demonstrate, both theoretically and by numerical examples, that when seeking for medium-accuracy solutions of large-scale ℓ_1 minimization problems, our randomized algorithms outperform significantly (and progressively as the sizes of the problem grow) the state-of-the art deterministic methods.

1 Introduction

This paper is motivated by the desire to develop efficient *randomized* first-order methods for solving well-structured large-scale convex optimization problems. Our primary (but not the only) target is the ℓ_1-minimization problem

$$\mathrm{Opt}_p = \min_u \{\|u\|_1 : \|Au - b\|_p \leq \delta\} \quad [A = [A_1, ..., A_n] \in \mathbf{R}^{m\times n}, m, n > 2], \tag{1}$$

where $p = \infty$ ("uniform fit") or $p = 2$ ("ℓ_2-fit"). We are interested in the large-scale case, where the sizes m, n of (possibly dense) matrix A are in the range of thousands/tens of thousands. Efficient solutions to the problems of this type are of paramount importance for sparsity-oriented Signal Processing, in particular, in Compressed Sensing (see [2, 3, 6] and references therein). To give an overview of our results, here is what our approach yields for (1):

[*] LJK, Université J. Fourier, B.P. 53, 38041 Grenoble Cedex 9, France, `Anatoli.Juditsky@imag.fr`

[†] Georgia Institute of Technology, Atlanta, Georgia 30332, USA, `fkilinc@gatech.edu`

[‡] Georgia Institute of Technology, Atlanta, Georgia 30332, USA, `nemirovs@isye.gatech.edu`

Research of the second and the third authors was supported by the ONR grant N000140811104 and the NSF grant DMS-0914785.

Proposition 1.1 *Assume that* (1) *is feasible, δ is small enough, namely, $2m^{\frac{1}{p}}\delta \le \|b\|_p$. Given $\epsilon \in (0, \frac{1}{2}\mathrm{Opt}_p\|A\|_{1,p}]$,*[1] *let our goal be to find an ϵ-solution to (1), that is, a point x_ϵ satisfying*

$$\|x_\epsilon\|_1 \le \mathrm{Opt}_p \;\&\; \|Ax_\epsilon - b\|_p \le \delta + \epsilon.$$

Then, for every tolerance $\chi \in (0, 1/2]$, the outlined goal can be achieved with probability $\ge 1 - \chi$

(i) *in the case of $p = \infty$ (uniform fit) – in at most*

$$O(1)\left[\frac{\sqrt{\ln(m)\ln(n)}\|A\|_{1,\infty}\mathrm{Opt}_\infty}{\epsilon}\ln\left(\frac{\sqrt{\ln(m)\ln(n)}\|A\|_{1,\infty}\mathrm{Opt}_\infty}{\chi\epsilon}\right)\right]^2$$

steps of a randomized algorithm, with computational effort per step reduced to extracting from A two columns and two rows, given their indexes, plus "computational overhead" of $O(1)(m + n)$ operations.

(ii) *in the case of $p = 2$ (ℓ_2 fit) – in at most*

$$O(1)\left[\frac{\ln(mn)\kappa(A)\|A\|_{1,2}\mathrm{Opt}_2}{\epsilon}\ln\left(\frac{\ln(mn)\kappa(A)\|A\|_{1,2}\mathrm{Opt}_2}{\chi\epsilon}\right)\right]^2, \quad \kappa(A) = \frac{\sqrt{m}\|A\|_{1,\infty}}{\|A\|_{1,2}},$$

steps of a randomized algorithm with the same as in (i) *computational effort per step.*

Furthermore, there exists a randomized preprocessing of the data $[A, b]$ of the problem (1) of computational cost not exceeding $O(1)mn\ln(m)$, which ensures with probability $\ge 1 - \chi$ that $\kappa(A) \le O(1)\sqrt{\ln(mn/\chi)}$.

Note that the best known so far complexity of finding ϵ-solution to a large-scale problem (1) by a *deterministic* algorithm is *at least* $O(1)\frac{\sqrt{\ln(m)\ln(n)}\|A\|_{1,\infty}\mathrm{Opt}_\infty}{\epsilon}$ ($p = \infty$) or $\frac{\sqrt{\ln(n)}\|A\|_{1,2}\mathrm{Opt}_2}{\epsilon}$ ($p = 2$) steps[2] with complexity of a step dominated by the necessity to perform $O(1)$ multiplications $x \mapsto Ax$, $y \mapsto A^Ty$. When A is dense, the resulting operations count is, up to logarithmic terms, of order of $N_{\mathrm{det}} = \frac{mn}{\nu}$, where $\nu = \frac{\epsilon}{\|A\|_{1,p}\mathrm{Opt}_p}$ can be naturally interpreted as relative accuracy. For the randomized algorithms underlying Proposition 1.1, this count, again, up to logarithmic terms, is of order of $N_{\mathrm{rand}} = \frac{m+n}{\nu^2}$ (uniform fit) and $N_{\mathrm{rand}} = \frac{m+n}{\nu^2} + mn$ (ℓ_2 fit). We see that when $\nu \ll 1$ is fixed and m, n grow, the randomized algorithms eventually outperform the deterministic ones, becoming more significant as the problem size grows. Numerical results presented in Section 5 demonstrate that this acceleration is not a purely academic phenomenon and can be of real practical interest.

Our approach is based on saddle point reformulation of well-structured convex minimization problems and is applicable when the resulting saddle point problems are bilinear; in this respect, it goes back to the breakthrough paper of Nesterov [14]. The deterministic saddle point prototypes of the randomized algorithms we develop here were proposed in

[1] Here and below $\|A\|_{1,p} = \max_j \|A_j\|_p$ stands for the norm of the mapping $x \mapsto Ax$ induced by the norms $\|\cdot\|_1$ and $\|\cdot\|_p$ in the argument and the image spaces, respectively

[2] The indicated bounds are attainable, provided Opt_p is known in advance.

[11] and [12] and the prototypes of our randomization scheme were proposed in [13, Section 3.3] and [9]. In this paper, we demonstrate that in the case of a bilinear saddle point problem, a better randomization is possible. The advantage of this new randomization over those prototypes lies in the immediate possibility to assess, in a computationally cheap fashion, the quality of the resulting approximate solutions. This possibility is instrumental when solving *parametric* bilinear saddle point problems. In particular many important applications including the problems of the form (1) reduce to the class of parametric bilinear saddle point problems which we introduce and study in Section 2.2. In the hindsight, one can recognize utilizing a particular case of this randomization technique leads to the sublinear time randomized algorithm for solving matrix games due to Grigoriadis and Khachiyan [7].

The main body of this paper is organized as follows. In Section 2, we present a saddle-point-based framework for our developments together with a sample of interesting optimization problems fitting this framework. This sample includes, along with ℓ_1 minimization, the (semidefinite relaxation of the) problem of low-dimensional approximation to a collection of points in $\mathbf{R}^d$ and a specific version of the Support Vector Machine problem. Randomized algorithms for the problems fitting to our framework are developed and analyzed in Sections 3 and 4. Section 5 presents encouraging results of preliminary numerical experiments aimed at comparing the performance of the proposed randomized algorithm and a state-of-the-art deterministic algorithm as applied to large-scale ℓ_1 minimization problem. All proofs are relegated to the appendix.

2 Problems and Goals

We start with specifying and motivating two problems to be discussed in the paper and our goals.

2.1 A Bilinear Saddle Point Problem

2.1.1 The problem

The first generic problem we are interested in is a Bilinear Saddle Point (BSP) problem

$$\begin{array}{rcl}\mathrm{SV} & = & \min\limits_{z_1\in Z_1}\max\limits_{z_2\in Z_2}\phi(z_1,z_2),\\ \phi(z_1,z_2) & = & \upsilon+\langle a_1,z_1\rangle+\langle a_2,z_2\rangle+\langle z_2,Bz_1\rangle:\quad Z[=Z_1\times Z_2]\to\mathbf{R},\end{array}\qquad(\mathcal{S})$$

where Z_i are nonempty convex compact sets in Euclidean spaces E_i, $i=1,2$. Recall that $(\mathcal{S})$ gives rise to two dual to each other convex optimization programs

$$\begin{array}{rclr}\mathrm{Opt}(P) & = & \min\limits_{z_1\in Z_1}\overline{\phi}(z_1):=\max\limits_{z_2\in Z_2}\phi(z_1,z_2) & (P)\\ \mathrm{Opt}(D) & = & \max\limits_{z_2\in Z_2}\underline{\phi}(z_2):=\min\limits_{z_1\in Z_1}\phi(z_1,z_2) & (D)\end{array}\qquad(2)$$

with $\mathrm{Opt}(P)=\mathrm{Opt}(D)=\mathrm{SV}$, and to the variational inequality : find $z_*\in Z:=Z_1\times Z_2$ such that

$$\langle F(z),z-z_*\rangle\geq 0\ \text{ for all } z\in Z,\qquad(3)$$

where $F: Z \mapsto E_1 \times E_2$, is an affine monotone operator given by

$$\begin{array}{rcl} F(z_1,z_2) &=& \left[F_1(z_2)=\frac{\partial\phi(z_1,z_2)}{\partial z_1}; F_2(z_1)=-\frac{\partial\phi(z_1,z_2)}{\partial z_2}\right]=a+\mathcal{A}[z_1;z_2],\\ && a=[a_1;-a_2],\ \ \mathcal{A}=\left[\begin{array}{c|c} & B^* \\ \hline -B & \end{array}\right], \end{array}$$

(here B^* stands for the conjugate of B). Note that $\mathcal{A}$ is skew-symmetric: $\mathcal{A}^*=-\mathcal{A}$ and

$$\langle z, \mathcal{A}z\rangle = 0\ \forall z \in E := E_1 \times E_2. \tag{4}$$

It is well known that the solutions to $(\mathcal{S})$ — the saddle points of ϕ on $Z_1 \times Z_2$ — are exactly the pairs $z=[z_1;z_2]$ comprised of optimal solutions to problems (P) and (D) in (2), same as are exactly the solutions to the variational inequality (3). We quantify the accuracy of candidate solutions $z=[z_1;z_2]\in Z$ to $(\mathcal{S})$ by the *saddle point residual*

$$\epsilon_{\rm sad}(z)=\overline{\phi}(z_1)-\underline{\phi}(z_2)=\underbrace{\left[\overline{\phi}(z_1)-{\rm Opt}(P)\right]}_{\geq 0}+\underbrace{\left[{\rm Opt}(D)-\underline{\phi}(z_2)\right]}_{\geq 0}. \tag{5}$$

2.1.2 Assumptions and goal

When speaking about a BSP problem $(\mathcal{S})$, our goal is to solve the problem within a given accuracy $\epsilon>0$, that is, to find $z^\epsilon \in Z$ such that $\epsilon_{\rm sad}(z^\epsilon)\leq\epsilon$. Deterministic first order algorithms achieve this goal by working with the values of the associated operator F at the iterates z_t, $t=1,2,...$, generated by the method. When Z is simple and the problem is large-scale, computing the values $F(z_t)$ is the "leading term" in the computational effort. Our goal in this paper is to replace relatively expensive (in the large scale case) exact values $F(z_t)$ with their computationally cheap unbiased random estimates. Specifically, we assume that

> **[P]** every point $z\in Z$ is associated with a probability distribution P_z such that
>
> - P_z is supported on Z and $\mathbf{E}_{\zeta\sim P_z}\{\zeta\}=z$;
> - Given z, we can sample from the distribution P_z.

Under these assumptions, in order to get an unbiased estimate of $F(z_t)$, it suffices to draw a $\zeta_t\sim P_{z_t}$ and to take $F(\zeta_t)$ as a desired estimate of $F(z_t)$. In order to make this approach meaningful, the computational price of generating ζ_t and subsequent computation of $F(\zeta_t)$ should be significantly less than the price of a straightforward computation of $F(z_t)$. This requirement guided us in the selection of applications to be considered below as well as in building the corresponding saddle point reformulations .

Note that the deterministic algorithms remain in the scope of our approach since we always have an option to define P_z as δ_z (the unit mass sitting at z).

2.1.3 Application example: low dimensional approximation

We consider the following problem (related to a dimension reduction problem in statistics, see, e.g., [5]): let $V = \{v_1, ..., v_N\}$ be a collection of unit vectors in $\mathbf{R}^n$, and $d < n$ be a positive integer. We want to find a linear subspace $E \subset \mathbf{R}^n$ of dimension d such that the *deviation* $\delta(V, E)$ of the collection from E — the maximal, over i, Euclidean distance between v_i and E — is as small as possible.

Letting $\mathit{\Pi}^d$ be the family of all orthogonal projectors of $\mathbf{R}^n$ onto d-dimensional linear subspaces, the problem reads

$$\mathrm{Opt}_* = \max_{\Pi \in \mathit{\Pi}^d} \min_{1 \le i \le N} v_i^T \Pi v_i$$

and seems to be computationally intractable. It, however, admits the tractable relaxation

$$\mathrm{Opt} = \max_{Q \in \mathcal{P}^d} \min_{1 \le i \le N} v_i^T Q v_i, \quad \mathcal{P}^d = \{Q \in \mathbf{S}^n :\ 0 \preceq Q \preceq I,\ \mathrm{Tr}(Q) = d\}. \tag{6}$$

We refer to (6) as to the *problem of low dimensional approximation*. We clearly have $\mathrm{Opt}_* \le \mathrm{Opt} \le 1$, whence $\delta^2 := 1 - \mathrm{Opt} \le \delta_*^2 := 1 - \mathrm{Opt}_*$; note that δ_* is the deviation of V from the "ideal" d-dimensional space E_* underlying Opt_*. It is easily seen (see Lemma A.1 of Section A.1) that if Q_* is an optimal solution to the relaxation (6) and E is spanned by the d leading eigenvectors of Q_*, then $\delta(V, E) \le \sqrt{d+1}\delta_*$, that is, approximation (6) admits some quality guarantees.

Now, (6) is nothing but the BSP problem:

$$1 - \mathrm{Opt} = \min_{Q \in \mathcal{P}^d} \max_{\lambda \in \Delta_N} \left[1 - \mathrm{Tr}\left(Q \sum\nolimits_{i=1}^N \lambda_i v_i v_i^T\right)\right], \quad \Delta_N = \left\{\lambda \in \mathbf{R}_+^N :\ \sum\nolimits_i \lambda_i = 1\right\}. \tag{7}$$

In terms of $(\mathcal{S})$, E_1 is the space $\mathbf{S}^n$ of symmetric $n \times n$ matrices with Frobenius inner product, $Z_1 = \mathcal{P}^d \subset E_1$, $E_2 = \mathbf{R}^N$, $Z_2 = \Delta_N$. The associated operator F is

$$F(z_1, z_2) = F(Q, \lambda) = [\underbrace{-\sum\nolimits_{i=1}^N \lambda_i v_i v_i^T}_{F_1(z_2)};\ \underbrace{[v_1^T Q v_1;\ ...;\ v_N^T Q v_N]}_{F_2(z_1)}]. \tag{8}$$

Assuming that v_i are dense, the arithmetic cost of computing the value of F at a given point is $O(n^2 N)$. Now let us specify the distributions P_z, $z = (Q, \lambda) \in Z = Z_1 \times Z_2$. In order to generate $\zeta \sim P_{(Q,\lambda)}$, we proceed as follows:

- Given $Q \in \mathcal{P}^d$, we build the eigenvalue decomposition $Q = U\mathrm{Diag}\{q\}U^T$. Note that $q \in \Delta_{n,d} = \{q \in \mathbf{R}^n :\ 0 \le q_i \le 1 \forall i,\ \sum_{i=1}^n q_i = d\}$. The extreme points of $\Delta_{n,d}$ are Boolean vectors with exactly d nonzero entries. There exists a simple algorithm (see Section A.1) which, given as input a vector $q \in \Delta_{n,d}$, builds in $O(1)dn^2$ a.o. n extreme points q^j, $1 \le j \le n$, of $\Delta_{n,d}$ along with weights $\mu_j \ge 0$, $\sum_j \mu_j = 1$, such that $q = \sum_j \mu_j q^j$. We run this algorithm to build $\{q^j, \mu_j\}_{j=1}^n$, pick $\jmath \in \{1, ..., n\}$ at random, with $\mathrm{Prob}\{\jmath = j\} = \mu_j$, $j = 1, ..., n$, and set $\zeta_1^\jmath = U\mathrm{Diag}\{q^\jmath\}U^T$.

- Given $\lambda \in \Delta_N$, we pick $\imath \in \{1, ..., N\}$ at random, with $\mathrm{Prob}\{\imath = i\} = \lambda_i$, $1 \le i \le N$, and set $\zeta_2^\imath := e_\imath$, where e_i, $i = 1, ..., N$, are standard basic orths in $\mathbf{R}^N$.

- Finally, we set $\zeta=\zeta^{\imath\jmath}:=[\zeta_1^{\jmath};\zeta_2^{\imath}]\in\mathcal{P}^d\times\Delta_N$.

The family of distributions $P_{(Q,\lambda)}$ clearly satisfies [P]. The "setup costs" for sampling from $P_{(Q,\lambda)}$ reduce to those of 1) computing the eigenvalue decomposition of Q, 2) building $q^1,...,q^n$, $\mu_1,...,\mu_n$ (this cost is $O(n^3+dn^2)$ a.o.) and 3) computing the "cumulative distributions" $\{\mu^j=\sum_{s=1}^j\mu_s\}_{j=1}^n$ and $\{\lambda^i=\sum_{s=1}^i\lambda_s\}_{i=1}^N$ (what amounts to $O(n+N)$ a.o.). After the setup cost is paid, a sample $(\imath,\jmath)$ can be generated at the cost of just $O(\ln(n+N))$ a.o. Now let us look at the cost of computing $F(\zeta^{\imath\jmath})$ given $\imath,\jmath$. We have

$$F(\zeta^{\imath\jmath})=\left[-v_\imath v_\imath^T;\{v_i^TU\mathrm{Diag}\{q^\jmath\}U^Tv_i\}_{i=1}^N\right].$$

Since $q^\jmath$ has just d nonzero entries, all equal to 1, let the indices of the entries be $j_1,...,j_d$, we have $v_i^TU\mathrm{Diag}\{q^\jmath\}U^Tv_i=\sum_{\ell=1}^d(U_{j_\ell}^Tv_i)^2$, where U_j is j^{th} column of U. We see that computing $F(\zeta^{\imath\jmath})$ costs $O(n^2+dnN)$ a.o. Thus, the total cost (including that of the setup) of drawing a sample ζ from $P_{(Q,\lambda)}$ and computing $F(\zeta)$ is

$$O(n^3+dn^2+n^2+dnN)=O(n^3+dnN)\ \text{ a.o.}$$

When $d\ll n\ll N$, this cost is much smaller than the cost $O(n^2N)$ of computing $F(z)$ at a "general position" point $z=(Q,\lambda)\in Z$.

