Almost surely constrained convex optimization

Olivier Fercoq [1] **Ahmet Alacaoglu** [2] **Ion Necoara** [3] **Volkan Cevher** [2]

Abstract

We propose a stochastic gradient framework for solving stochastic composite convex optimization problems with (possibly) infinite number of linear inclusion constraints that need to be satisfied almost surely. We use smoothing and homotopy techniques to handle constraints without the need for matrix-valued projections. We show for our stochastic gradient algorithm $\mathcal{O}(\log(k)/\sqrt{k})$ convergence rate for general convex objectives and $\mathcal{O}(\log(k)/k)$ convergence rate for restricted strongly convex objectives. These rates are known to be optimal up to logarithmic factor, even without constraints. We conduct numerical experiments on basis pursuit, hard margin support vector machines and portfolio optimization problems and show that our algorithm achieves state-of-the-art practical performance.

1. Introduction

In many machine learning applications, optimization problems involve stochasticity in objective functions or constraint sets. Even though the problems with stochastic objective functions are well-studied in the literature, investigation of stochastic constraints seems to be rather scarce. In particular, we focus on the following stochastic convex optimization template:

$$\min_{x\in\mathbb{R}^d}\{P(x) := F(x) + h(x)\} \tag{1}$$
$$A(\xi)x \in b(\xi) \;\; \xi\text{-almost surely},$$

where $F(x) = \mathbb{E}_\xi\left[f(x,\xi)\right]$, with a convex and smooth $f(\cdot,\xi) : \mathbb{R}^d \to \mathbb{R}$ such that $\mathbb{E}\left[\nabla f(x,\xi)\right] = \nabla F(x)$; and $h : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$ is a nonsmooth, proximable convex function. Moreover, $b(\xi)$ are projectable closed convex sets. We seek to satisfy the stochastic linear inclusion constraints in (1) *almost surely*. Note that this goal is different from satisfying constraints *in expectation*, which is studied e.g. in (Lan and Zhou, 2016). We argue that this change is also what sets (1) apart from the standard stochastic setting in the literature. Indeed, we assume that $A(\xi) \in \mathbb{R}^{m\times d}$ is a matrix-valued random variable and $b(\xi) \subseteq \mathbb{R}^m$ is a random convex set. For the special case when $A(\xi)$ is an identity matrix, (1) recovers optimization problems where the constraint set is the intersection of a possibly infinite number of sets.

Applications of almost surely constrained problems arise in many fields, such as machine learning, operations research, and mathematical finance. Interesting cases include semi-infinite linear programming, sparse regression, portfolio optimization, classification, distributed optimization and streaming settings, and consensus optimization problems in standard constrained optimization where the access to full data is not possible (Sonnenburg et al., 2006; Abdelaziz et al., 2007; Nedić et al., 2018; Towfic and Sayed, 2015).

Particular instances of (1) involve primal support vector machines (SVM) classification and sparse regression which are central in machine learning (Shalev-Shwartz et al., 2011; Garrigues and Ghaoui, 2009). Due to the huge volume of data that is used for these applications, storing or processing this data at once is in general not possible. Therefore, using these data points one by one or in mini batches in learning algorithms is becoming more important. One direction that the literature focused so far is solving unconstrained formulations of these problems, successes of which are amenable to regularization parameters that needs to be tuned. By presenting a method capable of solving (1) directly, we present a parameter-free approach for solving these problems.

The most popular method for solving constrained stochastic optimization problems is projected stochastic gradient descent (SGD) (Nemirovski et al., 2009). However, in the case where we have infinite number of constraints, it is not clear how to apply the projection step. To remedy this issue, many methods utilize alternating projections to tackle stochastic constraints by viewing them as an intersection of possibly infinite sets (Patrascu and Necoara, 2017) (see Section 5 for a detailed discussion). For the special case when $A(\xi)$ is a vector-valued random variable, or identity matrix, projection methods are efficient. However, in the general case,

[1] LTCI, Télécom ParisTech, Université Paris-Saclay [2] Laboratory for Information and Inference Systems, École Polytechnique Fédérale de Lausanne [3] Department of Automatic Control and Systems Engineering, University Politehnica Bucharest. Correspondence to: Olivier Fercoq <olivier.fercoq@telecom-paristech.fr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

applying projection with matrix-valued $A(\xi)$ may clearly impose a serious computational burden per iteration.

In this work, we take a radically different approach and use Nesterov's smoothing technique (Nesterov, 2005) for almost sure constraints instead of applying alternating projections. In doing so, we avoid the need for projections on the linear constraints. We make use of the stochastic gradients of $f(x, \xi)$, proximal operators of simple nonsmooth component $h(x)$ and simple projections on the set $b(\xi)$.

In a nutshell, our analysis technique combines ideas of smoothing and homotopy in the stochastic gradient framework. We extend the previous analysis on smoothing with homotopy (Tran-Dinh et al., 2018b) to stochastic optimization with infinitely many constraints. To our knowledge, this is the first application of smoothing for stochastic constraints. Our contributions can be summarized as follows:

- We provide a simple stochastic gradient type algorithm which does not involve projections with matrix-valued random variables or heuristic parameter tuning.
- We prove $\mathcal{O}(\log(k)/\sqrt{k})$ convergence rate for general convex objectives.
- We prove $\mathcal{O}(\log(k)/k)$ convergence rate for restricted strongly convex objectives.
- We include generalizations of our framework for composite optimization with general nonsmooth Lispchitz continuous functions in addition to indicator functions.
- We provide numerical evidence and verify our theoretical results in practice.

Roadmap. We recall the basic theoretical tools that we utilize and lay out the notation in Section 2. The algorithm and its convergence guarantees are presented in Section 3. Section 4 shows how our results can be used to recover and extend previous works. We review the related literature and compare our method with the existing ones in Section 5. We conclude by presenting the practical performance of our method on three different problem instances in Section 6. Proofs of the theoretical results are deferred to the appendix.

2. Preliminaries

Notation. We use $\|\cdot\|$ to denote Euclidean norm and $\langle\cdot,\cdot\rangle$ to denote Euclidean inner product. The adjoint of a continuous linear operator is denoted by $^\top$. We will write a.s. in place of "almost surely" in the sequel.