2.2 A Generalized Bilinear Saddle Point Problem

2.2.1 The problem

Assume that we are given a *single-parameter family* of bilinear saddle point problems

$$\mathrm{SV}(\rho)=\min_{z_1\in Z_1}\max_{z_2\in Z_2}\phi^\rho(z_1,z_2):=\phi(z_1,z_2)+\rho\psi(z_1,z_2),\tag{9}$$

where $\rho\geq0$ is a parameter and $\phi(z_1,z_2)$, $\psi(z_1,z_2)$ are bi-affine in z_1 and z_2. The *Generalized Bilinear Saddle Point* (GBSP) problem associated with this family is, by definition, the optimization program

$$\rho_*=\max\{\rho\geq0:\mathrm{SV}(\rho)\leq0\}\tag{10}$$

A highly desirable property of a GBSP problem, relative to our approach, is the convexity of SV(ρ) as a function of $\rho\geq0$. To ensure this property, we make from now on the following assumption on the structure of (9):

> **[A.1]** *$Z_1=Z_{11}\times Z_{12}$ is the direct product of two convex compact sets, and the bilinear functions $\phi(z_1,z_2)$, $\psi(z_1,z_2)$ in (9) are of the form*
>
> $$\begin{array}{rcl}\phi(z_1=[z_{11};z_{12}],z_2)&=&\upsilon+\langle a_{11},z_{11}\rangle+\langle b,z_2\rangle+\langle z_2,Bz_{11}\rangle,\\ \psi(z_1=[z_{11};z_{12}],z_2)&=&\chi+\langle a_{12},z_{12}\rangle+\langle c,z_2\rangle+\langle z_2,Cz_{12}\rangle,\end{array}\tag{11}$$
>
> *that is, $\phi(z_1,z_2)$ and $\psi(z_1,z_2)$ as functions of z_1 depend each on its own "block" of z_1, and these blocks z_{11} and z_{12}, independently of each other, run through the respective convex compact sets Z_{11} and Z_{12}.*

From now on, we denote by $F^\rho(z) = \Phi(z) + \rho\Psi(z)$ the affine monotone operator associated with ϕ^ρ according to (3).

Lemma 2.1 *In the case of* **A.1** *the function* $\mathrm{SV}(\rho)$ *given by* (9) *is convex in* $\rho \geq 0$.

From now on we assume, in addition to **A.1**, that

> [**A.2**] *Function* $\mathrm{SV}(\rho)$ *given by* (9) *is nonpositive somewhere on* $\mathbf{R}_{++}$ *and tends to* $+\infty$ *as* $\rho \to +\infty$,

which implies solvability of (10) and positivity of ρ_*.

The goal. Given a GBSP problem (9) – (10) and a tolerance $\epsilon > 0$, our goal will be to find an ϵ-solution to the problem, that is, a pair ρ_ϵ, $z_1^\epsilon \in Z_1$ such that

$$\rho_\epsilon \geq \rho_* \text{ and } \max_{z_2 \in Z_2} \phi^{\rho_\epsilon}(z_1^\epsilon, z_2) \leq \rho_\epsilon \epsilon \tag{12}$$

We are about to point out several important application examples for GBSP problem.

2.2.2 Application example: ℓ_1 minimization with ℓ_p fit

The problem of interest is

$$\mathrm{Opt} = \min_x \{\|x\|_1 : \|Ax - b\|_p \leq \delta\} \quad [A \in \mathbf{R}^{m \times n}]. \tag{13}$$

Different versions of this problem arise in sparsity-oriented Signal Processing and Compressed Sensing. Setting $x = \rho u$, $\|u\|_1 \leq 1$, we rewrite the problem equivalently as

$$\frac{1}{\mathrm{Opt}} = \rho_* = \max\left\{\rho : \min_{\|u\|_1 \leq 1} \|Au - \rho b\|_p - \rho\delta \leq 0\right\}, \tag{14}$$

or, which is the same as

$$\frac{1}{\mathrm{Opt}} = \rho_* = \max\left\{\rho : \Phi(\rho) = \min_{\|u\|_1 \leq 1,\, \|v\|_p \leq 1} \|Au - \rho b - \rho\delta v\|_\infty \leq 0\right\}.$$

This is nothing but the GBSP problem (9) with $SV(\rho) = \phi^\rho(z)$, $z \in Z$, given by

$$\begin{array}{c} \phi^\rho(z_1(=[z_{11}; z_{12}]), z_2) = z_2^T J_m^T (AJ_n z_{11} - \rho[b + \delta z_{12}]), \\ Z_1 = \underbrace{\Delta_{2n}}_{Z_{11}} \times \underbrace{\{z_{12} \in \mathbf{R}^m : \|z_{12}\|_p \leq 1\}}_{Z_{12}}, \; Z_2 = \Delta_{2m}, \end{array} \tag{15}$$

where we denote $J_k = [I_k, -I_k]$, I_k being $k \times k$ identity matrix. This problem satisfies [**A.1**]; when $\|b\|_p > \delta$ (otherwise the optimal solution to (13) is $x = 0$), the problem satisfies [**A.2**] as well. The associated saddle value function is

$$\begin{array}{rcl} \mathrm{SV}(\rho) & = & \max\limits_{z_2 \in \Delta_{2m}} \min\limits_{z_{11} \in \Delta_{2n},\, z_{12} \in Z_{12}} \left[z_2^T J_m^T (AJ_n z_{11} - \rho[b - \delta z_{12}])\right] \\ & = & \max\limits_{w = J_m z_2,\, z_2 \in \Delta_{2m}} \min\limits_{u = J_n z_1,\, z_1 \in \Delta_{2n}} \min\limits_{z_{12} \in Z_{12}} \left[w^T (Au - \rho[b + \delta z_{12}])\right] \\ & = & \max\limits_{\|w\|_1 \leq 1} \min\limits_{\|u\|_1 \leq 1} \min\limits_{\|v\|_p \leq 1} \left[w^T (Au - \rho[b + \delta v])\right] = \Phi(\rho). \end{array}$$

Suppose that we are given an ε-solution ρ_ε, $z_1^\varepsilon = [z_{11}^\varepsilon; z_{12}^\varepsilon]$ to the problem (12), (15) with $\varepsilon = \epsilon m^{-\frac{1}{p}}$. When setting $x_\varepsilon = \rho_\varepsilon^{-1} J_n z_{11}^\varepsilon$ and $v_\varepsilon = z_{12}^\varepsilon$ we get an approximate solution to (13) such that

$$\|x_\epsilon\|_1 \leq \mathrm{Opt}\ \&\ \|Ax_\epsilon - b\|_p \leq \|\delta v_\varepsilon\|_p + \|Ax_\varepsilon - b - \delta v_\varepsilon\|_p \leq \delta + \varepsilon m^{1/p} = \delta + \epsilon.$$

Finally, we associate with $z = [z_{11}; z_{12}; z_2] \in Z = Z_1 \times Z_2$ distribution P_z satisfying [P], namely, as follows. Note that for $z \in Z$, z_{11} and z_2 are vectors from the standard simplices and thus can be considered as probability distributions on the corresponding index sets $\{1, ..., 2n\}$, $\{1, ..., 2m\}$. To generate $\zeta = [\zeta_{11}; \zeta_{12}; \zeta_2] \sim P_z$, we draw at random index $\imath$ from the distribution z_{11} and make $[\zeta_{11}]_\imath = 1$ the only nonzero entry in ζ_{11}. ζ_2 is built similarly, with z_2 in the role of z_{11}, and ζ_{12} is nothing but z_{12}. It is immediately seen that it takes just $O(m+n)$ a.o. to generate a sample $\zeta \sim P_z$ and to compute the vector $F^\rho(\zeta)$.

It is worth to mention that in the important case $p = \infty$ the construction of the GBSP which corresponds to (13) can be substantially simplified. Indeed, one can see immediately that for $p = \infty$ (14) is equivalent to the GBSP problem on the direct product of just two unit ℓ_1-balls (since $\|Az_1 - b\|_\infty = \max\limits_{\|z_2\|_1 \leq 1} z_2^T(Az_1 - b)$). It is more convenient to pass from ℓ_1-balls to the standard simplexes, as it was done in the case of (15). The resulting GBSP problem is given by

$$\begin{array}{c} \phi^\rho(z_1, z_2) = z_2^T J_m^T A J_n z_1 - \rho z_2^T J_m^T b - \rho\delta, \\ Z_1 = Z_{11} = \Delta_{2n},\ Z_{12} = \{0\},\ Z_2 = \Delta_{2m}, \end{array} \tag{16}$$

and satisfies **[A.1]** and **[A.2]** when $\delta < \|b\|_\infty$.

2.2.3 Application example: ℓ_1 Support Vector Machine.

One of the "statistically solid" SVM models (see [4] and [16, Section 2.3.3]) is as follows. We are given a *training sample* — a matrix $X \in \mathbf{R}^{m\times n}$ with rows representing feature vectors, and a vector $y \in \mathbf{R}^m$ with entries ± 1 representing labels. Setting $R = \|X\|_\infty := \max_{i,j} |X_{ij}|$, $Y = \mathrm{Diag}\{y\}$ and $\mathbf{1} = [1; ...; 1] \in \mathbf{R}^m$, we want to solve the margin optimization problem

$$\mathrm{Opt} = \max_{w,b,\rho} \left\{\rho : \|w\|_1 \leq 1, \|[\rho\mathbf{1} - Y[Xw + b\mathbf{1}]]_+\|_2 \leq R\right\}, \tag{17}$$

where $[z]_+$ is the vector with coordinates $[z_i]_+ := \max[z_i, 0]$. We can convert this problem into a GBSP one as follows. Observe first that

$$\begin{array}{rcl} \mathrm{Opt} & = & \max\left\{\rho : \min\limits_{\|w\|_1 \leq 1,\, \|v\|_2 \leq 1} \max\limits_{1 \leq i \leq m} [\rho\mathbf{1} - Y[Xw + b\mathbf{1}] - Rv]_i \leq 0\right\} \\ & = & \max\left\{\rho : \min\limits_{\|w\|_1 \leq 1,\, \|v\|_2 \leq 1, b} \max\limits_{u \in \Delta_m} u^T[\rho\mathbf{1} - Y[Xw + b\mathbf{1}] - Rv] \leq 0\right\}. \end{array}$$

Assuming that the entries of y contain both 1 and -1 and setting $\Delta_m^+ = \{u \in \Delta_m : y^T u = 0\}$, we have $\min\limits_b \max\limits_{u\in\Delta_m} u^T[\rho\mathbf{1} - Y[Xw + b\mathbf{1}] - Rv] = \max\limits_{u \in \Delta_m^+} u^T(\rho\mathbf{1} - YXw - Rv)$. Hence, when setting $w = J_n s$, we come to

$$\mathrm{Opt} \;=\; \max\left\{\rho : \min_{s\in\Delta_{2n}, \|v\|_2\leq 1} \max_{u\in\Delta_m^+} u^T[\rho\mathbf{1} - YXJ_n s - Rv] \leq 0\right\}.$$

We see that (17) is equivalent to the GBSP problem given by

$$\begin{array}{c}\phi^\rho(z_1 = [z_{11} = s; z_{12} = v], z_2 = u) = u^T[\rho\mathbf{1} - YXJ_n s - Rv],\\ Z_1 = \{[s;v] : s \in \Delta_{2n}, \|v\|_2 \leq 1\}, Z_2 = \Delta_m^+.\end{array}$$

Note that this problem clearly satisfies $\mathbf{A.1-2}$. Besides this, $\sqrt{m}\max_i[x_i]_+ \geq \|[x]_+\|_2$, so that an ϵ-solution $(\rho_\epsilon, z_1^\epsilon = [s^\epsilon; v^\epsilon])$ to the GBSP problem induces the approximate solution $(\rho_\epsilon, w^\epsilon = J_n s^\epsilon, b^\epsilon)$ to (17) such that

$$\rho_\epsilon \geq \mathrm{Opt} \;\&\; \|[\rho_\epsilon\mathbf{1} - Y[Xw^\epsilon + b^\epsilon\mathbf{1}]]_+\|_2 \leq R + \sqrt{m}\rho_\epsilon\epsilon,$$

whence $(\rho_\epsilon(1-\sqrt{m}\epsilon), w^\epsilon, b^\epsilon)$ is a feasible solution to (17) with the value of the objective $\geq (1-\sqrt{m}\epsilon)\mathrm{Opt}$.

Finally, we associate with $z = [z_{11} = s; z_{12} = v; z_2 = u] \in Z = Z_1 \times Z_2$ a distribution P_z on $Z = Z_1 \times Z_2$ defined as follows. To generate $\zeta = [\zeta_{11}; \zeta_{12}; \zeta_2] \sim P_z$, we pick at random $\imath \in \{1, ..., 2n\}$, with $\mathrm{Prob}\{\imath = i\} = [z_{11}]_i$, $1 \leq i \leq 2n$, and set $\zeta_{11} = e_\imath$, e_i being the basic orths in $\mathbf{R}^{2n}$. We always set $\zeta_{12} = v$. To generate ζ_2, we act as follows. Let $I = \{i : y_i = 1\}$, $J = \{i : y_i = -1\}$, and let $p := \sum_{i\in I} u_i = \sum_{j\in J} u_j$ (recall that $\sum_i y_i u_i = 0$, that is, $\sum_{i\in I} u_i = \sum_{j\in J} u_j$). Note that $p \leq 1/2$ due to $\sum_{j=1}^m u_j \leq 1$. We first flip a coin with probability $1-2p$ to get head; if head appears, we set $\zeta_2 = 0$. If tail appears, we pick at random $\imath \in I$ with $\mathrm{Prob}\{\imath = i\} = u_i/p$, $i \in I$, pick at random $\jmath \in J$ with $\mathrm{Prob}\{\jmath = j\} = u_j/p$, $j \in J$, and set $\zeta_2 = \frac{1}{2}[e_\imath + e_\jmath]$, e_i being the basic orths in $\mathbf{R}^m$. It is immediately seen that P_z satisfies [P], and that it takes just $O(m+n)$ a.o. to generate a sample $\zeta \sim P_z$ and to compute the vector $F^\rho(\zeta)$.

3 Solving Bilinear Saddle Point Problem

We are about to present two randomized first order methods for solving BSPs and hence will be utilized in solving GBSPs — the *Stochastic Approximation* (SA) and the *Stochastic Mirror Prox* (SMP) algorithms, which are the randomized versions of the methods proposed in [11] and [12] respectively. Both SA and SMP are directly applicable to a BSP problem, and this is the situation we are about to consider here; the GBSP case will be considered in Section 4.

3.1 The Setup

Both SA and SMP algorithms are aimed at solving a BSP problem $(\mathcal{S})$. The setup for these methods is given by

- a norm $\|\cdot\|$ on the Euclidean space E where the domain $Z = Z_1 \times Z_2$ of $(\mathcal{S})$ lives, along with the conjugate norm $\|\zeta\|_* = \max_{\|z\|\leq 1}\langle\zeta, z\rangle$;

- a *distance-generating function* (d.g.f.) $\omega(z)$ which is convex and continuous on Z, admits continuous on the set $Z^o = \{z \in Z : \partial\omega(z) \neq \emptyset\}$ selection $\omega'(z)$ of subgradient (here $\partial\omega(x)$ is a subdifferential of $\omega\big|_Z$ taken at z), and is strictly convex with modulus 1 w.r.t. $\|\cdot\|$:
$$\forall z', z'' \in Z^o : \langle \omega'(z') - \omega'(z''), z' - z'' \rangle \geq \|z' - z''\|^2.$$
We shall refer to the latter property as to *compatibility* of $\omega(\cdot)$ and $\|\cdot\|$.

A d.g.f. ω gives rise to several important for us entities:

1. *Bregman distance* $V_z(u) = \omega(u) - \omega(z) - \langle \omega'(z), u - z\rangle$, where $z \in Z^o$ and $u \in Z$;

2. *Prox-mapping* $\mathrm{Prox}_z(\xi) = \mathrm{argmin}_{w\in Z}\{\langle \xi, w\rangle + V_z(w)\} : E \to Z^o$; here $z \in Z^o$ is a "prox center;"

3. "ω-center" $z_\omega = \mathrm{argmin}_{z\in Z}\,\omega(z) \in Z^o$ of Z and the quantities
$$\Omega = \max_{z\in Z} V_{z_\omega}(z) \leq \max_{z\in Z}\omega(z) - \min_{z\in Z}\omega(z),\ \ \Theta = \sqrt{2\Omega}. \tag{18}$$

In the sequel, we set
$$\mathcal{R} := \max_{z\in Z}\|z - z_\omega\| \leq \Theta, \tag{19}$$
where the concluding inequality follows from the fact that for every $z \in Z$ one has $\frac{1}{2}\|z - z_\omega\|^2 \leq V_{z_\omega}(z)$ by strong convexity of $\omega(\cdot)$. We also denote by $\mathcal{L}$ the ($\|\cdot\|$, $\|\cdot\|_*$)-Lipschitz constant of F:
$$\|F(z) - F(z')\|_* = \|\mathcal{A}(z - z')\|_* \leq \mathcal{L}\|z - z'\|, \quad \forall z, z'; \tag{20}$$
and set
$$M_* \;=\; \max_{z,z'\in Z}\|F(z) - F(z')\|_* \leq 2\mathcal{R}\mathcal{L} \leq 2\Theta\mathcal{L}, \tag{21}$$
$$F_* \;=\; \max_{z\in Z}\|F(z)\|_* \leq \|a\|_* + M_* \leq \|a\|_* + 2\Theta\mathcal{L}. \tag{22}$$

3.2 The SA and SMP Algorithms

Assume we have access to an "oracle" $\mathcal{O}$ which, at i-th call ($i = 1, 2, ...$), returns a vector $\xi_i \in E$ (this vector can be random with distribution depending on previous calls and, more generally, on the history of our computational process before the call). This oracle gives rise to two conceptual algorithms:
$$\begin{array}{ll} (a): & z_1 = z_\omega; \{z_t, \xi_t\} \mapsto \{z_{t+1} = \mathrm{Prox}_{z_t}(\gamma_t\xi_t), \xi_{t+1}\},\ t = 1, 2, ... \\ (b): & z_1 = z_\omega; \{z_t, \xi_{2t-1}\} \mapsto \{w_t = \mathrm{Prox}_{z_t}(\gamma_t\xi_{2t-1}), \xi_{2t}\} \mapsto \{z_{t+1} = \mathrm{Prox}_{z_t}(\gamma_t\xi_{2t}), \xi_{2t+1}\}, \\ & t = 1, 2, ... \end{array} \tag{23}$$
here γ_1, $\gamma_2, ...$ are positive stepsizes defined in a non-anticipative fashion, that is, γ_t depends on oracle's answers obtained prior to step t (i.e., γ_t depends solely on $\xi_1, ..., \xi_{t-1}$ in the case of (a), and solely on $\xi_1, ..., \xi_{2t-2}$ in the case of (b)). We refer to (23.a,b) as the Stochastic Approximation (SA) and Stochastic Mirror Prox (SMP) schemes, respectively. We will consider two implementations of these schemes, the *basic* and the *advanced* ones.

3.2.1 Basic implementation

Recall that we have associated with $(\mathcal{S})$ the affine operator $F(z) : Z \to E$ given by (3), and with every point $z \in Z$ — a probability distribution P_z supported on Z satisfying $\mathbf{E}_{\zeta\sim P_z}\{\zeta\} = z$. Suppose that

- the stepsizes $\gamma_t > 0$ are chosen in a non-anticipating fashion such that $\gamma_1 \geq \gamma_2 \geq ...$;
- in SA: ζ_t is drawn at random from the distribution P_{z_t}, and $\xi_t = F(\zeta_t)$;
- in SMP: $\xi_{2t-1} = F(\eta_t)$ with η_t drawn at random from the distribution P_{z_t}, and $\xi_{2t} = F(\zeta_t)$ with ζ_t drawn at random from the distribution P_{w_t}.

The approximate solution generated by the short-step SA/SMP in course of $t = 1, 2, ...$ steps is

$$z^t = t^{-1} \sum_{\tau=1}^{t} \zeta_\tau. \tag{24}$$

3.2.2 Advanced implementation

In Advanced implementation of SA and SMP, same as in the Basic one, the stepsizes $\gamma_t > 0$ still are chosen in a non-anticipating fashion, but the restriction $\gamma_1 \geq \gamma_2 \geq ...$ is now lifted. To explain how the oracle is built, observe that if $u \in Z$, then

$$\mathbf{E}_{\zeta\sim P_u}\{\langle F(\zeta), \zeta - u\rangle\} = 0$$

(recall that $F(z) = a + \mathcal{A}z$ with skew symmetric $\mathcal{A}$ and that $\mathbf{E}_{\zeta\sim P_u}\{\zeta\} = u$). It follows that given u and generating one by one independent samples $\eta^s \sim P_u$, $s = 1, 2, ...$, one with probability 1 eventually generates ζ such that

$$\langle F(\zeta), \zeta - u\rangle \leq 0. \tag{25}$$

At step t of SA, in order to define ξ_t, the oracle draws one by one samples $\eta^s \sim P_{z_t}$, $s = 1, 2, ...$, until a sample $\zeta_t := \eta^s$ satisfying (25) with $u = z_t$ is generated; when it happens, the oracle returns $\xi_t = F(\zeta_t)$. At a step t of SMP, the oracle is invoked twice, first to generate $\xi_{2t-1} = F(\eta_t)$, and then to generate $\xi_{2t} = F(\zeta_t)$. ξ_{2t-1} is generated exactly as in the basic implementation — by drawing a sample $\eta_t \sim P_{z_t}$ and returning $\xi_{2t-1} = F(\eta_t)$. To generate ξ_{2t}, the oracle draws one by one samples $\eta^s \sim P_{w_t}$, $s = 1, 2, ...$, until a sample $\zeta_t = \eta^s$ satisfying (25) with $u = w_t$ is generated; when it happens, the oracle returns $\xi_{2t} = F(\zeta_t)$.

Finally, in the advanced implementation we replace the rule (24) for generating approximate solutions with the rule

$$z^t = \frac{1}{\sum_{\tau=1}^{t} \gamma_\tau} \sum_{\tau=1}^{t} \gamma_\tau \zeta_\tau. \tag{26}$$

3.2.3 Quantifying quality of approximate solutions

Observe that by construction at a step τ both ζ_τ and $F(\zeta_\tau)$ become known. Recalling that F is affine, it follows that after t steps we have at our disposal both the approximate solution $z^t = [z_1^t; z_2^t]$ and the vector $F(z^t)$. As a result, with both Basic and Advanced implementations of both SA and SMP, after $t = 1, 2, ...$ steps we have at our disposal the quantities

$$\overline{\phi}(z_1^t) = \upsilon + \langle a_1, z_1^t\rangle + \max_{z_2\in Z_2}\langle z_2, -F_2(z_1^t)\rangle,\ \underline{\phi}(z_2^t) = \upsilon + \langle a_2, z_2^t\rangle + \min_{z_1\in Z_1}\langle z_1, F_1(z_2^t)\rangle \tag{27}$$

(see (3)) and consequently we know the residual $\epsilon_{\rm sad}(z^t) = \overline{\phi}(z^t) - \underline{\phi}(z^t)$ of the current approximate solution z^t. As we shall see in Section 4, this feature of our algorithms becomes instrumental when solving GBSP problems.[3] This is in sharp contrast with the prototypes of the SA and the SMP proposed, respectively, in [13, Section 3.3] and [9]. The approximate solutions z^t of those algorithms were computed according to the formula (26), but with z_τ [13] or w_τ [9] in the role of ζ_τ. As a result, in the prototype algorithms there is no computationally cheap way to quantify the quality of approximate solutions.

3.2.4 Efficiency estimates for Basic implementation

The accuracy bounds for Basic SA and SMP algorithms are given by the following

Proposition 3.1 *Let the BSP problem $(\mathcal{S})$ be solved by the short-step SA or SMP algorithm with positive stepsizes $\gamma_1 \geq \gamma_2 \geq ...$ chosen in a non-anticipative fashion. Then*
(i) *For every $t \geq 1$, for both SA and SMP one has*

$$\epsilon_{\rm sad}(z^t) \ \ \leq \ \ t^{-1}\left[\gamma_t^{-1}\Omega + R_t + S_t\right],\ R_t := \sum_{\tau=1}^t r_\tau,\ S_t := \sum_{\tau=1}^t s_\tau, \tag{28}$$

where

$$\begin{array}{rcl} r_t & = & \left\{\begin{array}{ll} \langle F(\zeta_t), \zeta_t - z_t\rangle & \text{in the case of SA,}\\ \langle F(\zeta_t), \zeta_t - w_t\rangle & \text{in the case of SMP,}\end{array}\right.\\ s_t & = & \left\{\begin{array}{ll} \langle F(\zeta_t), z_t - z_{t+1}\rangle - \gamma_t^{-1}V_{z_t}(z_{t+1}), & \text{in the case of SA,}\\ \langle F(\zeta_t), w_t - z_{t+1}\rangle - \gamma_t^{-1}V_{z_t}(z_{t+1}), & \text{in the case of SMP.}\end{array}\right.\end{array}$$

We have

$$s_t \leq \left\{\begin{array}{ll} \frac{\gamma_t}{2}\|F(\zeta_t)\|_*^2, & \text{in the case of SA,}\\ \frac{\gamma_t}{2}\|F(\zeta_t) - F(\eta_t)\|_*^2 - \frac{1}{2\gamma_t}\|w_t - z_t\|^2, & \text{in the case of SMP,}\end{array}\right. \tag{29}$$

with

$$s_t \leq \left\{\begin{array}{ll} \frac{\gamma_t}{2}F_*^2, & \text{in the case of SA,}\\ \frac{\gamma_t}{2}M_*^2, & \text{in the case of SMP.}\end{array}\right. \tag{30}$$

[3] Of course, computing the quantities in (27) is not completely costless; note, however, that the cost of this computation is dominated by the cost of computing the prox-mapping(s) at a step and thus is a small fraction of the overall computational effort.

In particular, if the stepsizes $\gamma_t > 0$ *satisfy* $S_t \leq \Omega/\gamma_t$, $t = 1, 2, ...$, *then*

$$\epsilon_{\rm sad}(z^t) \leq \frac{2\Omega}{t\gamma_t} + \frac{R_t}{t}. \tag{31}$$

(ii) *Further,* $\mathbf{E}\{R_t\} = 0$, *and in the case of SMP, under additional assumption that*

$$\gamma_t \leq (\sqrt{3}\mathcal{L})^{-1}, \tag{32}$$

we have

$$s_t \leq \frac{3\gamma_t}{2}\left[\|\mathcal{A}(\zeta_t - w_t)\|_*^2 + \|\mathcal{A}(\eta_t - z_t)\|_*^2\right], \tag{33}$$

so that $\mathbf{E}\{s_t\} \leq 3\gamma_t\sigma^2$, *where*

$$\sigma^2 = \sup_{z\in Z} \mathbf{E}_{\zeta\sim P_z}\left\{\|\mathcal{A}(\zeta - z)\|_*^2\right\} \leq M_*^2. \tag{34}$$

In particular, if the stepsizes $\gamma_t > 0$ *satisfy* $\mathbf{E}\{S_t\} \leq \Omega/\gamma_t$ *for* $t = 1$, 2, ..., *then*

$$\mathbf{E}\{\epsilon_{\rm sad}(z^t)\} \leq \frac{2\Omega}{t\gamma_t}.$$

The bound of Proposition 3.1 allows to easily conceive stepsize policies. Let us start with *offline* policies, where γ_t are chosen in advance deterministic reals. If the number of steps N is fixed in advance, one can use constant stepsizes $\gamma_1 = ... = \gamma_N = \gamma$. In particular, when choosing

$$\gamma = \left\{\begin{array}{ll} \frac{1}{F_*}\sqrt{\frac{2\Omega}{N}}\,, & \text{in the case of SA} \quad (a) \\ \min\left\{\frac{1}{\sigma}\sqrt{\frac{\Omega}{3N}}, \frac{1}{\sqrt{3}\mathcal{L}}\right\}, & \text{in the case of SMP} \quad (b) \end{array}\right. \tag{35}$$

(by (30), (22) this choice implies that $\mathbf{E}\{S_t\} \leq \Omega/\gamma_t$, $1 \leq t \leq N$), Proposition 3.1 implies the efficiency bound

$$\mathbf{E}\{\epsilon_{\rm sad}(z^N)\} \leq \left\{\begin{array}{ll} F_*\sqrt{\frac{2\Omega}{N}}, & \text{in the case of SA} \quad (a) \\ \max\left\{2\sigma\sqrt{\frac{3\Omega}{N}}, \frac{2\sqrt{3}\Omega\mathcal{L}}{N}\right\}, & \text{in the case of SMP} \quad (b) \end{array}\right. \tag{36}$$

When the number of steps is not fixed in advance, one can use the decreasing stepsizes

$$\forall t \geq 1, \;\; \gamma_t = \left\{\begin{array}{ll} \frac{1}{F_*}\sqrt{\frac{\Omega}{t}}\,, & \text{in the case of SA,} \\ \min\left\{\frac{1}{\sigma}\sqrt{\frac{\Omega}{6t}}, \frac{1}{\sqrt{3}\mathcal{L}}\right\}, & \text{in the case of SMP,} \end{array}\right. \tag{37}$$

which result in the accuracy bound

$$\forall t \geq 1, \;\; \mathbf{E}\{\epsilon_{\rm sad}(z^t)\} \leq \left\{\begin{array}{ll} 2F_*\sqrt{\frac{\Omega}{t}}\,, & \text{in the case of SA} \quad (a) \\ \max\left\{2\sigma\sqrt{\frac{6\Omega}{t}}, \frac{2\sqrt{3}\Omega\mathcal{L}}{t}\right\}, & \text{in the case of SMP} \quad (b) \end{array}\right. \tag{38}$$

completely similar to (36).

3.2.5 Online stepsize policies

From theoretical viewpoint, the main advantage of the outlined versions of SA and SMP with the "theoretically optimal" offline stepsize policies (35) and (37) are the explicit (and in fact — the best known under circumstances) efficiency estimates (36), (38). While they may appear attractive also from the practical viewpoint because of their apparent simplicity, their use may present several disadvantages: the quantity σ involved in the stepsize computation may not be available at hand and should be evaluated. Besides this, these policies are offline and worst-case oriented; we would prefer more flexible on line adjustable stepsizes.

A natural way to adjust the stepsizes online would be to choose at each step $t \geq 1$ the largest $\gamma_t \leq \gamma_{t-1}$ ensuring the balance $\Omega/\gamma_t \geq S_t$, and thus the bound (31). This idea cannot be implemented "as is," since the stepsize policy should be non-anticipative, while s_t is not yet available when γ_t is computed. This difficulty can be easily circumvented by using instead of s_t its a priori upper bound, which is either $\frac{\gamma_t}{2}F_*$ for the SA algorithm or $\frac{\gamma_t}{2}M_*^2$ for the SMP, see (29). Specifically, consider the online policy of choosing γ_t, $t \geq 1$ as follows:

$$\Omega\gamma_t^{-2} = \begin{cases} 2\sum_{\tau=1}^{t-1}\gamma_\tau^{-1}[s_\tau]_+ + F_*^2 & \text{in the case of SA,} \\ 2\sum_{\tau=1}^{t-1}\gamma_\tau^{-1}[s_\tau]_+ + 8\Omega\mathcal{L}^2 & \text{in the case of SMP,} \end{cases} \tag{39}$$

where we set $\sum_{\tau=1}^{0}\gamma_\tau^{-1}[s_\tau]_+ = 0$. With this policy, one clearly has $\gamma_1 \geq \gamma_2 \geq ....$

Proposition 3.2 *Let positive stepsizes γ_t, $t = 1, 2, ...$ of the Basic SA/SMP implementation be chosen according to (39). Then the approximate solution z^t satisfies*

$$\epsilon_{\rm sad}(z^t) \leq \frac{(1+\sqrt{2})\Omega}{t\gamma_t} + \frac{R_t}{t}. \tag{40}$$

As a consequence, we have

$$\epsilon_{\rm sad}(z^t) \leq \begin{cases} \frac{(1+\sqrt{2})\sqrt{\Omega}}{t}\left(F_*^2 + \sum_{\tau=1}^{t-1}\|F(\zeta_\tau)\|_*^2\right)^{1/2} + \frac{R_t}{t}, & \text{in the case of SA} \qquad (a) \\ \frac{(1+\sqrt{2})\sqrt{\Omega}}{t}\left(8\Omega\mathcal{L}^2 + \sum_{\tau=1}^{t-1}\varsigma_\tau\right)^{1/2} + \frac{R_t}{t} & \\ \qquad \leq \frac{7\Omega\mathcal{L}}{t} + \frac{R_t}{t} + \frac{(1+\sqrt{2})\sqrt{\Omega}}{t}\sqrt{\sum_{\tau=1}^{t-1}\varsigma_\tau}, & \text{in the case of SMP} \quad (b) \end{cases} \tag{41}$$

where

$$\varsigma_t = 3\left[\|F(\zeta_t) - F(w_t)\|_*^2 + \|F(\eta_t) - F(z_t)\|_*^2\right]. \tag{42}$$

Recalling that $\mathbf{E}\{R_t\} = 0$ and $\mathbf{E}\{\varsigma_t\} \leq 6\sigma^2$ (see (34)), we arrive at

Corollary 3.1 *Under the premise of Proposition 3.2, for the SMP algorithm one has*

$$\mathbf{E}\{\epsilon_{\rm sad}(z^t)\} \leq \frac{7\Omega\mathcal{L}}{t} + \frac{6\sqrt{\Omega}\sigma}{\sqrt{t}}. \tag{43}$$

Note that the bounds (41.a) and (43) within an absolute constant factor coincide with the respective bounds in (38), that is, our online stepsizes policy (which, in contrast to (37), does not require knowledge of σ) is not worse that the "theoretically optimal" stepsize policies underlying (38).