We define the distance function to quantify the distance between a point x and set $\mathcal{K}$ as $\operatorname{dist}(x, \mathcal{K}) = \inf_{z\in\mathcal{K}} \|x - z\|$. Given a function ϕ, we use $\partial\phi(x)$ to denote its subdifferential at x. For a given set $\mathcal{K}$, we denote the indicator function of the set by $\delta_{\mathcal{K}}(x) = 0$, if $x \in \mathcal{K}$ and $\delta_{\mathcal{K}}(x) = +\infty$ otherwise. We define the support function of the set $\mathcal{K}$ as $\operatorname{supp}_{\mathcal{K}}(x) = \sup_{y\in\mathcal{K}}\langle x, y\rangle$. The domain of a convex function f is $\operatorname{dom}(f) = \{x \ : \ f(x) < +\infty\}$. We use $\tilde{\mathcal{O}}$ notation to suppress the logarithmic terms.

We define the proximal operator of a convex function ϕ as

$$\operatorname{prox}_{\phi}(z) = \arg\min_{x} \phi(x) + \frac{1}{2}\|x - z\|^2.$$

We say that ϕ is a proximable function if computing its proximal operator is efficient.

Given a function f and $L > 0$, we say that f is L-smooth if ∇f is Lipschitz continuous, *i.e.*,

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \ \ \forall x, y \in \mathbb{R}^d.$$

The function f is $\mu > 0$ strongly convex if it satisfies,

$$f(x) \ge f(y) + \langle\nabla f(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2, \ \ \forall x, y \in \mathbb{R}^d,$$

and, we say that the function f is μ-restricted strongly convex if there exists $x_\star$ such that,

$$f(x) \ge f(x_\star) + \frac{\mu}{2}\|x - x_\star\|^2, \ \ \forall x \in \mathbb{R}^d.$$

It is known that restricted strong convexity is a weaker condition than strong convexity since it is implied by strong convexity along the direction of the solution (Necoara et al., 2018a; Bolte et al., 2017).

Space of random variables. We will consider in this paper random variables $y(\xi) \in \mathbb{R}^m$ belonging to the space

$$\mathcal{Y} = \{(y(\xi))_\xi \ : \ \mathbb{E}[\|y(\xi)\|^2] < +\infty\}.$$

We shall denote by μ the probability measure of the random variable ξ, and we endow $\mathcal{Y}$ with the scalar product

$$\langle y, z\rangle = \mathbb{E}[y(\xi)^\top z(\xi)] = \int y(\xi)^\top z(\xi)\mu(d\xi).$$

$\mathcal{Y}$ is a Hilbert space and the norm is defined as $\|y\| = \sqrt{\mathbb{E}[\|y(\xi)\|^2]}$.

Smoothing. We are going to utilize Nesterov's smoothing framework to process almost sure linear constraints. Due to (Nesterov, 2005), a smooth approximation of a nonsmooth convex function g can be obtained as

$$g_\beta(z) = \max_u \langle u, z\rangle - g^*(u) - \frac{\beta}{2}\|u\|^2, \qquad (2)$$

where $g^*(u) = \sup_z\langle z, u\rangle - g(z)$ is the Fenchel-conjugate of g and $\beta > 0$ is the *smoothness* parameter. As shown in (Nesterov, 2005), $g_\beta(\cdot)$ is convex and $1/\beta$-smooth.

For the special case of indicator functions, $g(x) = \delta_b(x)$, where b is a given convex set, $g^*(x) = \operatorname{supp}_b(x)$ and the smooth approximation is given by $g_\beta(z) = \frac{1}{2\beta}\,\mathrm{dist}\,(z,b)^2$. Smoothing the indicator function is studied in (Tran-Dinh et al., 2018b) for the case of deterministic optimization, which we extend to the stochastic setting in this work.

Duality. We define the stochastic function

$$g(A(\xi)x,\xi) = \delta_{b(\xi)}(A(\xi)x).$$

Using basic probability arguments, Problem (1) can be written equivalently as:

$$\min_{x\in\mathbb{R}^d}\mathbb{E}[f(x,\xi)] + h(x) + \mathbb{E}[g(A(\xi)x,\xi)] =: P(x)+G(Ax)$$

where $P(x) = \mathbb{E}[f(x,\xi)] + h(x)$, $A : \mathbb{R}^d \to \mathcal{Y}$ is the linear operator such that $(Ax)(\xi) = A(\xi)x$ for all x and $G : \mathcal{Y} \to \mathbb{R}\cup\{+\infty\}$ is defined by

$$G(z) = \int \delta_{b(\xi)}(z(\xi))\mu(d\xi).$$

We will assume that

$$\|A\|_{2,\infty} = \sup_{\xi}\|A(\xi)\| < +\infty, \tag{3}$$

so that A is in fact continuous. Note that assuming a uniform bound on $\|A(\xi)\|$ is not restrictive since, as soon as $\|A(\xi)\| \neq 0$, we can replace $A(\xi)x \in b(\xi)$ by

$$A'(\xi)x = \frac{A(\xi)x}{\|A(\xi)\|} \in b'(\xi) = \frac{b(\xi)}{\|A(\xi)\|},$$

without changing the set of vectors x satisfying the constraint, and projecting onto $b'(\xi)$ is as easy as projecting onto $b(\xi)$.

For the case of stochastic constraints, we define the Lagrangian $\mathcal{L} : \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}\cup\{+\infty\}$ as

$$\mathcal{L}(x,y) = P(x) + \int \langle A(\xi)x, y(\xi)\rangle - \mathrm{supp}_{b(\xi)}(y(\xi))\mu(d\xi).$$

Using the Lagrangian, one can equivalently define primal and dual problems as

$$\min_{x\in\mathbb{R}^d}\max_{y\in\mathcal{Y}}\mathcal{L}(x,y), \quad \text{and,} \quad \max_{y\in\mathcal{Y}}\min_{x\in\mathbb{R}^d}\mathcal{L}(x,y).$$

Strong duality refers to values of these problems being equal. It is known that Slater's condition is a sufficient condition for strong duality to hold (Bauschke et al., 2011). In the context of duality in Hilbert spaces, Slater's condition refers to the following:

$$0 \in \mathrm{sri}(\mathrm{dom}\,(G) - A(\mathrm{dom}\,(P)))$$

where $\mathrm{sri}(\cdot)$ refers to the strong relative interior of the set (Bauschke et al., 2011).