3.2.6 Discussion

Since $F_* \geq \mathcal{R}\mathcal{L} \geq \sigma/2$ (cf. (19)), the SA efficiency estimate (38.a) is at most within an absolute constant factor better than the corresponding estimate for the SMP. Besides this, the SMP bound (38.b) says that when the noise level σ of the oracle is small enough (specifically, $\sigma^2 = O\left(\frac{\Omega\mathcal{L}^2}{N}\right)$), then $\mathbf{E}\{\epsilon_{\rm sad}(z^t)\} \leq O(1)\frac{\Omega\mathcal{L}}{N}$, which, modulo expectation of the residual instead of the residual itself, coincides with the best known so far efficiency estimate of the deterministic first order algorithms solving bilinear saddle point problems. On the other hand, we do have a possibility to make σ small. The trivial way to do so is to use $P_z = \delta_z$, which results in $\sigma = 0$ and makes SMP a version of the Deterministic Mirror Prox algorithm (DMP) proposed in [12]. Another, more attractive, option to control σ is as follows. Given the family of distributions P_z supported on Z and such that $\mathbf{E}_{\zeta\sim P_z}\{\zeta\} = z$, and a positive integer k, we can convert P_z into the family of distributions $P_z^{(k)}$ with the same property as follows: in order to generate a random vector $\zeta \sim P_z^{(k)}$ and to compute $F(\zeta)$, we draw a k-element sample $\zeta^1, ..., \zeta^k$ from the distribution P_z, compute $F(\zeta^1), ..., F(\zeta^k)$ and then set $\zeta = \frac{1}{k}\sum_{i=1}^k \zeta^i$, so that

$$F(\zeta) = \frac{1}{k}\sum_{i=1}^k F(\zeta^i).$$

If, as in the examples of Section 2, drawing $\zeta^i \sim P_z$ and computing $F(z^i)$ is much cheaper than computing $F(z)$, the outlined procedure with a "reasonably large" value of k is still significantly cheaper than the direct computation of $F(z)$. At the same time, for "good enough" norms $\|\cdot\|_*$, passing from P_z to $P_z^{(k)}$ can significantly reduce the noise level σ. Specifically, given a norm $\|\cdot\|_*$ on a finite-dimensional Euclidean space E, one can associate with it its *regularity parameter* $\varkappa \geq 1$ (see Section 2.2, [8] for details) to ensure the following: whenever $k > 0$ is an integer and $\xi^1, ..., \xi^k$ are independent vectors from E with $\mathbf{E}\{\xi^i\} = 0$ and $\mathbf{E}\{\|\xi^i\|_*^2\} \leq \alpha_i^2$ and $\alpha = \max_i \alpha_i$, for $\xi = \frac{1}{k}\sum_{i=1}^k \xi_i$ it holds

$$\mathbf{E}\{\|\xi\|_*^2\} \leq \min\left[\frac{1}{k}, \frac{\varkappa}{k^2}\right]\sum_{i=1}^k \alpha_i^2 \leq \min\left[1, \frac{\varkappa}{k}\right]\alpha^2.$$

Suppose now that when running SMP we sample ζ_t, η_t from the distributions $P_z^{(k)}$ for some $k > 0$. It follows that if $\|\cdot\|_*$ is $\varkappa$-regular with certain $\varkappa$, then, passing from P_z to $P_z^+ = P_z^{(k)}$, we can reduce the "original" value of σ to the value $\sigma^+ = \min[1, \sqrt{\frac{\varkappa}{k}}]\sigma$. We shall see in a while that in the applications we have mentioned so far, $\varkappa$ is "small" — at most logarithmic in $\dim Z$. The bottom line is that there is a tradeoff between the computational cost of a call to a stochastic oracle and the noise level σ. Consequently, in the case of SMP, it is possible to tradeoff the computational effort per iteration and the iteration count to obtain an approximate solution of the desired expected quality, and we can use this tradeoff in order to save on the overall amount of computations. This option (which is the major advantage of SMP as compared to SA) is especially attractive when among the two components of our computational effort per iteration — one related to computing η_t, ζ_t, $F(\eta_t)$ $F(\zeta_t)$, and the other aimed at computing the prox mappings – the second component is essentially more significant than the first one. In such a situation, we basically can only gain by passing from P_z to $P_z^{(k)}$ with k chosen to balance the outlined two components of the computational effort.

3.2.7 Large deviations

In the above efficiency estimates, say, in (43), we upper-bounded the *expected* inaccuracy of approximate solutions z^t. In fact, one can get exponential upper bounds on probabilities of large deviations for the inaccuracy of the approximate solution. Though we do not need such bounds to access the inaccuracy of solutions, they are still useful to provide theoretical guarantees for the complexity of our algorithms (cf. Theorem 4.1 in the next section).

For the sake of definiteness, when presenting large deviation results, we restrict ourselves to the SMP algorithm and the stepsize strategy (39). Note that one can easily derive a deviation bound from the bound (43) on the expectation of $\epsilon_{\rm sad}(z^t)$ in the previous section. Indeed, let us fix the number t of iterations, run the algorithm m times and select the best, in terms of $\epsilon_{\rm sad}(\cdot)$, of the resulting approximate solutions. The probability that for this solution $\epsilon_{\rm sad}(\cdot)$ is worse than, say, twice the right hand side of (43) is at most 2^{-m} and thus can be made negligibly small with quite moderate values of m.

We also have the following bound on the deviations of the algorithm without restarts:

Proposition 3.3 *Assume we are solving problem* $(\mathcal{S})$ *by Basic implementation of SMP where* ζ_t, η_t *are sampled from the distributions* $P_z^{(k)}$, $k \geq 1$ *being a parameter of the construction. Assume also that the norm* $\|\cdot\|_*$ *is* $\varkappa$*-regular, and the online stepsize policy (39) is used. Then there are absolute constants* K_0, K_1 *such that the approximate solution* z^t *satisfies for all* $t \geq 1$ *and* λ, $\Lambda \geq 0$

$$\mathrm{Prob}\left\{\epsilon_{\rm sad}(z^t) \geq K_0\left[\frac{\Theta^2\mathcal{L}}{t} + \frac{\varkappa_*(k,\Lambda)\Theta^2\mathcal{L}}{\sqrt{kt}} + \Theta(\|a\|_* + \Theta\mathcal{L})\sqrt{\frac{\lambda}{kt}}\right]\right\} \leq e^{-\Lambda t} + e^{-\lambda}, \quad (44)$$

where $\varkappa_*(k,\Lambda) = \sqrt{\min[k,\,(\varkappa+\Lambda)]}$. *In particular, one has for all* $\varepsilon > 0$*:*

$$\begin{array}{l}\mathrm{Prob}\{\epsilon_{\rm sad}(z^N) \geq \varepsilon\} \leq e^{-\Lambda N} + e^{-\lambda} \quad \textit{for } N \geq N_\varepsilon, \textit{ where}\\ N_\varepsilon = K_1\mathrm{Ceil}\left(\max\left[\Theta^2\mathcal{L}\varepsilon^{-1},\ \frac{\varkappa_*^2(k,\Lambda)\Theta^4\mathcal{L}^2}{k\varepsilon^2},\ \frac{(\|a\|_*+\Theta\mathcal{L})^2\Theta^2\lambda}{k\varepsilon^2}\right]\right).\end{array} \quad (45)$$

3.3 Efficiency estimates for Advanced implementations of SA and SMP

The efficiency of Advanced implementations of SA and SMP stem from the following result (we use the notation from section 3.1):

Proposition 3.4 *Let the BSP problem $(\mathcal{S})$ be solved by the advanced-step SA or SMP algorithms. Then for every $t \geq 1$, for both SA and SMP one has*

$$\epsilon_{\rm sad}(z^t) \leq \Gamma_t^{-1}\left[\Omega + R_t + S_t\right] = \Gamma_t^{-1}\left[\Omega + \sum_{\tau=1}^{t} r_\tau + \sum_{\tau=1}^{t} s_\tau\right], \tag{46}$$

where

$$\begin{array}{rcl}
\Gamma_t &=& \sum\limits_{\tau=1}^{t}\gamma_\tau, \\
r_t &=& \left\{\begin{array}{ll} \gamma_t\langle F(\zeta_t), \zeta_t - z_t\rangle & \hbox{in the case of SA} \\ \gamma_t\langle F(\zeta_t), \zeta_t - w_t\rangle & \hbox{in the case of SMP} \end{array}\right. \\
s_t &=& \left\{\begin{array}{ll} \left[\gamma_t\langle F(\zeta_t), z_t - z_{t+1}\rangle - V_{z_t}(z_{t+1})\right], & \hbox{in the case of SA} \\ \left[\gamma_t\langle F(\zeta_t), w_t - z_{t+1}\rangle - V_{z_t}(z_{t+1})\right], & \hbox{in the case of SMP} \end{array}\right.
\end{array}$$

with $r_t \leq 0$ and

$$s_t \leq \left\{\begin{array}{ll} \frac{\gamma_t^2}{2}\|F(\zeta_t)\|_*^2 \leq \frac{\gamma_t^2}{2}F_*^2, & \hbox{in the case of SA} \\ \frac{\gamma_\tau^2}{2}\|F(\zeta_t) - F(\eta_t)\|_*^2 - \frac{1}{2}\|w_t - z_t\|^2 \leq \frac{\gamma_t^2}{2}M_*^2, & \hbox{in the case of SMP.} \end{array}\right. \tag{47}$$

In order to extract from (46) explicit efficiency estimates, we need to specify a stepsize policy. In this respect, the advanced implementations offer more freedom than the basic ones, since now we should not ensure neither the martingale property of the random sums R_t, nor the monotonicity of the stepsizes. One option here is to use constant stepsize policy

$$\gamma_t = \sqrt{\frac{2\Omega}{N}} \cdot \left\{\begin{array}{ll} \frac{1}{F_*}, & \hbox{in the case of SA} \\ \frac{1}{M_*}, & \hbox{in the case of SMP} \end{array}\right., \ 1 \leq t \leq N.$$

As it is easily seen, with this policy, (46) results in efficiency estimate (cf. (38))

$$\forall t \geq 1, \ \ \mathbf{E}\left\{\epsilon_{\rm sad}(z^t)\right\} \leq O(1)\left\{\begin{array}{lll} F_*\sqrt{\frac{\Omega}{t}}, & \hbox{in the case of SA} & (a) \\ \mathcal{R}\mathcal{L}\sqrt{\frac{\Omega}{t}}, & \hbox{in the case of SMP} & (b) \end{array}\right. \tag{48}$$

Our preliminary experiments, however, suggest to equip the advanced implementations of SA and SMP with the online stepsize policy as follows. Let us set

$$\delta_t = \frac{\Theta^2}{t}, \quad S_t^* = \sum_{\tau=1}^{t}\delta_\tau \quad [\leq \Theta^2(1 + \ln t)] \tag{49}$$

and let us choose γ_τ according to the "greedy" rule (the larger, the better) under the restriction that for all $t = 1, 2, ...$ it holds

$$R_t + S_t \leq S_t^*, \tag{$*_t$}$$

see (46). Specifically, assume that we have already carried out $t-1$ steps of the algorithm ensuring the relations $(*_\tau)$, $\tau \leq t-1$, and are about to define γ_t in order to carry out step t and to ensure $(*_t)$. At this time, we know $R_{t-1} \leq 0$ and S_{t-1}, same as know for sure that whatever be our choice of $\gamma_t > 0$, we would have

$$R_t - R_{t-1} = r_t \leq 0, \;\; S_t - S_{t-1} = s_t \leq \theta\gamma_t^2, \;\; \theta = \begin{cases} \frac{F_*^2}{2}, & \text{in the case of SA} \\ \frac{M_*^2}{2} \leq 2\mathcal{L}^2\mathcal{R}^2, & \text{in the case of SMP} \end{cases}$$

(see (47)). Thus, we can be sure that $S_t + R_t \leq [S_{t-1} + R_{t-1}] + \theta\gamma_t^2$, meaning that when choosing

$$\gamma_t = \sqrt{[S_t^* - S_{t-1} - R_{t-1}]/\theta} \tag{50}$$

we guarantee the validity of $(*_t)$ and the inequality $\gamma_t \geq \sqrt{\delta_t/\theta}$. This observation combined with (46) and $(*_N)$ implies that

$$\begin{array}{rcl} \forall N \geq 1: \quad \epsilon_{\rm sad}(z^N) & \leq & \frac{\Theta^2/2 + R_N + S_N}{\sum_{\tau=1}^N \sqrt{\delta_\tau/\theta}} \leq \frac{O(1)\Theta^2(1+\ln N)}{\sum_{\tau=1}^t \sqrt{\delta_\tau/\theta}} \\ & \leq & O(1)(1+\ln N) \cdot \begin{cases} \Theta F_* N^{-1/2}, & \text{in the case of SA,} \\ \Theta\mathcal{R}\mathcal{L}N^{-1/2}, & \text{in the case of SMP.} \end{cases} \end{array} \tag{51}$$

Observe that (51) is, within the logarithmic in N factor $O(1)(1+\ln N)$, the same as the bound (48). In fact, we could somehow reduce this logarithmic gap by modifying s_t^*, but we do not think this is necessary; we may hope (and the experiments to be reported in Section 5 fully support this hope) that "in reality" the rule (50) is much better than it is stated by the above worst-case analysis. The rationale behind this hope is that while we indeed are conservative when thinking how large could $S_t - S_{t-1}$ be, we account, to some extent, for the "past conservatism:" when $S_{t-1} + R_{t-1}$ is essentially less than S_{t-1}^*, γ_t as given by (50) is essentially larger than its lower bound used in the complexity analysis.

Finally, we remark that the major theoretical disadvantage of the efficiency estimate (51) as compared to (38) is much more serious than an extra log-factor. While with the basic implementation, in course of N steps the stochastic oracle is called $O(1)N$ times, the number of oracle calls in course of N steps of the advanced implementation is random and can be much larger than $O(1)N$; it is unclear why it should be $O(1)N$ even on average. Though for the time being we cannot support the empirical evidence by a solid theoretical complexity analysis, in our experiments the advanced implementation by far outperformed its basic counterpart.

3.4 The Favorable Geometry Case

We are about to present the "favorable geometry" case where we can point out the setup for SA/SMP which results in (nearly) *dimension-independent* efficiency estimates. Specifically, assume that

[G.1] The domain Z of $(\mathcal{S})$ is a *subset* of the direct product $Z^+ = B_1 \times .... \times B_{p+q}$ of $r = p+q$ "standard blocks" as follows:

- for $1 \leq i \leq p$, B_i is the unit Euclidean ball in $F_i = \mathbf{R}^{n_i}$;

- for $1 \leq j \leq q$, B_{p+j} is a subset of the space F_{p+j} of $n_{p+j} \times n_{p+j}$ ($n_{p+j} > 1$) symmetric block-diagonal matrices of a given block-diagonal structure and is the *spectahedron* of F_{p+j}, that is, the set of all positive semidefinite matrices from F_{p+j} with unit trace.

  In particular, B_{p+j} can be the standard simplex $\{x \in \mathbf{R}^k_+ : \sum_\ell x_\ell = 1\}$ (since the space of diagonal $k \times k$ matrices can be naturally identified with $\mathbf{R}^k$).

We equip $F_i = \mathbf{R}^{n_i}$, $i \leq p$, with the standard Euclidean structure and the associated Euclidean norm $\|\cdot\|_{(i)}$, and F_{p+j} – with the Frobenius Euclidean structure and the trace-norm (the sum of singular values of a matrix) $\|\cdot\|_{(p+j)}$. In particular, the embedding space $E = F_1 \times ... \times F_r$ of Z^+ becomes equipped with the direct product of the indicated Euclidean structures. Note that the norm $\|\cdot\|_{(i,*)}$ conjugate to $\|\cdot\|_{(i)}$ is either the norm $\|\cdot\|_{(i)}$ itself (this is so when $i \leq p$), or is the standard matrix norm (maximal singular value of a matrix) (this is so when $i > p$). We denote a vector form on E as $x = [x_1; ...; x_r]$, where x_ℓ is the F_ℓ-component of x.

G.2. The decomposition $Z = Z_1 \times Z_2 \subset E_1 \times E_2$ is compatible with the decomposition $Z = B_1 \times ... \times B_r$, that is, E_1 is the direct product of some of F_ℓ, $1 \leq \ell \leq p+q$, and E_2 is the direct product of the remaining F_ℓ. Besides this, we assume that Z intersects the relative interior of Z^+.

We refer to this case as to the one of *favorable geometry* and associate with this case the setup for SA and SMP as follows (cf. [12, Section 5]):

- The skew-symmetric linear mapping $\mathcal{A}$ (see (3)) can be written down as

$$\mathcal{A}[x_1; ...; x_r] = [\sum_{j=1}^r A^{1j} x_j; ...; \sum_{j=1}^r A^{rj} x_j],$$

  where A^{ij} is a linear mapping from F_j to F_i and $[A^{ij}]^* = -A^{ji}$. We denote by L_{ij} an a priori upper bound on $L^*_{ij} := \max_{x_j} \left\{ \|A^{ij} x_j\|_{(i,*)} : \|x_j\|_{(j)} \leq 1 \right\}$ such that $L_{ij} = L_{ji}$.[4]

- Further, we set

$$\begin{array}{l} \omega_i(x_i) = \frac{1}{2} x_i^T x_i : \; B_i \to \mathbf{R},\, \Omega_i = \frac{1}{2},\, 1 \leq i \leq p \\ \omega_{p+j}(x_{p+j}) = 2 \sum_{\ell=1}^{n_{p+j}} \lambda_\ell(x_{p+j}) \ln(\lambda_\ell(x_{p+j})) : B_{p+j} \to \mathbf{R},\, \Omega_{p+j} = 2\ln(n_j),\, 1 \leq j \leq q \end{array}$$

  where $\lambda_\ell(u)$ are the eigenvalues of a symmetric matrix u taken with their multiplicities. It is known that $\omega_\ell(\cdot)$ is a d.g.f. for B_ℓ compatible with the norm $\|\cdot\|_\ell$, $1 \leq \ell \leq r$.

- Finally, we define the norm $\|\cdot\|$ on E and the d.g.f. $\omega(\cdot)$ for Z according to

$$\mu_\ell = \frac{1}{\Omega_\ell} \frac{\sum_{j=1}^r L_{\ell j} \sqrt{\Omega_\ell \Omega_j}}{\sum_{i,j=1}^r L_{ij} \sqrt{\Omega_i \Omega_j}},\; \|[x_1; ...; x_r]\| = \sqrt{\sum_{\ell=1}^r \mu_\ell \|x_\ell\|^2_{(\ell)}},\; \omega(x) = \sum_{\ell=1}^r \mu_\ell \omega_\ell(x_\ell), \tag{52}$$

[4] The latter restriction is natural, since $L^*_{ij} = L^*_{ji}$ due to $[A^{ij}]^* = -A^{ji}$.

which results in

$$\Omega \leq 1,\ \mathcal{R} \leq \Theta \leq \sqrt{2},\ \mathcal{L} = \sum_{i,j=1}^{r} L_{ij}\sqrt{\Omega_i\Omega_j}, \tag{53}$$

see [12, Section 5].

Remark 3.1 *From the results of [8] it follows that the norm* $\|\xi\|_* = \sqrt{\sum_{\ell=1}^r \mu_\ell^{-1}\|\xi_\ell\|_{(i,*)}^2}$ *is* $\varkappa$*-regular (see discussion in Section 3.2.4) with nearly dimension-independent* $\varkappa$*, namely,* $\varkappa = 3\max\limits_{1\leq j\leq q}\ln(n_{p+j})$.

Note that the applications presented in Sections 2.2.2 and 2.2.3 are of favorable geometry; the same is true for the low dimension approximation problem of Section 2.1.3 after passing from the variable Q to the variable $R = d^{-1}Q$.

4 Solving the Generalized Bilinear Saddle Point Problem

Here we explain how a GBSP problem (9) – (10) can be reduced to a "small series" of BSP problems; the strategy to follow originates from [10]. From now on we assume, in addition to **A.1-2**, that we have an a priori upper bound $\bar{\rho}$ on the optimal value ρ_* of (10). For example, it is immediately seen that when finding an ϵ-solution to ℓ_1 minimization problem with ℓ_p fit (Section 2.2.2) in the only nontrivial case $\|b\|_p > \delta$ relation (13) implies that

$$\bar{\rho} := \frac{\|A\|_{1,p}}{\|b\|_p - \delta} \geq \rho_* := \frac{1}{\mathrm{Opt}},\ \|A\|_{1,p} = \max_j \|A_j\|_p, \tag{54}$$

where $A_1, ..., A_n$ are the columns of A. In particular, when finding an ϵ-solution to ℓ_1 minimization problem with the uniform fit in the only nontrivial case $\|b\|_\infty > \delta$ we have

$$\bar{\rho} := \frac{\|A\|_{1,\infty}}{\|b\|_\infty - \delta} \geq \rho_* := \frac{1}{\mathrm{Opt}},\ \|A\|_{1,\infty} = \max_{i,j} |A_{ij}|; \tag{55}$$

.

For the sake of definiteness, we assume that we are in the Favorable Geometry case, and that the decomposition $Z = Z_{11} \times Z_{12} \times Z_2 \subset E$, see (11), is compatible with the decomposition $E = F_1 \times ... \times F_r$, that is, the embedding spaces of Z_{11}, Z_{12} and Z_2 are products of some of F_ℓ's. To save space, we restrict ourselves with the SMP algorithm; modifications in the case of SA are straightforward.

The algorithm solves the problem of interest (10) by applying to $\mathrm{SV}(\cdot)$ a Newton-type root finding routine, with (approximate) first order information on SV at a point ρ given by SMP as applied to the saddle point problem specifying $\mathrm{SV}(\rho)$. Specifically, the algorithm works stage by stage. At a stage s, we have at our disposal an upper bound ρ_s on ρ_* and a piecewise linear function $\ell_{s-1}(\rho)$ which underestimates $\mathrm{SV}(\cdot)$:

$$\mathrm{SV}(\rho) \geq \ell_{s-1}(\rho)\ \ \forall \rho \geq 0.$$

here $\rho_1 = \bar{\rho}$, $\ell_0 \equiv -\infty$. At a stage, we apply SMP to the BSP problem

$$\mathrm{SV}(\rho_s) = \min_{z_1 \in Z_1} \max_{z_2 \in Z_2} \phi^{\rho_s}(z_1, z_2) \tag{$\mathcal{S}_s$}$$

namely, act as follows.

A. We start stage s with building the setup for SMP as explained in Section 3.4. The affine operator associated with $(\mathcal{S}_s)$ is

$$\begin{array}{rcl} F^{\rho_s}(z_1 = [z_{11}; z_{12}], z_2) & = & \Phi(z_1, z_2) + \rho_s \Psi(z_1, z_2) \\ & = & [[a_{11} + B^* z_2; \rho_s(a_{12} + C^* z_2)]\,; -b - Bz_{11} - \rho_s(c + Cz_{12})]\,, \end{array}$$

see (9), (11). In matrix $\mathcal{A} = \mathcal{A}_s$ of the linear part of F^{ρ_s}, some blocks A^{ij} are independent of ρ_s, while the remaining blocks are proportional to ρ_s. Consequently, the Lipschitz constant of F^{ρ_s} as given by (53) is

$$\mathcal{L} = \mathcal{L}(\rho_s) = \mathcal{M} + \rho_s \mathcal{N},\ \mathcal{M}, \mathcal{N} \geq 0. \tag{56}$$

An analogous decomposition holds for vector $a = a_s$:

$$\|a\|_* = \mu + \rho_s \nu,\ \mu, \nu \geq 0.$$

B. We apply to $(\mathcal{S}_s)$ either the basic, or the advanced implementation of the SMP. When running the basic SMP, we use the distributions $P_z^{(k)}$, see Section 3.2.4 (here $k \geq 1$ is a parameter of the construction) and use the online stepsize policy (39), where we set $\mathcal{L} = \mathcal{M} + \rho_s \mathcal{N}$ and $\Omega = 1$ (see (53)). When $(\mathcal{S}_s)$ is solved by the advanced SMP, we use the online stepsize policy (49) – (50), with $\Theta = \sqrt{2}$ in (49).
B.1. Let $z^{ti} = [z_1^{ti}, z_2^{ti}]$ be the approximate solution to $(\mathcal{S}_s)$ generated after t steps of stage s; recall that along with this solution, we have at our disposal the quantities

$$\begin{array}{rcl} \overline{\phi}^{ts} & = & \max_{z_2 \in Z_2} \phi^{\rho_s}(z_1^{ts}, z_2) = \upsilon + \langle a_{11}, z_{11}^{ts} \rangle + \rho_s[\chi + \langle a_{12}, z_{12}^{ts} \rangle] \\ & & \qquad\qquad + \min_{z_2 \in Z_2} \langle z_2, b + \rho_s c + B z_{11}^{ts} + \rho C z_{12}^{ts} \rangle, \\ \underline{\phi}^{ts} & = & \min_{z_1 \in Z_1} \phi^{\rho_s}(z_1, z_2^{ts}) = \overbrace{\upsilon + \langle b, z_2^{ts} \rangle + \min_{z_{11} \in Z_{11}} \langle a_{11} + B^* z_2^{ts}, z_{11} \rangle}^{p_{ts}} \\ & & \qquad + \rho_s \overbrace{\left[\kappa + \langle c, z_2^{ts} \rangle + \min_{z_{12} \in Z_{12}} \langle a_{12} + C^* z_2^{ts}, z_{12} \rangle\right]}^{q_{ts}} \end{array} \tag{57}$$

(cf. (27) and see (9), (11)). We set

$$u^{ts} = \min_{\tau \leq t} \overline{\phi}^{\tau s},\ \ell^{ts} = \max_{\tau \leq t} \underline{\phi}^{\tau s},\ \ell_{ts}(\rho) = \max[\ell_{s-1}(\rho), \max_{1 \leq \tau \leq t} [p_{\tau s} + q_{\tau s}\rho]].$$

Note that u^{ts} is a nonincreasing in t upper bound on $\mathrm{SV}(\rho_s)$, ℓ^{ts} is a nondecreasing in t lower bound on $\mathrm{SV}(\rho_s)$, and $\ell_{ts}(\rho)$ underestimates $\mathrm{SV}(\rho)$ for all $\rho \geq 0$. In addition, $\ell_{ts}(\rho_s) \geq \ell^{ts}$. Note also that after t steps we have at our disposal vectors $w_1^{ts} \in Z_1$, $w_2^{ts} \in Z_2$ such that

$$\max_{z_2 \in Z_2} \phi^{\rho_s}(w_1^{ts}, z_2) = u^{ts} \leq \overline{\phi}^{ts},\ \min_{z_1 \in Z_1} \phi^{\rho_s}(z_1, w_2^{ts}) = \ell^{ts} \geq \underline{\phi}^{ts},$$

meaning that $w^{ts} = [w_1^{ts}; w_2^{ts}]$ is a feasible solution to $(\mathcal{S}_s)$ and $\epsilon_{\rm sad}(w^{ts}) = u^{ts} - \ell^{ts} \leq \overline{\phi}^{ts} - \underline{\phi}^{ts} = \epsilon_{\rm sad}(z^{ts})$.

B.2. We proceed with solving $(\mathcal{S}_s)$ until one of the following two situations occurs:
A) We get $u^{ts} \leq \epsilon\rho_s$. In this case we terminate with the claim that ρ_s, w_1^{ts} is the desired ϵ-solution to (9) – (10).
B) We get $\ell^{ts} \geq \frac{3}{4}u^{ts}$. When it happens, we set

$$\rho_{s+1} = \max\left\{\rho : \ell_{ts}(\rho) \leq 0\right\}, \; \ell_s(\cdot) \equiv \ell_{ts}(\cdot) \tag{58}$$

and pass to the stage $s+1$.

Theorem 4.1 *When solving a Generalized Bilinear Saddle Point problem* (9) – (10) *by the outlined algorithm:*

(i) *The algorithm terminates in finite time with probability 1, and the resulting solution is an ϵ-solution, as defined in Section 2.2, to the GBSP problem in question;*

(ii) *The number of stages does not exceed the quantity* $O(1)\ln\left(\frac{\|\phi\|_\infty + \bar{\rho}\|\psi\|_\infty}{\epsilon\rho_*} + 2\right)$, *where* $\|\phi\|_\infty = \max_{z\in Z}|\phi(z)|$, $\|\psi\|_\infty = \max_{z\in Z}|\psi(z)|$, *see* (9).

(iii) *The (random) number N_s of steps at every stage s of the basic implementation satisfies for all $\varepsilon > 0$ the relation*

$$\mathrm{Prob}\{N_s \geq N(\epsilon)\} \leq e^{-\Lambda N(\epsilon)} + e^{\lambda},$$

where

$$N(\epsilon) = O(1)\mathrm{Ceil}\left[\frac{\mathcal{M}+\rho_*\mathcal{N}}{\epsilon\rho_*} + \frac{\varkappa_*(k,\Lambda)^2}{k}\left(\frac{\mathcal{M}+\rho_*\mathcal{N}}{\epsilon\rho_*}\right)^2 + \frac{\lambda}{k}\left(\frac{\mu+\rho_*\nu}{\epsilon\rho_*}\right)^2\right]. \tag{59}$$

The number of steps at every stage of the advanced implementation of the algorithm does not exceed

$$N_{\rm adv}(\epsilon) = O(1)\left[\frac{\mathcal{M}+\rho_*\mathcal{N}+2\epsilon\rho_*}{\epsilon\rho_*}\ln\left(\frac{\mathcal{M}+\rho_*\mathcal{N}+2\epsilon\rho_*}{\epsilon\rho_*}\right)\right]^2. \tag{60}$$

For proof, see Section A.4.

In the case of ℓ_1 minimization problems with uniform and ℓ_2 fits, Theorem 4.1 as applied to the basic implementation of SMP with $k = 1$, initialized according to (55), resp., (54), after completely straightforward computations implies the complexity bounds stated in Proposition 1.1. The preprocessing mentioned in item (ii) of Proposition is as follows: we choose an $m \times m$ orthogonal matrix U with moduli of entries not exceeding $O(1)/\sqrt{m}$ and such that multiplication of a vector by U takes $O(m\ln m)$ operations (e.g., U can be the matrix of the Cosine Transform). We then draw at random a ± 1 vector ξ from the uniform distribution on the vertices of the unit m-dimensional box and pass from the data $[A, b]$ to the data

$$[A' = U\mathrm{Diag}\{\xi\}A, \;\; b' = U\mathrm{Diag}\{\xi\}b],$$

thus obtaining an equivalent reformulation of the problem of interest. Note that this preprocessing costs $O(1)mn\ln(m)$ operations. We clearly have $\|A'\|_{1,2} = \|A\|_{1,2}$. Applying the Hoeffding inequality, it is immediately seen that with probability $\geq 1-\chi$ one has $\|A'\|_{1,\infty} < O(1)\sqrt{\ln(mn/\chi)}m^{-1/2}\|A\|_{1,2}$, that is, $\Gamma(A') \leq O(1)\sqrt{\ln(mn/\chi)}$, as stated in Proposition 1.1.

5 Numerical Results

Below we report on a series of numerical experiments aimed at comparing the performances of the Stochastic Mirror Prox algorithm SMP (in its advanced implementation) and its prototype — Deterministic Mirror Prox algorithm (DMP) proposed in [12][5]. The algorithms were tested on the GBSP problems of ℓ_1 minimization with uniform and ℓ_2 fits, see Section 2.2.2.

Test problems we use are of the "Compressive Sensing" origin. Specifically, given the sizes m, n of a test problem, we picked at random an $m \times n$ matrix B with i.i.d. entries taking values ± 1 with probabilities 0.5, and a sparse (with $\mathrm{Ceil}(\sqrt{m})$ nonzero entries) "true signal" x_* normalized to have $\|x_*\|_1 = 1$, thus giving rise to the test problem

$$\mathrm{Opt}_p = \min_x \left\{ \|x\|_1 : \|Ax - y\|_p \leq \delta \right\},\ A = m^{-1/p}B,\ y = Ax_* + \xi \qquad (P_p)$$

where $p = \infty$ (uniform fit) or $p = 2$ (ℓ_2 fit). The "observation noise" ξ was chosen at random and then normalized to have $\|\xi\|_p = \delta$. Our goal is to solve (P_p) within accuracy ϵ, i.e., to find x_ϵ satisfying $\|x_\epsilon\|_1 \leq \mathrm{Opt}_p$ and $\|Ax_\epsilon - y\|_p \leq \delta + \epsilon$. In all our experiments, $\delta = 0.005$ and $\epsilon = 0.0025$ were used.

Implementation of the algorithms. The GBSP reformulations of problems (P_p) were solved by SMP (in advanced implementation) and DMP according to the scheme presented in Section 4. In the case $p = \infty$ of uniform fit, both SMP and DMP used the GBSP problem reformulation given by (16). In the case $p = 2$ of ℓ_2 fit, SMP used the GBSP reformulation (15), while DMP was applied to the GBSP problem stemming directly from (14) with $p = 2$, namely, given by

$$\phi^\rho(z_1, z_2) = z_2^T(AJ_n z_1 - \rho b) - \rho\delta, Z_1 = Z_{11} = \Delta_{2n},\ Z_2 = \{\|z_2\|_2 \leq 1\}. \qquad (61)$$

The rationale here is that the GBSP given by (61) "by itself" is easier than the GBSP given by (15): an ϵ-solution to the latter problem induces straightforwardly an ϵ-solution to the former one, but not vice versa. As a compensation, the problem (15), in contrast to (61), is better suited for randomization[6]. The latter fact, which is crucial for SMP, is irrelevant for DMP, this is why we apply this algorithm to the GBSP given by (61). In order to make a fair comparison, when running SMP for ℓ_2-fit, we terminate the run based on the ℓ_2-residual of the solution.

In our implementations, we have tested different policies for choosing the starting point at each stage and different choices of the distance generating function (d.g.f.) for

[5] DMP is nothing but SMP with precise information (i.e., P_z is the unit mass sitting at z) and on-line stepsize policy described in [12, Section 6].

[6] Indeed, in the second problem all nontrivial matrix-vector multiplications required to compute $F^\rho(z)$ are multiplications of vectors from the ℓ_1-balls by A and A^T; since a vector from ℓ_1-ball is the expectation of an extremely sparse (just one nonzero entry) random vector taking values in the same ball, the required matrix-vector multiplications admit cheap randomized versions. In the first problem, some of the required matrix-vector multiplications involve vectors from the $\|\cdot\|_2$-ball, and such a vector typically cannot be represented as the expectation of a sparse random vector taking values in the ball.

the simplexes. Specifically, along with the entropy d.g.f. discussed in Section 3.4, we tested the power d.g.f. $\omega(x) = \frac{\mathrm{e}}{\kappa(1+\kappa)}\sum_{i=1}^n x_i^{1+\kappa} : \{x \in \mathbf{R}_+^n : \sum_i x_i \leq 1\} \to \mathbf{R}$, with $\kappa = \frac{1}{\ln(n)}$; the theoretical complexity bounds associated with this choice of d.-g.f. coincide, within absolute constant factors, with those for the entropy. The best policies we ended up with are as follows:
— for SMP: entropy d.g.f., restarts from the ω-center of Z ("C00E" implementation);
— for DMP, in the case of uniform fit: power d.g.f., restarts from the convex combination of the best (with the smallest ϵ_{sad}) point found so far and the ω-center of Z, the weights being 0.75 and 0.25, respectively ("B75P" implementation);
— for DMP, in the case of ℓ_2-fit: power d.g.f., restarts from the convex combination of the last search point of the previous stage and the ω-center of Z, the weights being 0.25 and 0.75, respectively ("L25P" implementation).

When implementing SMP, we utilized the option, discussed in Section 3.2.4, of building an estimate $F(\zeta)$ of $F(z)$ by generating k samples $\zeta^\ell \sim P_z$, $\ell = 1, ..., k$, and setting $\zeta = \frac{1}{k}\sum_{\ell=1}^k \zeta^\ell$. The "multiplicity" k was set to 40 for small instances and 100 for large (those with at least 10^8 nonzeros in A) instances.

The MATLAB 7.10.0 implementation of the algorithms was executed on an eight-core machine with two quad-core Intel Xeon E5345 CPU@2.33GHz, 8 MB L2 cache per quad-core chip and 12GB FB-DIMM total RAM (the computations were running single-core and single-threaded).

The results, I. In order to avoid too time-consuming experimentation, we primarily dealt with "moderate size" test problems. These problems were split into four groups according to the total number of nonzeros in A ($2 \cdot 10^6$, $8 \cdot 10^6$, $32 \cdot 10^6$, $128 \cdot 10^6$). Every group was further split into two subgroups according to the ratio $n : m$ (8 and 2). For every one of the resulting pairs (m, n), we generated 5 instances of problem (P_2) and 5 instances of problem (P_∞) and solved them by DMP and SMP. Thus, the methods were compared on totally 70 problems split into 14 series of 5 experiments each, with common for all experiments of a series sizes m, n and the value of p. The results are presented in Tables 1 (uniform fit) and 2 (ℓ_2 fit). For every series of 5 experiments, we present the corresponding minimal, maximal and average values of several performance characteristics, specifically

- `CPU` — the CPU time (sec) of the entire computation
- `Calls` — the total number of computations of the values of F
- `FCalls` — the equivalent number of calls to the deterministic oracle for the randomized algorithm. This quantity is defined as follows. For DMP, computing a value of F at a point reduces to a pair of matrix-vector multiplications, one involving A and the other one involving A^T; the cost of this computation is $2mn$ operations. For SMP invoked with multiplicity k (see above), the computation of (an unbiased estimate of) $F(z)$ requires multiplying one vector with $\leq k$ nonzero entries by A, and another vector with $\leq k$ nonzero entries by A^T, the total cost of these two computations being $k(m+n)$ operations. Thus, the "deterministic equivalent" of the randomized computation of F used by SMP is $\frac{k(m+n)}{2mn}$. The quantity `FCalls` is the induced by this definition deterministic equivalent of all randomized computations of F in a run of the SMP.