Optimality conditions. We denote by $(x_\star, y_\star) \in \mathbb{R}^d \times \mathcal{Y}$ a saddle point of $\mathcal{L}(x,y)$. For the constrained problem, we say that x is an ϵ-solution if it satisfies the following objective suboptimality and feasibility conditions

$$|P(x) - P(x_\star)| \le \epsilon, \quad \sqrt{\mathbb{E}\left[\mathrm{dist}(A(\xi)x, b(\xi))^2\right]} \le \epsilon. \tag{4}$$

3. Algorithm & Convergence

We derive the main step of our algorithm from smoothing framework. The problem in (1) is nonsmooth both due to $h(x)$ and the constraints encoded in $g(A(\xi)x,\xi)$. We keep $h(x)$ intact since it is proximable, and smooth g to get

$$P_\beta(x) = \mathbb{E}\left[f(x,\xi)\right] + h(x) + \mathbb{E}\left[g_\beta(A(\xi)x,\xi)\right], \tag{5}$$

where $g_\beta(A(\xi)x, b(\xi)) = \frac{1}{2\beta}\,\mathrm{dist}(A(\xi)x, b(\xi))^2$. We note that $\mathbb{E}\left[f(x,\xi)\right] + \mathbb{E}\left[g_\beta(A(\xi)x,\xi)\right]$ is $L(\nabla F) + \frac{\|A\|_{2,2}^2}{\beta}$-smooth where

$$\|A\|_{2,2} = \sup_{x\neq 0}\frac{\sqrt{\mathbb{E}[\|A(\xi)x\|^2]}}{\|x\|} \le \|A\|_{2,\infty},$$

$\|A\|_{2,\infty}$ being defined in (3). Note that (5) can also be viewed as a quadratic penalty (QP) formulation.

The main idea of our method is to apply stochastic proximal gradient (SPG) (Rosasco et al., 2014) iterations to (5) by using homotopy on the smoothness parameter β. Our algorithm has a double loop structure where for each value of β, we solve the problem (5) with SPG upto some accuracy. This strategy is similar to inexact quadratic penalty (QP) methods which are studied for deterministic problems in (Lan and Monteiro, 2013). In stark contrast to inexact QP methods, Algorithm 1 has explicit number of iterations for the inner loop which is determined by theoretical analysis, avoiding difficult-to-check stopping criteria for the inner loop in standard inexact methods. We decrease β to 0 according to update rules from theoretical analysis to ensure the convergence to the original problem (1) rather than the smoothed problem (5).

In Algorithm 1, we present our stochastic approximation method for almost surely constrained problems (SASC, pronounced as "sassy"). We note that Case 1 refers to parameters for general convex case and Case 2 refers to restricted strongly convex case.

It may look unusual at first glance that in the restricted strongly convex case, the step size α_s is decreasing faster than in the general convex case. The reason is that restricted strong convexity allows us to decrease faster the smoothness parameter β_s, and that the step size is driven by the smoothness of the approximation.

We will present a key technical lemma which is instrumental in our convergence analysis. It will serve as a bridge to transfer bounds on the smoothness parameter and an auxiliary function that we define in (6) to optimality results in the usual sense for constrained optimization, $i.e.$ (4). This lemma can be seen as an extension of Lemma 1 from (Tran-Dinh et al., 2018b) to the case of almost sure constraints. We first define the auxiliary function that we are going to

Algorithm 1 SASC

```
Input: $x_0^0 \in \mathbb{R}^d$
Parameters: $\alpha_0 \le \dfrac{3}{4L(\nabla F)}$, and $\omega > 1$
  Case 1: $m_0 \in \mathbb{N}_*$.
  Case 2: $m_0 \ge \frac{\omega}{\mu\alpha_0}$.
for $s \in \mathbb{N}$ do
  $m_s = \lfloor m_0\omega^s \rfloor$, and $\beta_s = 4\alpha_s \left\|A\right\|_{2,\infty}^2$
  Case 1: $\alpha_s = \alpha_0\omega^{-s/2}$.
  Case 2: $\alpha_s = \alpha_0\omega^{-s}$.
  for $k \in \{0, \ldots, m_s - 1\}$ do
    Draw $\xi = \xi_{k+1}^s$, and define $z = A(\xi)x_k^s$.
    $D(x_k^s, \xi) := \nabla f(x_k^s, \xi) + A(\xi)^\top \nabla_z g_{\beta_s}(A(\xi)x_k^s, \xi)$
    $x_{k+1}^s = \mathrm{prox}_{\alpha_s h}\left(x_k^s - \alpha_s D(x_k^s, \xi)\right)$
  end for
  $\bar{x}^s = \frac{1}{m_s}\sum_{k=1}^{m_s} x_k^s$
  Case 1: $x_0^{s+1} = x_{m_s}^s$.
  Case 2: $x_0^{s+1} = \bar{x}^s$.
end for
return $\bar{x}^s$
```

utilize, which we name as the smoothed gap function

$$S_\beta(x) = P_\beta(x) - P(x_\star). \qquad (6)$$

Lemma 3.1. *Let $(x_\star, y_\star)$ be a saddle point of*

$$\min_{x\in\mathbb{R}^d}\max_{y\in\mathcal{Y}} \mathcal{L}(x, y),$$

and note that $S_\beta(x) = P_\beta(x) - P(x_\star) = P(x) - P(x_\star) + \frac{1}{2\beta}\int \mathrm{dist}\left(A(\xi)x, b(\xi)\right)^2\mu(d\xi)$. Then, the following hold:

$$S_\beta(x) \ge -\frac{\beta}{2}\|y_\star\|^2,$$

$$P(x) - P(x_\star) \ge -\frac{1}{4\beta}\int \mathrm{dist}\left(A(\xi)x, b(\xi)\right)^2\mu(d\xi) - \beta\|y_\star\|^2,$$

$$P(x) - P(x_\star) \le S_\beta(x),$$

$$\int \mathrm{dist}\left(A(\xi)x, b(\xi)\right)^2\mu(d\xi) \le 4\beta^2\|y_\star\|^2 + 4\beta S_\beta(x).$$

The simple message of Lemma 3.1 is that if an algorithm decreases $S_\beta(x)$ and β simultaneously, then it obtains approximate solutions to (1) in the sense of (4), $i.e.$ it decreases feasibility and objective suboptimality.

The main technical challenge of applying SPG steps to problem (5) with homotopy stems from the stochastic term due to constraints, which is $\mathbb{E}[g_\beta(A(\xi)x, \xi)]$, with $g_\beta(A(\xi)x, b(\xi)) = \frac{1}{2\beta}\,\mathrm{dist}(A(\xi)x, b(\xi))^2$.

Even though this term is in a suitable form to apply SPG, its variance bound and Lipschitz constant of its gradient becomes worse and worse as $\beta \to 0$. A naive solution for this problem would be to decrease β slowly, so that these bounds will increase slowly so that they can be dominated by the step size. Due to Lemma 3.1 however, the rate of decrease of β directly determines the convergence rate, so a slowly decaying β would result in slow convergence for the method. Our proof technique carefully balances the rate of β_k and the additional error terms due to using stochastic gradients of $\mathbb{E}[g_\beta(A(\xi)x, \xi)]$, so that the optimal rate of SPG is retained even with constraints.

We are going to present the main theorems in the following two sections for general convex and restricted strongly convex objectives, respectively. The main proof strategy in Theorem 3.2 and Theorem 3.3 is to analyze the convergence of $S_\beta(x)$ and β and use Lemma 3.1 to translate the rates to objective residual and feasibility measures.

3.1. Convergence for General Convex Objectives

In this section, we present the convergence results for solving (1) where only general convexity is assumed for the objective $P(x)$.

Theorem 3.2. *Assume F is convex and $L(\nabla F)$ smooth, and $\exists\sigma_f$ such that $\mathbb{E}[\|\nabla f(x,\xi) - \nabla F(x)\|^2] \le \sigma_f^2$. Denote $M_s = \sum_{l=0}^s m_l$. Let us set $\omega > 1$, $\alpha_0 \le \frac{3}{4L(\nabla f)}$, $m_0 \in \mathbb{N}_*$, $m_s = \lfloor m_0\omega^s \rfloor$, and $\beta_s = 4\alpha_s\|A\|_{2,\infty}^2$. Then, for all s,*

$$\mathbb{E}[P(\bar{x}^s) - P(x_\star)] \le \frac{C_1}{\sqrt{M_s}}\left[C_2 + \frac{\log(M_s/m_0)}{\log(\omega)}C_3\right]$$

$$\mathbb{E}[P(\bar{x}^s) - P(x_\star)] \ge -\frac{2C_4}{\sqrt{M_s}}\|y_\star\|^2 - \frac{C_1}{\sqrt{M_s}}\left[C_2 + \frac{\log(M_s/m_0)}{\log(\omega)}C_3\right]$$

$$\sqrt{\mathbb{E}\left[\mathrm{dist}(A(\xi)\bar{x}^s, b(\xi))^2\right]} \le \frac{1}{\sqrt{M_s}}\left[2C_4\|y_\star\| + 2\sqrt{C_1C_4}\sqrt{C_2 + \frac{\log(M_s/m_0)}{\log(\omega)}C_3}\right]$$

where $C_1 = \frac{\sqrt{m_0\omega}}{\alpha_0(m_0-1)\sqrt{\omega-1}}$, $C_2 = \frac{\|x_\star - x_0^0\|^2}{2} + 2\alpha_0 m_0\sigma_f^2$, $C_3 = 2\alpha_0^2\|A\|_{2,\infty}^2 m_0\|y_\star\|^2 + 2\alpha_0 m_0\sigma_f^2$ and $C_4 = 4\alpha_0\sqrt{m_0}\|A\|_{2,\infty}^2\frac{\sqrt{\omega}}{\sqrt{\omega-1}}$.

Note that the $\mathcal{O}(1/\sqrt{k})$ rate is known to be optimal for solving (1) with SGD (Polyak and Juditsky, 1992; Agarwal et al., 2009). In Theorem 3.2, we show that by handling infinite number of constraints without projections, we only lose a logarithmic factor from this rate.

3.2. Convergence for Restricted Strongly Convex Objectives

In this section, we assume $P(x)$ in (1) to be restricted strongly convex in addition to F being $L(\nabla F)$ smooth. Note that requiring restricted strong convexity of $P(x)$ is substantially weaker than requiring strong convexity of component functions $f(x, \xi)$ or $h(x)$, see (Necoara et al., 2018a) for more details. In this setting, we have:

Theorem 3.3. *Assume F is convex and $L(\nabla F)$ smooth, P is μ-restricted strongly convex and $\exists \sigma_f$ such that $\mathbb{E}[\|\nabla f(x, \xi) - \nabla F(x)\|^2] \leq \sigma_f^2$. Denote $M_s = \sum_{l=0}^{s} m_l$. Let us set $\omega > 1$, $\alpha_0 \leq \frac{3}{4L(\nabla f)}$, $m_0 \geq \frac{\omega}{\mu \alpha_0}$, $m_s = \lfloor m_0 \omega^s \rfloor$, and $\beta_s = 4\alpha_s \|A\|_{2,\infty}^2$. Then, for all s,*

$$\mathbb{E}[P(\bar{x}^s) - P(x_\star)] \leq \frac{1}{M_s}\left[D_1 + \frac{\log(M_s/m_0)}{\log(\omega)} D_2\right]$$

$$\mathbb{E}[P(\bar{x}^s) - P(x_\star)] \geq - \frac{2D_3}{M_s}\|y_\star\|^2 - \frac{1}{M_s}\left[D_1 + \frac{\log(M_s/m_0)}{\log(\omega)} D_2\right]$$

$$\sqrt{\mathbb{E}\left[\operatorname{dist}(A(\xi)\bar{x}^s, b(\xi))^2\right]} \leq \frac{1}{M_s}\left[2D_3\|y_\star\| + 2\sqrt{D_3}\sqrt{D_1 + \frac{\log(M_s/m_0)}{\log(\omega)} D_2}\right]$$

where $D_1 = \frac{\omega}{\omega-1}\frac{m_0}{\alpha_0(m_0-1)}\frac{1}{2}\left\|x_0^0 - x_\star\right\|^2 + 2\alpha_0 m_0 \frac{\omega}{\omega-1}\sigma_f^2$, $D_2 = \frac{2m_0^2\alpha_0\omega}{(m_0-1)(\omega-1)}\left(\|A\|_{2,\infty}^2\|y_\star\|^2 + \sigma_f^2\right)$, $D_3 = 4\alpha_0 m_0 \|A\|_{2,\infty}^2 \frac{\omega}{\omega-1}$.