The data in Tables 1, 2 suggest the following interpretations:

1. As the sizes of instances grow, the randomized algorithm eventually outperforms its deterministic counterpart in terms of the CPU time, and the corresponding "savings" grow with the size $m \times n$ of the instance, and for instances of a given size – grow as the ratio n/m decreases. Both phenomena are quite natural: the larger is mn and the smaller is $n/m \geq 1$ for a given mn, the smaller is the deterministic equivalent $k\frac{m+n}{2mn}$ of a randomized computation of F.

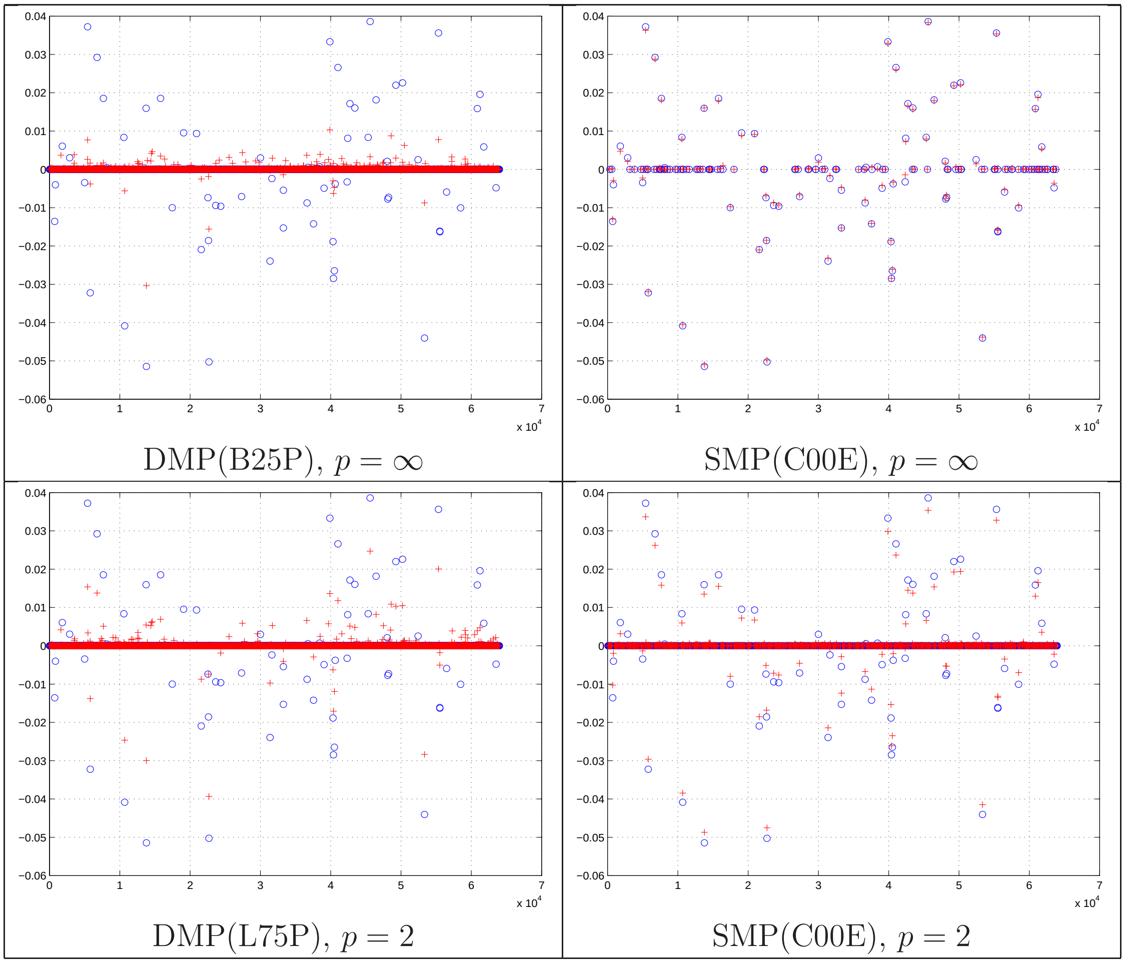
2. Even for our "not too large" test problems, the savings stemming from randomization can be quite significant: for the 8000×16000 instances, SMP is, at average, nearly 4.6 times faster than the best version of DMP for problems with uniform fit and 2.0 times faster than DMP for problems with ℓ_2 fit.

   When interpreting the CPU time data one should keep in mind that oracle calls of DMP make use of very efficient MATLAB implementation of matrix-vector multiplication, while SMP relies upon much less efficient (with respect to, e.g., C language) implementation of long DO loops.

3. The advantages, if any, of SMP as compared to DMP are more significant in the case of uniform fit than in the case of ℓ_2 fit. This phenomenon is quite natural: as we have already explained, in the case of ℓ_2 fit the methods are applied to different GBSP reformulations of (P_2), and the reformulation DMP works with is easier than the one processed by SMP.

The results, II. In order to get impression of what happens when the matrix A in (P_p) is too large to be stored in RAM, we carried out two experiments where the goal was to solve the ℓ_1 minimization problem with uniform and with ℓ_2 fits and fully dense $(m = 32000) \times (n = 64000)$ matrix A given by a simple analytical expression. This expression allows to compute a column/a row of A with a given index in $O(m)$, resp., $O(n)$ operations. Matrix $A = A_p$ was normalized to have $\|A\|_{1,p} = 1$. While the sizes of A make it impossible to store the matrix in the RAM of the computer we used for the experiments, we still can multiply vectors by A and A^T by computing all necessary columns and rows, and thus can run DMP and SMP. In our related experiments, we generated at random a sparse (64 nonzeros) "true" signal $x_* \in \mathbf{R}^{64000}$ with $\|x_*\|_1 = 1$, computed $y = Ax + \xi$, ξ, $\|\xi\|_p = \delta = 0.005$, being observation noise, and ran DMP and SMP in order to find an ϵ-solution x_ϵ, $\epsilon = 0.0025$, to the resulting problem (P_p); in particular, we should have $\|x_\epsilon\|_1 \leq \|x_*\|_1 = 1$ and $\|Ax_\epsilon - b\| \leq \delta + \epsilon = 0.0075$. In every experiment, each of the methods was allowed to run at most 7,200 sec. The results are as follows.

- In the allowed 7,200 sec, the deterministic algorithms on every one of the two test problems ($p = 2$ and $p = \infty$) was able to carry out just about 30 steps with the total of about 67 computations of $F(\cdot)$; this is by far not enough to get meaningful results, see Table 3. In contrast to this, the numbers of steps and randomized computations of F carried out by the randomized algorithm in the same 7,200 sec was in the range



DMP(B25P), $p = \infty$

SMP(C00E), $p = \infty$

DMP(L75P), $p = 2$

SMP(C00E), $p = 2$

Figure 1: DMP-based (left) and SMP-based (right) recovery of sparse signals in the $32,000 \times 64,000$ experiment, entries vs. their indexes. Circles: x_*; crosses: recovery.

of tens of thousands, which was enough to fully achieve the required accuracy for both $p = \infty$ and $p = 2$.

- While the quality of approximation of x_* by the solution yielded by DMP is basically nonexisting, the SMP produced fairy reasonable approximations of x_*, see Table 3 and Figure 1.

In our opinion, the preliminary numerical results we have reported suggest that "acceleration via randomization" possesses a significant practical potential when solving extremely large-scale convex programs of appropriate structure.

Table 1: Numerical Results for ℓ_1-minimization with $\| \cdot \|_\infty$-fit

		DMP		SMP				
Sizes		Calls	CPU	Calls	FCalls	CPU	$\frac{\text{Calls,DMP}}{\text{FCalls,SMP}}$	$\frac{\text{CPU,DMP}}{\text{CPU,SMP}}$
500 x 4000	Mean (C00E)	2661.6	106.6	10511.0	236.5	57.2	11.89	1.98
	Min (C00E)	1683.0	50.0	8159.0	183.6	34.0	6.91	1.16
	Max (C00E)	4395.0	179.4	11783.0	265.1	83.4	23.94	4.14
	Mean (B25P)	1453.4	104.1				6.15	1.89
1000 x 2000	Mean (C00E)	1830.8	64.0	10568.8	158.5	42.9	11.69	1.54
	Min (C00E)	1344.0	41.0	8434.0	126.5	28.8	7.82	1.02
	Max (C00E)	2507.0	91.5	11576.0	173.6	70.4	15.83	2.02
	Mean (B25P)	1530.6	97.9				9.64	2.48
1000 x 8000	Mean (C00E)	2338.0	227.9	12406.6	139.6	113.2	16.68	1.99
	Min (C00E)	1453.0	119.4	11579.0	130.3	88.2	11.15	1.27
	Max (C00E)	2739.0	370.2	13895.0	156.3	168.9	18.99	2.39
	Mean (B25P)	1545.6	248.9				11.08	2.30
2000 x 8000	Mean (C00E)	2691.6	227.6	12922.8	96.9	74.5	27.93	3.10
	Min (C00E)	1132.0	97.7	10934.0	82.0	56.6	12.24	1.37
	Max (C00E)	3355.0	313.1	15632.0	117.2	88.8	35.46	4.25
	Mean (B25P)	1426.4	207.8				14.74	2.84
2000 x 16000	Mean (C00E)	2384.6	494.2	13174.8	74.1	184.9	32.30	2.68
	Min (C00E)	2288.0	486.3	11735.0	66.0	174.4	29.78	2.53
	Max (C00E)	2491.0	505.5	14729.0	82.9	195.3	34.66	2.84
	Mean (B25P)	1575.2	533.7				21.41	2.89
4000 x 8000	Mean (C00E)	2923.6	798.7	19750.2	74.1	228.4	39.42	3.30
	Min (C00E)	2032.0	407.6	17262.0	64.7	159.0	28.86	2.34
	Max (C00E)	3895.0	1539.7	22945.0	86.0	343.1	48.61	4.49
	Mean (B25P)	1554.6	576.2				21.12	2.63
4000 x 32000	Mean (C00E)	2482.8	2054.3	11973.2	84.2	515.8	29.47	3.98
	Min (C00E)	1826.0	1448.9	11331.0	79.7	499.9	22.39	2.90
	Max (C00E)	3479.0	2904.2	12715.0	89.4	525.0	42.65	5.70
	Mean (B25P)	1604.8	1736.3				19.19	3.36
8000 x 16000	Mean (C00E)	2680.4	2227.7	12474.6	58.5	375.0	45.78	5.92
	Min (C00E)	2297.0	1890.1	11493.0	53.9	341.9	41.12	5.44
	Max (C00E)	3177.0	2609.0	13759.0	64.5	408.8	49.26	6.48
	Mean (B25P)	1615.8	1752.7	12474.6	58.5	375.0	27.57	4.63

Table 2: Numerical Results for ℓ_1-minimization with $\|\cdot\|_2$-fit

		DMP		SMP				
Sizes		Calls	CPU	Calls	FCalls	CPU	$\frac{\text{Calls,DMP}}{\text{FCalls,SMP}}$	$\frac{\text{CPU,DMP}}{\text{CPU,SMP}}$
500 x 4000	Mean (C00E)	579.8	21.0	4771.6	106.7	24.6	5.91	0.93
	Min (C00E)	410.0	14.5	3412.0	76.3	16.9	3.18	0.49
	Max (C00E)	722.0	40.3	6868.0	153.5	36.0	8.40	1.94
	Mean (L75P)	287.8	16.1				2.95	0.70
1000 x 2000	Mean (C00E)	553.0	19.0	3910.8	54.8	13.6	10.73	1.47
	Min (C00E)	463.0	9.1	3315.0	46.4	11.5	5.68	0.52
	Max (C00E)	664.0	30.1	5890.0	82.5	17.4	13.56	2.34
	Mean (L75P)	282.4	14.1				5.44	1.07
1000 x 8000	Mean (C00E)	617.0	56.6	5148.8	57.5	50.7	11.25	1.17
	Min (C00E)	486.0	34.7	3745.0	41.9	36.1	7.68	0.74
	Max (C00E)	794.0	87.1	6050.0	67.6	64.8	18.35	1.93
	Mean (L75P)	318.8	40.9				5.84	0.86
2000 x 8000	Mean (C00E)	634.8	39.8	5853.6	41.0	47.2	15.94	0.86
	Min (C00E)	487.0	30.0	3926.0	27.5	33.1	11.17	0.59
	Max (C00E)	796.0	51.0	6869.0	48.1	54.0	20.49	1.12
	Mean (L75P)	318.8	25.9				8.05	0.58
2000 x 16000	Mean (C00E)	531.8	150.7	5055.6	28.3	90.0	19.88	1.80
	Min (C00E)	438.0	108.3	3947.0	22.1	60.2	11.64	0.87
	Max (C00E)	608.0	180.3	6736.0	37.6	125.1	24.80	2.49
	Mean (L75P)	346.0	110.6				12.74	1.28
4000 x 8000	Mean (C00E)	675.2	138.5	6504.6	22.8	101.7	29.71	1.36
	Min (C00E)	531.0	99.1	5868.0	20.5	83.3	22.71	0.99
	Max (C00E)	810.0	193.6	7143.0	25.0	113.9	34.52	1.70
	Mean (L75P)	346.4	86.3				15.21	0.85
4000 x 32000	Mean (C00E)	672.2	486.0	5613.4	39.2	287.2	17.66	1.74
	Min (C00E)	506.0	382.5	3418.0	23.9	197.2	12.08	1.26
	Max (C00E)	817.0	579.1	6611.0	46.2	336.4	22.57	2.15
	Mean (L75P)	355.4	311.6				9.39	1.12
8000 x 16000	Mean (C00E)	592.4	591.4	5815.0	25.4	177.6	24.15	3.51
	Min (C00E)	509.0	472.4	3765.0	16.5	117.3	16.56	2.36
	Max (C00E)	696.0	798.1	7038.0	30.8	214.1	30.90	5.06
	Mean (L75P)	329.8	360.2				13.38	2.10

Table 3: Experiments with dense $32,000 \times 64,000$ matrix A. Percents: $\|\widehat{x} - x_*\|/\|x_*\|$.

Method	p	Steps	Calls	FCalls	CPU, sec	$\|A\widehat{x} - b\|_p$	$\|\widehat{x} - x_*\|_1$	$\|\widehat{x} - x_*\|_2$	$\|\widehat{x} - x_*\|_\infty$
DMP (C00E)	∞	30	71	71	7564	0.16018	1.406 (141%)	0.143 (89%)	0.041 (79%)
DMP (B25P)	∞	31	67	67	7363	0.15975	1.361 (136%)	0.136 (85%)	0.035 (69%)
SMP (C00E)	∞	7501	22141	25.9	5352	0.00744	0.048 (5%)	0.005 (3%)	0.002 (4%)
DMP (C00E)	2	29	67	67	7471	0.03653	1.455 (146%)	0.135 (84%)	0.035 (68%)
DMP (L75P)	2	30	67	67	7536	0.02480	0.976 (98%)	0.093 (58%)	0.022 (42%)
SMP (C00E)	2	2602	7749	8.5	2350	0.00715	0.264 (26%)	0.021 (13%)	0.004 (7%)

A Appendix: Proofs

A.1 Low dimensional approximation

We use the notations of Section 2.1.3.

Lemma A.1 *Let Q_* be an optimal solution to (6), $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ be the eigenvalues of Q_*, $e_1, ..., e_n$ be the corresponding eigenvectors of Q_*, and $E = \mathrm{Lin}(e_1, ..., e_d)$. Then for any $v \in V$, $\mathrm{dist}(v, E) \leq \delta_* \sqrt{d+1}$ (here $\mathrm{dist}(x, E)$ stands for the Euclidean distance from v to E).*

Proof. Note that $0 \leq \lambda_j \leq 1$ and $\sum_j \lambda_j = d$, so that $\lambda_j \leq \frac{d}{d+1}$ when $j \geq d+1$. Denoting by $y_j(x)$ the coordinates of x in the eigenbasis $\{e_j\}$, we have

$$\begin{array}{l}
\mathrm{dist}^2(v,E) = 1 - \sum_{j=1}^d y_j^2(v) \leq 1 - \sum_{j=1}^d \lambda_j y_j^2(v) \text{ [since } \lambda_j \leq 1]\\
\leq 1 - \sum_{j=1}^n \lambda_j y_j^2(v) + \lambda_{d+1} \sum_{j=d+1}^n y_j^2(v) = 1 - v^T Q v + \lambda_{d+1} \mathrm{dist}^2(v,E)\\
\Rightarrow \quad (1-\lambda_{d+1})\mathrm{dist}^2(v,E) \leq 1 - v^T Q v \leq 1 - \mathrm{Opt} \leq 1 - \mathrm{Opt}_* = \delta_*^2\\
\Rightarrow \quad \mathrm{dist}^2(v,E) \leq (d+1)\delta_*^2 \text{ [since } \lambda_{d+1} \leq \frac{d}{d+1}],
\end{array}$$

as claimed. □

Representing a vector from $\Delta_{n,d}$ as a convex combination of extreme points. The case of $d = n$ is trivial, thus, let $d < n$. Let

$$q \in \Delta_{n,d} = \left\{ q \in \mathbf{R}^n_+ : \; 0 \leq q_i \leq 1 \,\forall i, \; \sum_{i=1}^n q_i = d \right\}.$$

To represent q as a convex combination of n extreme points of $\Delta_{n,d}$ we act as follows:

- *Initialization:* We set $p^0 = [1; q]$, $\mu^0 = 1$. Note that $p^0 \in \Delta = \{p = [1; p_1; ...; p_n] \in \Delta_{n+1,d+1}\}$.
- *Step $t = 1, 2, ...$:* Given $p^{t-1} = [1; p_1^{t-1}; ...; p_n^{t-1}] \in \Delta$, we find the $d+1$ largest among the entries p_i^{t-1}, $i = 1, ..., n$, let their indexes be $i_1, ..., i_{d+1}$, where $p_{i_1}^{t-1} \geq p_{i_2}^{t-1} \geq ... \geq p_{i_{d+1}}^{t-1}$.
  a) It may happen that $p_{i_\ell}^{t-1} = 1$ for $1 \leq \ell \leq d$; since $p^{t-1} \in \Delta$, $r^t := p^{t-1}$ is a Boolean vector with exactly $d+1$ entries equal to 1, and $q^t = [p_1^{t-1}; ...; p_n^{t-1}]$ is an extreme point of $\Delta_{n,d}$. We set $\nu_t = 1$, $p^t = 0$ and terminate.
  b) When not all $p_{i_\ell}^{t-1}$, $1 \leq \ell \leq d$, are equal to 1, we set $\nu_t = \min[1 - p_{i_{d+1}}^{t-1}, p_{i_d}^{t-1}]$, define r^t as Boolean $(n+1)$-dimensional vector with $d+1$ entries equal to 1, the indexes of the entries being $0, i_1, ..., i_d$, set $p^t = [p^{t-1} - \nu_t r^t]/(1-\nu_t)$, $q^t = [r_1^t; ...; r_n^t]$ (note that q^t is an extreme point of $\Delta_{n,d}$) and pass to step $t+1$.