Similar comments to Theorem 3.2 can be made for Theorem 3.3.

4. Extensions

In this section, we present a basic extension of our framework to illustrate its flexibility. We extend our method for solving problems considered in (Ouyang and Gray, 2012):

$$\min_{x \in \mathbb{R}^d} P_u(x) := \mathbb{E}\left[f(x, \xi) + g(A(\xi)x, \xi)\right] + h(x), \quad (7)$$

where the assumptions for f and h are the same as (1) and g is not an indicator function, but is Lipschitz continuous, *i.e.*,

$$|g(x, \xi) - g(y, \xi)| \leq L_g\|x - y\|, \forall x, y \in \mathbb{R}^d, \forall \xi.$$

This assumption is equivalent to $\operatorname{dom}(g^*)$ being bounded (Bauschke et al., 2011), where g^* is the Fenchel-conjugate function of $g(\cdot, \xi)$. This special case with $h(x) = 0$ is studied in (Ouyang and Gray, 2012) with the specific assumptions in this section. Inspired by (Nesterov, 2005), it has been shown in (Ouyang and Gray, 2012), that one has the following bound for the smooth approximation of $g(\cdot, \xi)$ in the sense of (2)

$$\mathbb{E}[g(A(\xi)x, \xi)] \leq \mathbb{E}[g_\beta(A(\xi)x, \xi)] + \frac{\beta}{2}L_g^2. \quad (8)$$

We illustrate that we can couple our main results with (8) to recover similar guarantees as (Ouyang and Gray, 2012) with the addition of the nonsmooth proximable term $h(x)$.

Corollary 4.1. *Denote by $x_\star$ a solution of (7).*
(a) Under the same assumptions as Theorem 3.2, and Lipschitz continuous $g(\cdot, \xi)$, one has

$$\mathbb{E}[P_u(\bar{x}^s) - P_u(x_\star)] \leq \frac{C_1}{\sqrt{M_s}}\left[C_2 + \frac{\log(M_s/m_0)}{\log(\omega)}C_3\right] + \frac{C_4}{\sqrt{M_s}}L_g^2,$$

where the constants C_1, C_2, C_3, C_4 are defined in Theorem 3.2.
(b) Under the same assumptions as Theorem 3.3, and Lipschitz continuous $g(\cdot, \xi)$, one has

$$\mathbb{E}[P_u(\bar{x}^s) - P_u(x_\star)] \leq \frac{1}{M_s}\left[D_1 + \frac{\log(M_s/m_0)}{\log(\omega)}D_2\right] + \frac{D_3}{M_s}L_g^2,$$

where the constants D_1, D_2, D_3 are defined in Theorem 3.3.

Lastly, we can combine the problem template in (1) with (7) to arrive at the problem

$$\min_{x \in \mathbb{R}^d} \mathbb{E}\left[f(x, \xi) + g_1(A_1(\xi)x, \xi)\right] + h(x), \qquad A_2(\xi)x \in b(\xi), \xi\text{-almost surely},$$

where $g_1(\cdot, \xi)$ is Lipschitz continuous and we have the same assumptions as (1) for almost sure constraints. Arguments in Corollary 4.1 can be combined in a straightforward way with our results from Section 3 for solving this template.

5. Related Works

The most prominent work for stochastic optimization problems is stochastic gradient descent (SGD) (Robbins and Monro, 1951; Nemirovski et al., 2009; Moulines and Bach, 2011; Polyak and Juditsky, 1992). Even though SGD is very well studied, it only applies when there does not exist any constraints in the problem template (1). For the case of simple constraints, *i.e.* $h(x) = \delta_{\mathcal{K}}(x)$ in (1) and almost sure constraints are not present, projected SGD can be used (Nemirovski et al., 2009). However, it requires $\mathcal{K}$ to be a projectable set, which does not apply to the general template of (1). In the case where $h(x)$ in (1) is a nonsmooth proximable function (Rosasco et al., 2014) studied the convergence of stochastic proximal gradient (SPG) method which utilizes stochastic gradients of $f(x, \xi)$ in addition to the proximal operator of $h(x)$. This method generalizes projected SGD, however, they cannot handle infinitely many constraints that we consider in (1) since it is not possible to project onto their intersection in general.

A line of work that is known as alternating projections, focuses on applying random projections for solving problems that are involving the intersection of infinite number of sets. In particular, these methods focus on the following template

$$\min_{x\in\mathbb{R}^d} \mathbb{E}\left[f(x,\xi)\right]: \quad x\in\mathcal{B}(:=\cap_{\xi\in\Omega}\mathcal{B}(\xi)). \qquad (9)$$

Here, the feasible set $\mathcal{B}$ consists of the intersection of a possibly infinite number of convex sets. The case when $f(x,\xi)=0$ which corresponds to the convex feasibility problem is studied in (Necoara et al., 2018b). For this particular setting, the authors combine the smoothing technique with minibatch SGD, leading to a stochastic alternating projection algorithm having linear convergence.

The most related to our work is (Patrascu and Necoara, 2017) where the authors apply a proximal point type algorithm with alternating projections. The main idea behind (Patrascu and Necoara, 2017) is to apply smoothing to $f(x,\xi)$ and apply stochastic gradient steps to the smoothed function, which results in a stochastic proximal point type of update, combined with alternating projection steps. The authors show $\mathcal{O}(1/\sqrt{k})$ rate for general convex, and $\mathcal{O}(1/k)$ rate for smooth and strongly convex objectives. For strongly convex objectives, (Patrascu and Necoara, 2017) requires smoothness of the objective which renders their results not applicable to our composite objective function in (1). In addition, they require strong convexity of the objective function while our results are valid for a more relaxed strong convexity assumption. Lastly, (Patrascu and Necoara, 2017) assumes the projectability of individual sets, whereas in our case, the constraints $A(\xi)x\in b(\xi)$ might not be projectable unless $A(\xi)$ and $b(\xi)$ are of very small dimension since the projection involves solving a linear system at each iteration.

Stochastic forward-backward algorithms can also be applied to solve (1). However, the papers introducing those very general algorithms focused on proving convergence and did not present convergence rates (Bianchi, 2015; Bianchi et al., 2017; Salim, 2018). There are some other works that focus on (9) (Wang et al., 2015; Mahdavi et al., 2013; Yu et al., 2017) where the authors assume the number of constraints is finite, which is more restricted than our setting. Other related works are (Xu, 2018; Mishchenko and Richtárik, 2018) where the number of constraints are finite in (1).