Observe that the algorithm is well defined. Indeed, $0 \leq \nu_t \leq 1$ by construction, and $\nu_t = 1$ if and only if $p_{i_{d+1}}^{t-1} = 0$ and $p_{i_d}^{t-1} = 1$, that is, when we terminate at step t according to a). Thus, p^t is well defined at every non-termination step t. Moreover, from

b) it is immediately seen that at such a step we have $p_0^t = 1$, $0 \leq p_i^t \leq 1$ for all i and $\sum_{i=0}^n p_i^T = d+1$, that is, $p^t \in \Delta$ for all t for which p^t is well defined. Beside this, it is immediately seen that those entries in p^{t-1} which are zeros and ones remain zeros and ones in p^t as well, and that the total number of these entries increases at every step of the algorithm by at least 1. The latter observation implies that the algorithm terminates in at most n steps. Finally, by construction $p^{t-1} = (1-\nu_t)p^t + \nu_t r^t$, whence, denoting by $\bar{t}$ the termination step, p^0 is a convex combination of $r^1, ..., r^{\bar{t}}$ with coefficients μ_t readily given by $\nu_1, ..., \nu_{\bar{t}}$. Discarding in $r^1, ..., r^{\bar{t}}$ the entries with index 0, we get extreme points $q^1, ..., q^{\bar{t}}$ of $\Delta_{n,d}$ such that $q = \sum_{i=1}^{\bar{t}} \mu_t q^t$. Finally, the computational effort per step clearly does not exceed $O(1)dn$, that is, the total computational effort is at most $O(1)dn^2$.

A.2 Proof of Lemma 2.1

We have

$$\begin{aligned}
\mathrm{SV}(\rho) &= \max_{z_2\in Z_2} \min_{z_1 \in Z_1} \phi^\rho(z_1, z_2) \\
&= \max_{z_2\in Z_2} \min_{z_{11}\in Z_{11}, z_{12}\in Z_{12}} \Big[v + \rho\chi + \langle a_{11}, z_{11}\rangle + \langle b, z_2\rangle + \langle z_2, Bz_{11}\rangle \\
&\qquad\qquad + \rho \left[\langle a_{12}, z_{12}\rangle + \langle c, z_2\rangle + \langle z_2, Cz_{12}\rangle \right] \Big] \\
&= \max_{z_2\in Z_2} \Big[v + \rho\chi + \langle b, z_2\rangle + \rho\langle c, z_2\rangle + \min_{z_{11}\in Z_{11}} \left[\langle a_{11}, z_{11}\rangle + \langle z_2, Bz_{11}\rangle \right. \\
&\qquad\qquad \left. + \rho \overbrace{\min_{z_{12}\in Z_{12}} \left[\langle a_{12} + C^* z_2, z_{12}\rangle \right]}^{g(z_2)} \right] \Big] \\
&= \max_{z_2\in Z_2} \left[v + \rho\chi + \langle b, z_2\rangle + \rho\langle c, z_2\rangle + \rho g(z_2) + \overbrace{\min_{z_{11}\in Z_{11}} \left[\langle a_{11}, z_{11}\rangle + \langle z_2, Bz_{11}\rangle \right]}^{h(z_2)} \right] \\
&= \max_{z_2\in Z_2} \left[v + \langle b, z_2\rangle + h(z_2) + \rho \left[\chi + \langle c, z_2\rangle + g(z_2) \right] \right]
\end{aligned}$$

and thus $\mathrm{SV}(\rho)$ is the supremum of affine functions of ρ. □

A.3 Proofs for Section 3

We start with the following

Lemma A.2 [cf. [12], Lemma 3.1.(b)] *Given $z \in Z^o$, $\gamma > 0$ and $\xi, \eta \in E$, let us set*

$$\begin{aligned}
w &= \mathrm{Prox}_z(\gamma\xi) = \mathrm{argmin}_{v\in Z} \left\{ \langle \gamma\xi - \omega'(z), v\rangle + \omega(v) \right\}, \\
z_+ &= \mathrm{Prox}_z(\gamma\eta) = \mathrm{argmin}_{v\in Z} \left\{ \langle \gamma\eta - \omega'(z), v\rangle + \omega(v) \right\}.
\end{aligned}$$

Then $w, z_+ \in Z^o$, and for every $u \in Z$ one has

$$\begin{array}{ll}
(a) & \gamma\langle \eta, w - u\rangle \leq V_z(u) - V_{z_+}(u) + \gamma\langle \eta, w - z_+\rangle - V_z(z^+) \\
(b) & \leq V_z(u) - V_{z_+}(u) + \gamma\langle \eta - \xi, w - z_+\rangle - V_z(w) - V_w(z_+) \\
(c) & \leq V_z(u) - V_{z_+}(u) + \gamma\|\eta - \xi\|_* \|w - z_+\| - \frac{1}{2}\left[\|w - z\|^2 + \|w - z_+\|^2 \right] \\
(d) & \leq V_z(u) - V_{z_+}(u) + \frac{1}{2}\left[\gamma^2 \|\eta - \xi\|_*^2 - \|w - z\|^2 \right].
\end{array} \tag{62}$$

Proof. The inclusions $w, z_+ \in Z^o$ are evident (a subgradient of $\omega(\cdot)$ at w, taken w.r.t. Z, is, e.g., $\omega'(z) - \gamma\xi$, and similarly for z_+). Now let $u \in Z$. z_+ is an optimal solution of certain explicit convex optimization problem; taking into account that $\omega'(\cdot)$ is continuous on Z^o, it is easily seen that the necessary optimality condition in this problem reads $\langle \gamma\eta + \omega'(z_+) - \omega'(z), u - z_+\rangle \geq 0$, whence $\gamma\langle \eta, w - u\rangle \leq \gamma\langle \eta, w - z_+\rangle + \langle \omega'(z_+) - \omega'(z), u - z_+\rangle$, and the latter inequality, after rearranging terms in the right hand side, becomes (a). By similar reasons, $0 \leq \langle \gamma\xi + \omega'(w) - \omega'(z), v - w\rangle$ for all $v \in Z$; setting $v = z_+$, summing up the resulting inequality with (a) and rearranging terms in the right hand side of what we get, we arrive at (b). (c) follows from (b) due to $V_a(b) \geq \frac{1}{2}\|a - b\|^2$ (recall that ω is strongly convex, modulus 1 w.r.t. $\|\cdot\|$, on Z). Finally, (d) follows from (c) due to $\mu\nu - \frac{1}{2}\mu^2 \leq \frac{1}{2}\nu^2$. □

A.3.1 Proof of Proposition 3.1

Let us prove the bound (28). Consider first the case of SMP. Applying Lemma A.2 to $z = z_\tau$, $\gamma = \gamma_\tau$, $\xi = F(\eta_\tau)$, $\eta = F(\zeta_\tau)$, which results in $w = w_\tau$ and $z_+ = z_{\tau+1}$, we get for all $u \in Z$:

$$\gamma_\tau\langle F(\zeta_\tau), w_\tau - u\rangle \leq V_{z_\tau}(u) - V_{z_{\tau+1}}(u) + [\gamma_\tau\langle F(\zeta_\tau), w_\tau - z_{\tau+1}\rangle - V_{z_\tau}(z_{\tau+1})]$$

whence for all $u \in Z$

$$\begin{array}{rcl} \langle F(\zeta_\tau), \zeta_\tau - u\rangle & \leq & \gamma_\tau^{-1}(V_{z_\tau}(u) - V_{z_{\tau+1}}(u)) + r_\tau + s_\tau, \\ s_\tau & = & \langle F(\zeta_\tau), w_\tau - z_{\tau+1}\rangle - \gamma_\tau^{-1} V_{z_\tau}(z_{\tau+1}) \\ & \leq & \frac{1}{2}\left[\gamma_\tau\|F(\zeta_\tau) - F(\eta_\tau)\|_*^2 - \gamma_\tau^{-1}\|w_\tau - z_\tau\|^2\right], \quad (*) \\ r_\tau & = & \langle F(\zeta_\tau), \zeta_\tau - w_\tau\rangle. \end{array} \tag{63}$$

with $(*)$ given by (62). When summing up inequalities (63) over τ and taking into account that $\gamma_1 \geq \gamma_2 \geq ...$, $V_z(u) \geq 0$ and $V_{z_1}(u) = V_{z_\omega}(u) \leq \Omega$ by definition of Ω, we get

$$\sum_{\tau=1}^{t}\langle F(\zeta_\tau), \zeta_\tau - u\rangle \leq \gamma_t^{-1}\Omega + \sum_{\tau=1}^{t}[s_\tau + r_\tau]. \tag{64}$$

On the other hand,

$$\begin{array}{rl} \displaystyle\sum_{\tau=1}^{t}\langle F(\zeta_\tau), \zeta_\tau - u\rangle = \sum_{\tau=1}^{t}\langle a + \mathcal{A}\zeta_\tau, \zeta_\tau - u\rangle & \\ = & t\langle a, z^t - u\rangle - \displaystyle\sum_{\tau=1}^{t}\langle \mathcal{A}\zeta_\tau, u\rangle \quad [\mathcal{A} \text{ is skew symmetric}] \\ = & t\left[\langle a, z^t - u\rangle - \langle \mathcal{A}z^t, u\rangle\right] = t\left[\langle a, z^t - u\rangle + \langle \mathcal{A}z^t, z^t - u\rangle\right] \\ = & t\langle F(z^t), z^t - u\rangle. \end{array}$$

Thus, for all $u \in Z$ it holds

$$t\langle F(z^t), z^t - u\rangle \leq \Omega\gamma_t^{-1} + \sum_{\tau=1}^{t}[s_\tau + r_\tau] = \gamma_t^{-1}\Omega + S_t + R_t. \tag{65}$$

Setting $z^t = [z_1^t; z_2^t]$ and $u = [u_1; u_2]$, we get from (3) $\langle F(z^t), z^t - u\rangle = \phi(z_1^t, u_2) - \phi(u_1, z_2^t)$; the supremum of the latter quantity over $u \in Z$ is nothing that the saddle point residual $\epsilon_{\rm sad}(z^t)$. Since the right hand side in (65) is independent of u, we arrive at the SMP-version of (28).

Now consider the case of SA. Applying Lemma A.2 to $\gamma = \gamma_\tau$, $z = z_\tau$, $\xi = 0$, $\eta = F(\zeta_\tau)$, which results in $w = z_\tau$ and $z_+ = z_{\tau+1}$, and acting exactly as in the case of SMP, we arrive at the SA-version of (28).

Let us prove (ii). The conditional to the "past" (the answers of the oracle prior to the call for $\xi_{2\tau}$) distribution of ζ_τ is P_{w_τ}, which combines with the affinity of F and the facts that the linear part of F is skew symmetric and the expectation of P_z is z, to imply that

$$\begin{array}{rcl}\mathbf{E}\{\langle F(\zeta_\tau), \zeta_\tau - w_\tau\rangle\} &=& \langle a, \mathbf{E}\{\zeta_\tau\} - w_\tau\rangle + \mathbf{E}\{\langle \mathcal{A}\zeta_\tau, \zeta_\tau - w_\tau\rangle\} = -\mathbf{E}\{\langle \mathcal{A}\zeta_\tau, w_\tau\rangle\} \\ &=& \mathbf{E}\{\langle \mathcal{A}(w_\tau - \zeta_\tau), w_\tau\rangle\} = 0,\end{array}$$

whence $\mathbf{E}\{R_t\} = 0$ for all t. By completely similar reasoning, $\mathbf{E}\{R_t\} = 0$ in the case of SA. To complete the proof (ii), we need to prove (33). We have

$$\begin{array}{rcl} s_t &\leq& \frac{\gamma_t}{2}\|F(\zeta_t) - F(\eta_t)\|_*^2 - \frac{1}{2\gamma_t}\|w_t - z_t\|^2 \text{ [see (29)]} \\ &\leq& \frac{\gamma_t}{2}\left[\|F(w_t) - F(z_t)\|_* + \|F(\zeta_t) - F(w_t)\|_* + \|F(\eta_t) - F(z_t)\|_*\right]^2 - \frac{1}{2\gamma_t}\|w_t - z_t\|^2 \\ &\leq& \underbrace{\left[\frac{3\gamma_t}{2}\mathcal{L}^2 - \frac{1}{2\gamma_t}\right]}_{\leq 0 \text{ by (32)}}\|w_t - z_t\|^2 + \frac{3\gamma_t}{2}\left[\|F(\zeta_t) - F(w_t)\|_*^2 + \|F(\eta_t) - F(z_t)\|_*^2\right].\end{array}$$

It remains to note that $\|F(\zeta_t) - F(w_t)\|_*^2 + \|F(\eta_t) - F(z_t)\|_*^2 \leq 2M_*^2$ since $\zeta_t, w_t, \eta_t, z_t \in Z$ and that the conditional, over the respective pasts, expectations of $\|F(\zeta_t) - F(w_t)\|_*^2$ and $\|F(\eta_t) - F(z_t)\|_*^2$ do not exceed σ^2. □

A.3.2 Proof of Proposition 3.2

We start with observing that (39) $\gamma_1 \geq \gamma_2 \geq ....$
1^0. Let us verify first that with the choice (39) of γ_τ, $\tau = 1, 2, ...$ we have for all $t = 1, 2, ...,$

$$\sqrt{2}\Omega\gamma_t^{-1} \geq S_t. \tag{66}$$

Indeed, for $t = 2, 3, ...$ we have (with $2S_0 = F_*^2$ in the case of SA and $2S_0 = 8\Omega\mathcal{L}^2, \geq M_*^2$ by (21), in the case of SMP)

$$\frac{\gamma_{t-1}^2}{\gamma_t^2} = \frac{\sum_{\tau=1}^{t-1} 2[s_\tau]_+/\gamma_\tau + 2S_0}{\sum_{\tau=1}^{t-2} 2[s_\tau]_+/\gamma_\tau + 2S_0} \leq 1 + \frac{2[s_{t-1}]_+/\gamma_{t-1}}{2S_0} \leq 2 \tag{67}$$

(recall that $2s_t/\gamma_\tau \leq 2S_0$ by (30)). On the other hand

$$\gamma_t^{-2} - \gamma_{t-1}^{-2} = \frac{2[s_{t-1}]_+}{\Omega\gamma_{t-1}},$$

and

$$\gamma_t^{-1} - \gamma_{t-1}^{-1} \geq \tfrac{\gamma_t}{2}(\gamma_t^{-2} - \gamma_{t-1}^{-2}) = \tfrac{\gamma_t[s_{t-1}]_+}{\gamma_{t-1}\Omega} \geq \tfrac{[s_{t-1}]_+}{\sqrt{2}\Omega} \Rightarrow \sqrt{2}\Omega[\gamma_t^{-1} - \gamma_{t-1}^{-1}] \geq [s_{t-1}]_+$$

where the second inequality in this chain follows from $\gamma_{t-1} \leq \sqrt{2}\gamma_t$ which is implied by (67). By summing up the resulting inequalities in the above chain, we get

$$\sqrt{2}\Omega\gamma_t^{-1} \geq \sum_{\tau=1}^{t-1} s_\tau + \sqrt{2}\Omega\gamma_1^{-1}. \tag{68}$$

In the case of SMP, we have $\gamma_1 = (2\sqrt{2}\mathcal{L})^{-1}$, whence $\sqrt{2}\Omega\gamma_1^{-1} = 4\Omega\mathcal{L} \geq \gamma_1 M_*^2 \geq \gamma_t M_*^2$ (see (21)), whence $\sqrt{2}\Omega\gamma_1^{-1} \geq s_t$ in view of (29), and (68) implies (66). In the case of SA, we have $\gamma_1 = \sqrt{\Omega}/F_*$, whence $\sqrt{2}\Omega\gamma_1^{-1} = \sqrt{2}\sqrt{\Omega}F_* \geq \gamma_1 F_*^2 \geq \gamma_t F_*^2$, whence $\sqrt{2}\Omega\gamma_1^{-1} \geq s_t$ by (29), and (66) again is given by (68).