Semi-infinite linear programming (Reemtsen and Rückmann, 1998) also deals with problems with infinitely many constraints with a different approach. The classical approach is to select finitely many constraints, for instance by column generation, and to solve a sequence of problems with this selection of constraints. The works (Lin et al., 2017; Wei et al., 2018) focus on inexact primal dual algorithms for semi-infinite programs. The methods in these papers require bounded primal domain, continuously varying constraints and the knowledge of parameters such as the upper bound on the norm of dual variables which is not known in general. Lastly, they do not have fast rates for strongly convex case.

In the case where the number of constraints in (1) is finite and the objective function is deterministic, Nesterov's smoothing framework is studied in (Tran-Dinh et al., 2018b; Van Nguyen et al., 2017; Tran-Dinh et al., 2018a) in the setting of accelerated proximal gradient methods. These methods obtain $\mathcal{O}(1/k)$ ($\mathcal{O}(1/k^2)$) rate when the number of constraints is finite and $F(x)$ is a (strongly) convex function whose gradient ∇F can be computed.

In (Ouyang and Gray, 2012), the authors apply Nesterov's smoothing to (7). However, this approach does not apply to (1), due to the Lipschitz continuous assumption on $g(\cdot,\xi)$. Note that in our main template (1), $g(\cdot,\xi)=\delta_{b(\xi)}(\cdot)$, which is not Lipschitz continuous on $\mathbb{R}^d$.

6. Numerical Experiments

We present numerical experiments on a basis pursuit problem on synthetic data, a hard margin SVM problem on the `kdd2010`, `rcv1`, `news20` datasets from (Chang and Lin, 2011) and a portfolio optimization problem on `NYSE`, `DJIA`, `SP500`, `TSE` datasets from (Borodin et al., 2004).

6.1. Sparse regression with basis pursuit on synthetic data

In this section, we consider the basis pursuit problem which is widely used in machine learning and signal processing applications (Donoho, 2006; Arora et al., 2018):

$$\begin{aligned}\min_{x\in\mathbb{R}^d}\ & \|x\|_1 \\ \text{st: } & a^\top x = b, a.s.\end{aligned} \qquad (10)$$

where $a\in\mathbb{R}^d$, $b\in\mathbb{R}$. We consider the setting where the measurements a arrive in a streaming fashion, similar to (Garrigues and Ghaoui, 2009). For generating the data, we defined Σ as the matrix such that $\Sigma_{i,j}=\rho^{|i-j|}$ with $\rho=0.9$. We generated a random vector $x^*\in\mathbb{R}^d$, $d=100$ with 10 nonzero coefficients and independent $\mathcal{N}(0,\Sigma)$ random variables a_i which are then centered and normalized. We also define $b_i=a_i^\top x^*$. Because of the centering, there are multiple solutions to the infinite system $a^\top x=b$ a.s., and we wish to recover x^* as the solution of the basis pursuit problem (10). We compare SASC (Algorithm 1), SGD (Nemirovski et al., 2009) and SPP (Patrascu and Necoara, 2017). We manually tuned the step sizes for the methods and included the best obtained results. Since the basis pursuit problem does not possess (restricted) strong convexity, we use the parameters from Case 1 in SASC and a fixed step size μ for SPP which is used for the analysis in Corollary 6 in (Patrascu and Necoara, 2017). We used the parameters $\mu=10^{-5}$ for SPP, $m_0=2$, $\omega=2$, $\alpha_0=10^{-2}\|a_1b_1\|_\infty$, where a_1 is the first measurement and b_1 is the corresponding result. We take $n=10^5$ and make two passes over the

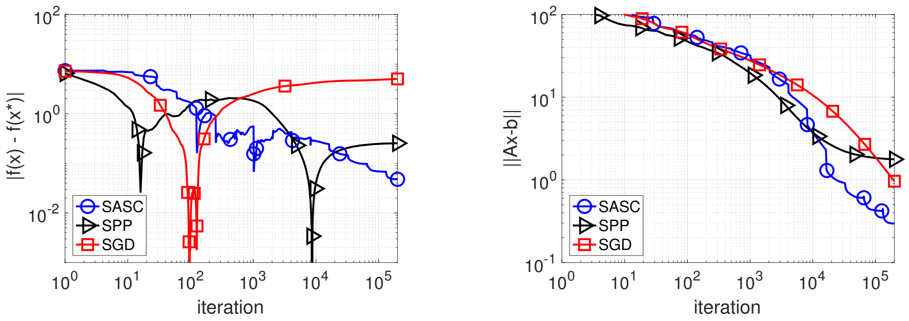

Figure 1. Performance of SGD, SPP and SASC on synthetic basis pursuit problem.

data. Figure 1 illustrates the behaviour of the algorithms for the synthetic basis pursuit problem. We can observe that SASC does exhibit a $\tilde{O}(1/\sqrt{k})$ convergence in feasibility and objective suboptimality. The stair case shape of the curves comes from the double-loop nature of the method. SPP can also solve this problem since the projection onto a hyperplane is easy to do when the constraints are processed one by one. As observed in Figure 1, SPP reaches to that accuracy almost as fast as SASC, however, it stagnates once it reaches the pre-determined accuracy since the fixed step size μ determines the accuracy that the algorithm will reach. We also tried running SGD on $\min_x \frac{1}{2}\mathbb{E}(\|a^\top x - b\|_2^2)$ but this leads to non-sparse solutions, therefore SGD converges to another solution than SASC and SPP.

A common technique that is used in stochastic optimization is to use mini-batches to parallelize and speed up computations. Since SPP utilizes projections at each iteration, it needs to project onto linear constraints each iteration. When the data is processed in mini-batches, this will require matrix inversions of sizes equal to mini-batches. On the other hand, SASC can handle mini-batches without any overhead.

6.2. Portfolio optimization

In this section, we consider Markowitz portfolio optimization with the task of maximizing the expected return given a maximum bound on the variance (Abdelaziz et al., 2007). The precise formulation we consider is the following:

$$\min_{x \in \mathbb{R}^d} -\langle a_{avg}, x\rangle : \sum_{i=1}^{d} x_i = 1 \qquad (11)$$
$$|\langle a_i - a_{avg}, x\rangle| \leq \epsilon, \forall i \in [1, n],$$

where short positions are allowed and $a_{avg} = \mathbb{E}[a_i]$ is assumed to be known.