2^0. Invoking (28), (66) implies (40). Now, by (29) in the case of SA we have $2[s_\tau]_+/\gamma_\tau \leq \|F(\zeta_\tau)\|_*^2$. In the case of SMP we have

$$\begin{array}{l}
2[s_\tau]_+/\gamma_\tau \leq \|F(\zeta_\tau) - F(\eta_\tau)\|_*^2 - \gamma_\tau^{-2}\|w_\tau - z_\tau\|^2 \text{ [see (29)]}\\
\leq [\|F(\zeta_\tau) - F(w_\tau)\|_* + \|F(w_\tau) - F(z_\tau)\|_* + \|F(z_\tau) - F(\eta_\tau)\|_*]^2 - \gamma_1^{-2}\|w_\tau - z_\tau\|^2\\
\leq 3\,[\|F(\zeta_\tau) - F(w_\tau)\|_*^2 + \|F(z_\tau) - F(\eta_\tau)\|_*^2] + \left[3\|F(w_\tau) - F(z_\tau)\|_*^2 - \gamma_1^{-2}\|w_\tau - z_\tau\|^2\right]\\
\leq \varsigma_\tau := 3\,[\|F(\zeta_\tau) - F(w_\tau)\|_*^2 + \|F(z_\tau) - F(\eta_\tau)\|_*^2] \text{ [by (20) due to } \gamma_1^{-1} = 2\sqrt{2}\mathcal{L}]
\end{array}$$

Invoking (39), we get

$$\gamma_t^{-1} \leq \Omega^{-1/2} \cdot \left\{ \begin{array}{ll} \left(F_*^2 + \sum_{\tau=1}^{t-1} \|F(\zeta_\tau)\|_*^2\right)^{1/2}, & \text{in the case of SA} \\ \left(8\Omega\mathcal{L}^2 + \sum_{\tau=1}^{t-1} \varsigma_\tau\right)^{1/2}, & \text{in the case of SMP} \end{array} \right. \tag{69}$$

which combines with (40) to imply (41). □

A.3.3 Proof of Proposition 3.3

1^0. Let us denote

$$\varphi_t = 8\Omega\mathcal{L}^2 + \sum_{\tau=1}^{t-1} \varsigma_\tau,$$

where $\varsigma_t = 3\,[\|F(\zeta_t) - F(w_t)\|_*^2 + \|F(\eta_t) - F(z_t)\|_*^2]$ (cf. (42)). Let us show that under the premise of Proposition 3.3

$$\forall \Lambda \geq 0: \quad \text{Prob}\left\{\varphi_t \geq O(1)\left[\Omega\mathcal{L}^2 + \frac{M_*^2 t}{k}\varkappa_*^2(k, \Lambda)\right]\right\} \leq \exp\{-\Lambda t\}, \tag{70}$$

where $O(1)$ is an absolute constant factor. We use the following result (see, e.g., Theorem 2.1 (iii) of [8]): let $\xi^i, ..., \xi^k$ be k independent vectors from E with $\|\xi^i\|_* \leq \sigma$ and $\mathbf{E}\{\xi^i\} = 0$, where the norm $\|\cdot\|_*$ is $\varkappa$-regular, $\varkappa \geq 1$. Then for any $u \geq 0$

$$\text{Prob}\left\{\|\sum_{i=1}^k \xi^i\|_* \geq \left[\sqrt{2\varkappa} + u\sqrt{2}\right]\sigma\sqrt{k}\right\} \leq \exp\{-u^2/2\}.$$

When rewriting the above bound for $\xi^i = F(\zeta^i) - F(w)$ and $\xi^i = F(\eta^i) - F(z)$ and taking into account that $\|\xi^i\|_* \leq M_*$ we obtain

$$\forall u \geq 0: \quad \mathrm{Prob}\left\{\|\sum_{i=1}^{k} \xi^i\|_*^2 \geq M_*^2 k(\sqrt{2\varkappa} + \sqrt{2}u)^2\right\} \leq \exp\{-u^2/2\}.$$

So, if we denote Prob_t conditional probability over $\zeta_1, \eta_1, ..., \zeta_{t-1}, \eta_{t-1}$ being fixed, we get

$$\forall u \geq 0: \quad \mathrm{Prob}_t\left\{\varsigma_t \geq \frac{24M_*^2}{k}(\varkappa + u)\right\} \leq 2\exp\{-u/2\},$$

where $\varsigma_t = 3\left[\|F(\zeta_t) - F(w_t)\|_*^2 + \|F(\eta_t) - F(z_t)\|_*^2\right]$ (cf. (42)). When setting $\nu_t = \frac{\varsigma_t k}{24M_*^2}$, we have for the conditional expectation $\mathbf{E}_t$ over $\zeta_1, \eta_1, ..., \zeta_{t-1}, \eta_{t-1}$ being fixed and $0 \leq \alpha < 1$

$$\begin{aligned}
\mathbf{E}_t\left\{\exp\{\frac{\alpha}{2}\nu_t\}\right\} &\leq e^{\frac{\alpha\varkappa}{2}} + \frac{\alpha}{2}\int_{\varkappa}^{\infty} e^{\frac{\alpha u}{2}}\mathrm{Prob}_t\{\nu_t \geq u\}du \\
&\leq e^{\frac{\alpha\varkappa}{2}} + \alpha\int_{\varkappa}^{\infty}\exp\{-\frac{(1-\alpha)u}{2}\}du = \frac{1+\alpha}{1-\alpha}\exp\{\frac{\alpha\varkappa}{2}\}
\end{aligned}$$

When choosing $\alpha_* = \frac{\exp\{1\}-1}{\exp\{1\}+1}$ we get $\mathbf{E}_t\left\{\exp\{\frac{\alpha_*\nu_t}{2}\}\right\} \leq \exp\{\frac{\alpha_*\varkappa}{2} + 1\}$, so that

$$\begin{aligned}
\mathbf{E}\left\{\exp\{\sum_{\tau=1}^{t}\frac{\alpha_*\nu_\tau}{2}\}\right\} &= \mathbf{E}\left\{\mathbf{E}_t\left\{\exp\{\sum_{\tau=1}^{t-1}\frac{\alpha_*\nu_\tau}{2}\}\exp\{\frac{\alpha_*\nu_t}{2}\}\right\}\right\} \\
&= \mathbf{E}\left\{\exp\{\sum_{\tau=1}^{t-1}\frac{\alpha_*\nu_\tau}{2}\}\mathbf{E}_t\left\{\exp\{\frac{\alpha_*\nu_t}{2}\}\right\}\right\} \leq \exp\{t(\frac{\alpha_*\varkappa}{2} + 1)\}
\end{aligned}$$

Hence, when applying the Tchebychev inequality we find

$$\forall \Lambda \geq 0: \quad \mathrm{Prob}\left\{\sum_{\tau=1}^{t}\nu_\tau \geq t\left(\varkappa + \frac{2}{\alpha_*}(1+\Lambda)\right)\right\} \leq \exp\{-\Lambda t\}.$$

When recalling that $\varsigma_t \leq 6M_*^2$, we conclude that

$$\forall \Lambda \geq 0: \quad \mathrm{Prob}\left\{\sum_{\tau=1}^{t-1}\varsigma_\tau \geq \min\left[6M_*^2 t,\ \frac{24M_*^2 t}{k}\left(\varkappa + \frac{2}{\alpha_*}(1+\Lambda)\right)\right]\right\} \leq \exp\{-\Lambda t\}.$$

Since $\varkappa \geq 1$, $\varkappa + \frac{2}{\alpha_*}(1+\Lambda) \leq O(1)\varkappa_*^2(k,\Lambda)$, and we arrive at (70).

2^0. We have

$$\forall \lambda \geq 0: \ \mathrm{Prob}\left\{\frac{R_t}{t} \geq O(1)F_*\sqrt{\frac{\Omega\lambda}{kt}}\right\} \leq e^{-\lambda}. \tag{71}$$

Indeed, since $\mathcal{A}$ is skew-symmetric, i.e. $\langle \mathcal{A}z, z\rangle = 0$,

$$r_t \;=\; \langle F(\zeta_t),\, \zeta_t - w_t\rangle = \langle a + \mathcal{A}\zeta_t,\, \zeta_t - w_t\rangle = \langle a + \mathcal{A}w_t, \zeta_t - w_t\rangle = \langle F(w_t), \zeta_t - w_t\rangle.$$

We conclude that

$$\begin{array}{rcl}\frac{R_t}{t} &=& \frac{1}{t}\sum_{\tau=1}^{t} r_t = \frac{1}{t}\sum_{\tau=1}^{t}\langle F(w_\tau), \zeta_\tau - w_\tau\rangle = \frac{1}{t}\sum_{\tau=1}^{t}\left\langle F(w_\tau),\, \frac{1}{k}\sum_{i=1}^{k}\zeta_\tau^i - w_\tau\right\rangle \\ &=& \frac{1}{tk}\sum_{\tau=1}^{t}\sum_{i=1}^{k}\langle F(w_\tau),\, \zeta_\tau^i - w_\tau\rangle = \frac{1}{tk}\sum_{\tau=1}^{t}\sum_{i=1}^{k}\xi_\tau^i,\end{array}$$

where $\xi_\tau^i := \langle F(w_\tau),\, \zeta_\tau^i - w_\tau\rangle$ is a scalar martingale-difference with $|\xi_\tau^i| \leq 2\mathcal{R}F_* \leq 2\Theta F_*$ (cf. (19)). Then by the Azuma-Hoeffding inequality [1],

$$\forall \lambda \geq 0: \;\; \mathrm{Prob}\left\{\frac{R_t}{t} \geq 2\Theta F_*\sqrt{\frac{2\lambda}{kt}}\right\} \leq e^{-\lambda},$$

which implies (71).

Now we are done – when substituting the bounds (70) and (71) into (41) we get

$$\mathrm{Prob}\left\{\epsilon_{\mathrm{sad}}(z^t) \geq O(1)\left[\frac{\Omega\mathcal{L}}{t} + M_*\varkappa_*(k,\Lambda)\sqrt{\frac{\Omega}{kt}} + \Theta F_*\sqrt{\frac{\lambda}{kt}}\right]\right\} \leq e^{-\Lambda t} + e^{-\lambda},$$

which is (44) if we recall that $\Theta = \sqrt{2\Omega}$ and $F_* \leq \|a\|_* + 2\Theta\mathcal{L}$ (cf. (22)). □

A.3.4 Proof of Proposition 3.4

This proof is completely similar to the one of Proposition 3.1 and is omitted.

A.4 Proof of Theorem 4.1

1^0. From the description of the method it follows that

$$\forall t, s \geq 1, \rho \geq 0: u^{ts} \geq \mathrm{SV}(\rho_s) \geq \ell^{ts},\, \ell_{ts}(\rho) \leq \mathrm{SV}(\rho), \ell^{ts} \leq \ell_{ts}(\rho_s). \tag{72}$$

Let us prove by induction in s that $\rho_* \leq \rho_s \leq \rho_1$. The base $s = 1$ is evident. Now let $\rho_* \leq \rho_s \leq \rho_1$, and let stage $s+1$ take place. When passing from stage s to stage $s+1$, we are in the case B) and thus have $u^{ts} > \epsilon\rho_s$, $\ell^{ts} \geq \frac{3}{4}u^{ts} > \frac{3}{4}\epsilon\rho_s$, whence, in view of (72),

$$\ell_s(\rho_s) = \ell_{ts}(\rho_s) \geq \ell^{ts} \geq \frac{3}{4}\max[\epsilon\rho_s, \mathrm{SV}(\rho_s)] \;\&\; \ell_s(\rho_s) > 0. \tag{73}$$

This combines with $\ell_{ts}(\rho_*) \leq \mathrm{SV}(\rho_*) \leq 0$ and convexity of $\ell_{ts}(\cdot)$ to imply that $\rho_* \leq \rho_{s+1} < \rho_s$. Induction is complete.

Since $\rho_s \geq \rho_*$, u^{ts} is an upper bound on $\mathrm{SV}(\rho_s)$ and $u^{ts} \geq \overline{\phi}^{\rho_s}(w_1^{ts})$, we conclude that if the algorithm terminates at stage s, then the result ρ_s, w_1^{ts} is an ϵ-solution to the GBSP in question.

2^0. Let us prove (ii). The reasoning to follow goes back to [10]; we reproduce it here to make the paper self-contained. Let s be such that the stage $s+1$ takes place, and let u_s be the last bound u^{ts} built at stage s. Observe that

$$\frac{3}{4}\epsilon\rho_s < \frac{3}{4}u_s \leq \ell_s(\rho_s) \leq \mathrm{SV}(\rho_s) \leq u_s. \tag{74}$$

Since the convex function $\ell_s(\rho)$ is nonpositive at $\rho = \rho_{s+1}$ and is $\geq \frac{3}{4}u_s > 0$ at $\rho = \rho_s > \rho_{s+1}$, we have $g_s := \ell'_s(\rho_s) > 0$ and

$$\rho_s - \rho_{s+1} \geq \ell_s(\rho_s)/g_s \geq \frac{3}{4}u_s/g_s. \tag{75}$$

Now assume that $s > 1$ is such that the stage $s+1$ takes place. Applying (75) and (74) to $s-1$ in the role of s, we get $\rho_{s-1} - \rho_s \geq \frac{3}{4}u_{s-1}/g_{s-1}$ and $\frac{3}{4}u_s \leq \ell_s(\rho_s)$, whence, by convexity of $\ell_s(\cdot)$ and in view of (72), $u_{s-1} \geq \mathrm{SV}(\rho_{s-1}) \geq \ell_s(\rho_{s-1}) \geq \ell_s(\rho_s) + g_s(\rho_{s-1} - \rho_s) \geq \frac{3}{4}u_s + g_s\frac{3}{4}\frac{u_{s-1}}{g_{s-1}}$, so that $\frac{4}{3}u_{s-1} \geq u_s + \frac{g_s u_{s-1}}{g_{s-1}}$, or $\frac{u_s}{u_{s-1}} + \frac{g_s}{g_{s-1}} \leq \frac{4}{3}$, whence $\frac{u_s g_s}{u_{s-1}g_{s-1}} \leq (1/4)(4/3)^2 = 4/9$. It follows that

$$\sqrt{u_s g_s} \leq (2/3)^{s-1}\sqrt{u_1 g_1}. \tag{76}$$

We have $\ell_s(\rho_*) \leq \mathrm{SV}(\rho_*) = 0$, $\ell_s(\rho_s) \geq \frac{3}{4}u_s$ (see (74)) and $\ell_s(\rho_s) - \ell_s(\rho_*) \leq g_s(\rho_s - \rho_*)$ (convexity of $\ell_s(\cdot)$), whence $g_s \geq \frac{3}{4}u_s(\rho_s - \rho_*)^{-1} \geq \frac{3}{4\rho_1}u_s$, and (76) implies that

$$u_s \leq (2/3)^{s-1}\sqrt{u_1 g_1}\sqrt{4\rho_1/3}. \tag{77}$$

Now, $g_1 = \ell'_1(\rho_1)$ and $\ell_1(\rho) \leq \mathrm{SV}(\rho) \leq \|\phi\|_\infty + \rho\|\psi\|_\infty$, whence $g_1 \leq \|\psi\|_\infty$, and clearly $u_1 \leq \|\phi\|_\infty + \rho_1\|\psi\|_\infty$. At the same time, $u_s > \epsilon\rho_s \geq \epsilon\rho_*$, so that (77) implies that $\epsilon\rho_* \leq (2/3)^{s-1}[\|\phi\|_\infty + \rho_1\|\psi\|_\infty]$. The resulting upper bound on s implies (ii).

3^0. Let us prove (iii). From the description of the algorithm it follows that at every stage s before termination of the stage the residual of current approximate solutions w^{ts} is $\geq \frac{1}{4}\epsilon\rho_s$ (since $u^{ts} > \epsilon\rho_s$ and $\ell^{ts} < \frac{3}{4}u^{ts}$). In the case of short-step implementation we use the result of Proposition (3.3) with $\varepsilon = \epsilon\rho_s$. Let us denote $N_s(\epsilon)$ the corresponding value of N_ε as in (45). We conclude that the number N_s of steps at stage s is finite with probability 1 and satisfies $\mathrm{Prob}\{N_s > N_s(\epsilon)\} \leq \exp\{-\Lambda N_s(\epsilon)\} + \exp\{-\lambda\}$. As we have seen, $\rho_* \leq \rho_s$ for all s, and therefore $N_s \leq N(\epsilon)$ for all s, provided that the absolute constant $O(1)$ in (59) is properly chosen.

For the aggressive-step implementation, similar reasoning based on the bound (51) with $\mathcal{L} = \mathcal{M} + \rho_s\mathcal{N}$ justifies (60).

4^0. Combining (ii), (iii) and the concluding claim in item 1^0 above, we arrive at (i). □

References

[1] Azuma, K., "Weighted sums of certain dependent random variables" *Tökuku Math. J.*, **19** (1967), 357-367.

[2] Candès, E.J., Tao, T., "Decoding by linear programming" – *IEEE Trans. Inform. Theory*, **51** (2006), 4203-4215.

[3] Candès, E.J., "Compressive sampling", Marta Sanz-Solé, Javier Soria, Juan Luis Varona, Joan Verdera, Eds. *International Congress of Mathematicians, Madrid 2006*, Vol. III, 1437–1452. European Mathematical Society Publishing House, (2006).

[4] Cristiani, N., Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, (2000).

[5] Dalalyan, A. S., Juditsky, A., Spokoiny, V., A New Algorithm for Estimating the Effective Dimension-Reduction Subspace, *J. of Machine Learning Res.*, **9** (2008), 1647-1678.

[6] Donoho, D. Huo, X., "Uncertainty principles and ideal atomic decomposition" – *IEEE Trans. Inform. Theory*, **47:7** (2001), 2845-2862.

[7] Grigoriadis, M.D., Khachiyan, l.G., "A sublinear-time randomized approximation algorithm for matrix games" – *Oper. Res. Lett.*, **18** (1995), 53-58.

[8] Judistky, A., Nemirovski, A., (2008), Large Deviations of Vector-Valued Martingales in 2-Smooth Normed Spaces.
E-print: http://www2.isye.gatech.edu/∼nemirovs/LargeDevSubmitted.pdf

[9] Juditsky, A., Nemirovski, A., Tauvel, C. (2008), "Solving variational inequalities with Stochastic Mirror Prox algorithm" – submitted to *SIAM Journal on Optimization*.
E-print: http://www2.isye.gatech.edu/∼nemirovs/SMP_240408.pdf

[10] Lemarechal, C., Nemirovski, A., Nesterov, Yu., "New variants of bundle methods" - *Mathematical Programming*, **69:1** (1995), 111-148

[11] Nemirovskii, A., "Efficient methods for large-scale convex problems" – *Ekonomika i Matematicheskie Metody (in Russian)*, **15** (1979).

[12] Nemirovski, A., "Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems" – *SIAM Journal on Optimization*, **15** (2004), 229–251.

[13] Nemirovski, A., Juditsky, A. Lan, G., Shapiro, A., "Stochastic Approximation Approach to Stochastic Programming" - *SIAM Journal on Optimization*, **19:4** (2009), 1574-1609.

[14] Nesterov, Yu. "Smooth minimization of non-smooth functions" – CORE Discussion Paper 2003/12, February 2003; *Math. Progr.*, **103** (2005), 127-152.

[15] Nesterov, Yu. "Excessive gap technique in nonsmooth convex minimization" – *SIAM J. Optim.*, **16** (2005), 235-249.

[16] Rubinstein, E., *Support Vector Machines via advanced optimization techniques*, M.Sc. Thesis, Technion, 2005.
E-print: http://www2.isye.gatech.edu/∼nemirovs/Eitan.pdf