This problem fits to our template (1), with a deterministic objective function, $2n$ linear constraints and one indicator function for enforcing $\sum_{i=1}^{d} x_i = 1$ constraint.

We implement SASC and SPP from (Patrascu and Necoara, 2017). Since the structure of (11) does not have any restricted strong convexity due to linear objective function, we are applying the general convex version of SPP, which suggests setting a smoothness parameter μ depending on the final accuracy we would like to get as also discussed in basis pursuit problem. We run SPP with two different μ values 10^{-1} and 10^{-2}. We run SASC with the parameters $\alpha_0 = 1, \omega = 1.2, m_0 = 2$ and Case 1 in Algorithm 1. We use NYSE ($d = 36, n = 5651$), DJIA ($d = 30, n = 507$), SP500 ($d = 25, n = 1276$) and TSE ($d = 88, n = 1258$) where d is the number of stocks and n is the number of days for which the data is collected and we set ϵ in (11) to be 0.2. These datasets are also used in (Borodin et al., 2004).

We compute the ground truth using `cvx` (Grant et al., 2008) and plotted the distance of the iterates of the algorithms to the solution $\|x - x^\star\|$. We include the plots for convergence in terms of objective value and feasibility in supplementary material, Section 7.3. We compile the results in Figure 4.

We can observe the behaviour of SPP from Figure 4 for different step size values μ. Larger μ causes a fast decrease in the beginning, however, it also affects the accuracy that the algorithm is going to reach. Therefore, large μ has the problem of stagnating at a low accuracy. Smaller μ causes SPP to reach to higher accuracies at the expense of slower initial behaviour. SASC has a steady behaviour since it does not have a parameter depending on the final accuracy. It removes the necessity of tuning μ in SPP, as we can observe the steady decrease of SASC throughout, beginning from the initial stage of the algorithm.

6.3. Primal support vector machines without regularization parameter

In this section, we consider the classical setting of binary classification, with a small twist. For the standard setting, given a training set $\{a_1, a_2, \ldots, a_n\}$ and labels $\{b_1, b_2, \ldots, b_n\}$, where $a_i \in \mathbb{R}^p, \forall i$ and $b_i \in [-1, +1]$ the aim is to train a model that will classify the correct labels for the unseen examples.

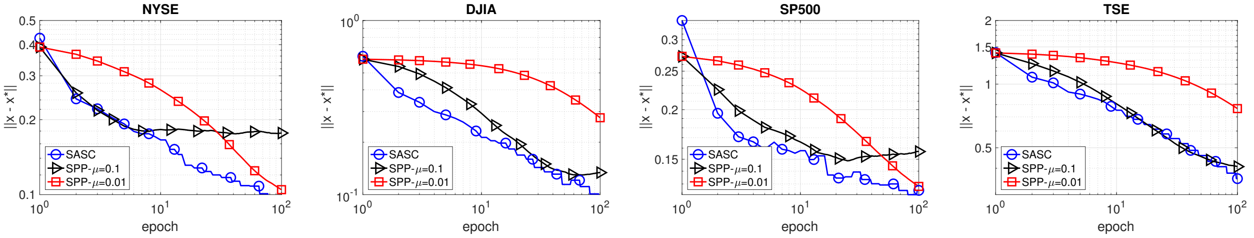

Figure 2. Performance of SASC and SPP on portfolio optimization for four different datasets. Objective value and feasibility plots are illustrated in Section 7.3 of the supplementary material.

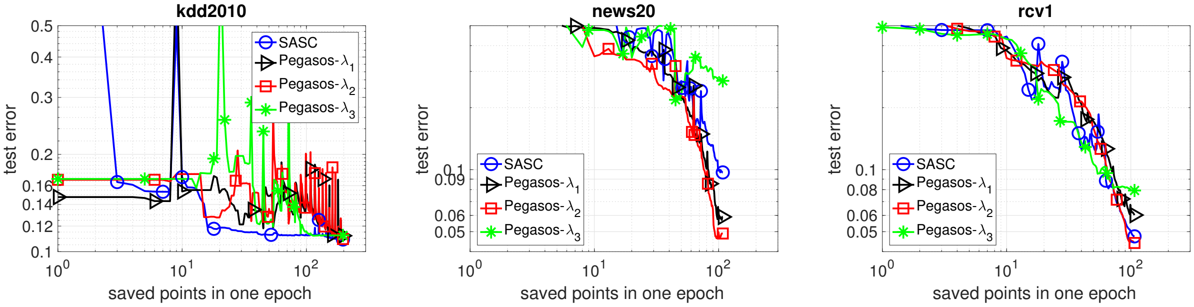

Figure 3. Performance of SASC and Pegasos on SVM for three different datasets.

Primal hard margin SVM problem is

$$\min_{x\in\mathbb{R}^d} \frac{1}{2}\|x\|^2 : b_i\langle a_i, x\rangle \geq 1, \forall i. \tag{12}$$

Since this problem does not have a solution unless the data is linearly separable, the standard way is to relax the constraints, and solve the soft margin SVM problem with hinge loss instead:

$$\min_{x\in\mathbb{R}^d} \frac{1}{2}\|x\|^2 + C\sum_{i=1}^{n} \max\{0, 1 - b_i\langle a_i, x\rangle\}, \tag{13}$$

where C has the role of a regularization parameter to be tuned. The choice for C has a drastic effect on the performance of the classifier as also been studied in the literature (Hastie et al., 2004). It is known that poor choices of C may lead to poor classification models.

We are going to have a radically different approach for the SVM problem. Since the original formulation (12) fits to our template (1), we can directly apply SASC to this formulation. Even though the hard margin SVM problem does not necessarily have solution, applying SASC to (12) corresponds to solving a sequence of soft margin SVM problems with squared hinge loss, with changing regularization parameters. The advantage of such an approach will be that there will be no necessity for a regularization parameter C since this parameter will correspond to $\frac{1}{\beta}$ in our case where β is the smoothness parameter, for which we have theoretical guideline from our analysis.

We compare SASC with Pegasos algorithm (Shalev-Shwartz et al., 2011) which solves (13) by applying stochastic subgradient algorithm. Since the selection of the regularization parameter C effects the performance of the model, we use 3 different values for the λ, namely $\{\lambda_1, \lambda_2, \lambda_3\} = \{10^{-3}/n, 1/n, 10^3/n\}$. We use the following datasets from libsvm database (Chang and Lin, 2011): `kdd2010 raw version (bridge to algebra)` with $19,264,997$ training examples, $748,401$ testing examples and $1,163,024$ features, `rcv1.binary` with $20,242$ training examples, $677,399$ testing examples and $47,236$ features. For the last dataset, `news20.binary`, since there was not a dedicated testing dataset, we randomly split examples for training and testing with 17.996 training examples, $2,000$ testing examples and $1,355,191$ features. For SASC, we use $\alpha_0 = 1/2$, $\omega = 2$ in all experiments and use the parameter choices in Case 2 in Algorithm 1 due to strong convexity in the objective. We computed the test errors for one pass over the data and compile the results in Figure 3.

We illustrate the performance of SASC and Pegasos in Figure 3. SASC seems to be comparable to Pegasos for different regularization parameters. As can be seen in Figure 3, Pegasos performs well for good selection of the regularization parameter. However, when the parameter is selected incorrectly, it might stagnate at a high test error which can be observed in the plots. On the other hand, SASC gets comparable, if not better, performance without the need to tune regularization parameter.

Acknowledgements

The work of A. Alacaoglu and V. Cevher was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no 725594 - time-data) and by the Swiss National Science Foundation (SNSF) under grant number 200021_178865/1. The work of I. Necoara was supported by UEFISCDI, Romania, PNIII-P4-PCE-2016-0731, project ScaleFreeNet, no. 39/2017.

References

F. B. Abdelaziz, B. Aouni, and R. El Fayedh. Multi-objective stochastic programming for portfolio selection. *European Journal of Operational Research*, 177(3):1811–1823, 2007.

A. Agarwal, M. J. Wainwright, P. L. Bartlett, and P. K. Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

S. Arora, M. Khodak, N. Saunshi, and K. Vodrahalli. A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and lstms. In *Proc. of the 6th International Conference on Learning Representations*, 2018.

H. H. Bauschke, P. L. Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

P. Bianchi. A stochastic proximal point algorithm: convergence and application to convex optimization. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*, pages 1–4. IEEE, 2015.

P. Bianchi, W. Hachem, and A. Salim. A constant step forward-backward algorithm involving random maximal monotone operators. *arXiv preprint arXiv:1702.04144*, 2017.

J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017.

A. Borodin, R. El-Yaniv, and V. Gogan. Can we learn to beat the best stock. In *Advances in Neural Information Processing Systems*, pages 345–352, 2004.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

P. Garrigues and L. E. Ghaoui. An homotopy algorithm for the lasso with online observations. In *Advances in neural information processing systems*, pages 489–496, 2009.

M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, 2008.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(Oct):1391–1415, 2004.

G. Lan and R. D. Monteiro. Iteration-complexity of first-order penalty methods for convex programming. *Mathematical Programming*, 138(1-2):115–139, 2013.

G. Lan and Z. Zhou. Algorithms for stochastic optimization with expectation constraints. *arXiv preprint arXiv:1604.03887*, 2016.

Q. Lin, S. Nadarajah, and N. Soheili. Revisiting approximate linear programming using a saddle point based reformulation and root finding solution approach. *optimization-online preprint*, 2017.

M. Mahdavi, T. Yang, and R. Jin. Stochastic convex optimization with multiple objectives. In *Advances in Neural Information Processing Systems*, pages 1115–1123, 2013.

K. Mishchenko and P. Richtárik. A stochastic penalty model for convex and nonconvex optimization with big constraints. *arXiv preprint arXiv:1810.13387*, 2018.

E. Moulines and F. R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

I. Necoara, Y. Nesterov, and F. Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, doi: 10.1007/s10107-018-1232-1, 2018a.

I. Necoara, P. Richtarik, and A. Patrascu. Randomized projection methods for convex feasibility problems: conditioning and convergence rates. *arXiv preprint arXiv:1801.04873*, 2018b.

A. Nedić, A. Olshevsky, and M. G. Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5): 953–976, 2018.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

H. Ouyang and A. Gray. Stochastic smoothing for nonsmooth minimizations: Accelerating SGD by exploiting structure. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 33–40, 2012.

A. Patrascu and I. Necoara. Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization. *The Journal of Machine Learning Research*, 18(1):7204–7245, 2017.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

R. Reemtsen and J.-J. Rückmann. *Semi-infinite programming*, volume 25. Springer Science & Business Media, 1998.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

L. Rosasco, S. Villa, and B. C. Vũ. Convergence of stochastic proximal gradient algorithm. *arXiv preprint arXiv:1403.5074*, 2014.

A. Salim. *Random monotone operators and application to stochastic optimization*. PhD thesis, Université Paris-Saclay, 2018.

S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7(Jul):1531–1565, 2006.

Z. J. Towfic and A. H. Sayed. Stability and performance limits of adaptive primal-dual networks. *IEEE Transactions on Signal Processing*, 63(11):2888–2903, 2015.

Q. Tran-Dinh, A. Alacaoglu, O. Fercoq, and V. Cevher. An adaptive primal-dual framework for nonsmooth convex minimization. *arXiv preprint arXiv:1808.04648*, 2018a.

Q. Tran-Dinh, O. Fercoq, and V. Cevher. A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *SIAM Journal on Optimization*, 28 (1):96–134, 2018b.

P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. 2008. submitted to SIAM J. Optim.

Q. Van Nguyen, O. Fercoq, and V. Cevher. Smoothing technique for nonsmooth composite minimization with linear operator. *arXiv preprint arXiv:1706.05837*, 2017.

M. Wang, Y. Chen, J. Liu, and Y. Gu. Random multi-constraint projection: Stochastic gradient methods for convex optimization with many constraints. *arXiv preprint arXiv:1511.03760*, 2015.

B. Wei, W. B. Haskell, and S. Zhao. An inexact primal-dual algorithm for semi-infinite programming. *arXiv preprint arXiv:1803.10898*, 2018.

Y. Xu. Primal-dual stochastic gradient method for convex programs with many functional constraints. *arXiv preprint arXiv:1802.02724*, 2018.

H. Yu, M. Neely, and X. Wei. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems*, pages 1428–1438, 2017.