Random Multi-Constraint Projection: Stochastic Gradient Methods for Convex Optimization with Many Constraints*

Mengdi Wang†, Yichen Chen‡

April 13, 2017

Abstract

Consider stochastic convex optimization subject to a large number of constraints. We focus on stochastic problems in which the objective takes the form of expected values and the feasible set is the intersection of a large number of convex sets. We propose a class of algorithms that perform both stochastic gradient descent and random feasibility updates simultaneously. At every iteration, the algorithms sample a number of projection points onto a randomly selected small subsets of all constraints. Three feasibility update schemes are considered: averaging over random projected points, projecting onto the most distant sample, projecting onto a special polyhedral set constructed based on sample points. We prove the almost sure convergence of these algorithms, and analyze the feasibility and optimality errors of the iterates. We provide new convergence rate benchmarks for stochastic first-order optimization with many constraints. Numerical experiments are conducted to validate our theoretical results, which suggest that the algorithm using the polyhedral-set projection scheme outperforms the others.

1 Introduction

Consider the optimization problem

$$\begin{aligned} &\text{minimize} \quad \Big\{F(x) = \mathbf{E}[f(x;v)]\Big\} \\ &\text{subject to} \quad x \in \mathcal{X} = \mathcal{X}_0 \cap (\cap_{i=1}^m \mathcal{X}_i)\,, \end{aligned} \tag{1}$$

where $F : \Re^n \mapsto \Re$ is a continuous function, $f(\cdot; v) : \Re^n \mapsto \Re$ is a function parameterized by v, $\mathcal{X}_0$ is compact and convex, $\mathcal{X}_i$ are closed and convex sets in $\Re^n$, v is a random variable, m is the total number of constraints (which is potentially a large number), and the expectation $\mathbf{E}[\cdot]$ is taken over the distribution of v. *Throughout this paper, we assume that F is a convex function but not necessarily smooth.* We denote by x^* an arbitrary optimal solution to problem (1).

*This work is supported by National Science Foundation Grant DMS-1619818 and CMMI-1653435.

†Mengdi Wang is with Department of Operations Research and Financial Engineering, Princeton University, Princeton 08544, USA.

‡Yichen Chen is with Department of Computer Science, Princeton University, Princeton 08544, USA.

The objective F is an expectation of random functions, which models empirical average of loss functions over static data set or some expected loss function over streaming unknown data. The function F is not available to us directly. Instead, we are given a sampling oracle that allows us to query for random realizations of the subgradients of F. The compact set $\mathcal{X}_0$ models one's prior knowledge about a bounded set in which one aims to search for the optimal solution. It is usually a big box constraint or a big Euclidean ball, which is often implicitly assumed in the machine learning literature. Problem (1) applies to optimization problems with a large system of inequality constraints, for example,

$$\mathcal{X}_0 = \{x \mid \|x\|_2 \leq R\}, \qquad \mathcal{X}_i = \{g_i(x) \leq 0\}, \qquad i = 1, \ldots, m.$$

When m is a large number, it is often inefficient or even impossible to operate over $\mathcal{X}$ directly. Instead, the sampling oracle is able to sample a small subset of $\{\mathcal{X}_i\}_{i=1}^m$ and perform projection onto individual sample constraints. We focus on situations where each individual set $\mathcal{X}_i$ is simple such that computing the projection is efficient.

Problem (1) is a canonical problem frequently arising from large-scale computing, stochastic optimization, machine learning, estimation and filtering. In the case where $\mathcal{X} = \Re^n$ or $\mathcal{X}$ is a simple set, *stochastic gradient descent* (SGD) is a popular method and has been studied extensively. SGD updates incrementally according to

$$x_{k+1} = \Pi_{\mathcal{X}} \left\{ x_k - \alpha_k g(x_k, v_{k+1}) \right\},$$

where $g(x_k, v_{k+1})$ is a sample realization of a subgradient of F at x_k and α_k is a positive stepsize. SGD is one of the most important stochastic first-order methods and has drawn significant attention from various communities. Theoretically, it has been shown that after k samples/iterations, the average of the iterates has $\mathcal{O}(1/k)$ optimization error for strongly convex problems, and $\mathcal{O}(1/\sqrt{k})$ error for general convex problems (see Nemirovski et al. [NJLS08]; also see Moulines and Bach [MB11], Rakhlin et al. [RSS12], Shamir and Zhang [SZ13]). These convergence rates match the corresponding information-theoretic lower bounds (see e.g., Agarwal et al. [ABRW12] and Nemirovski and Yudin [NY83]), suggesting that SGD is non-improvable with respect to the sample size. Practically, SGD is a fast algorithm that is able to process one data entry per iteration. It can be implemented as a distributed or parallel algorithm to process large-scale data sets. It can also be implemented as an online algorithm to process streaming data. The nice theoretical property (e.g., the non-improvable rate of convergence) and flexibility of implementation makes SGD one of the most popular methods for many machine learning applications. It has motivated extensive theoretical research as well as many new algorithms for specific applications (see [RNV09,GL12,GL13,SZ13] for examples).

For constrained optimization, SGD often becomes inefficient when $\mathcal{X}$ is a complicated set like $\mathcal{X} = \cap_{i=1}^m \mathcal{X}_i$. In particular, each iteration of SGD requires calculating the Euclidean projection onto the feasible set $\mathcal{X}$, which can be computationally expensive. This is a serious limitation. Although $\mathcal{X} = \cap_{i=1}^m \mathcal{X}_i$ is hard to work with, it is often convenient to project a vector x onto a single constraint

set $\mathcal{X}_i$. This has motivated the *random projection algorithm*, taking the form

$$x_{k+1} = \Pi_{\mathcal{X}_{\omega_k}}\{x_k - \alpha_k g(x_k, v_{k+1})\}, \tag{2}$$

where $\Pi_{\mathcal{X}_{\omega_k}}$ denotes the projection on $\mathcal{X}_{\omega_k}$ and ω_k is a random variable taking value in $\{1, \ldots, m\}$. At each iteration, algorithm (2) randomly picks one out of all constraint sets and finds the projection onto it. When the feasible set $\mathcal{X}$ is the intersection of many simple sets (e.g., half spaces determined by linear inequalities), algorithm (2) is able to solve large-scale problems by fast incremental updates. The idea of incremental projection is also related to a line of works on the feasibility problem, which is to find a common point in the intersection of many convex sets (see von Neumann [Neu50], Halperin [Hal62], Gubin et al. [GPR67], Tseng [Tse90], Bauschke et al. [BBL97], Deutsch and Hundal [DH06a], [DH06b], [DH08], Cegielski and Suchocka [CS08], Lewis and Malick [LM08], Leventhal and Lewis [LL10], and Nedić [Ned10]).

Optimization algorithms using random feasibility updates were first considered by Nedić [Ned11], and were later studied by Wang and Bertsekas in [WB15] in the context of stochastic variational inequalities. A recent paper [WB16] studied the random projection algorithm (2) and its proximal variant, and provided a unified analytic framework for its almost sure convergence. There remain several open questions. First, a comprehensive convergence rate analysis addressing various situations (such as the case of strongly convex optimization) is yet to be established. Second, it is not clear how the constraint randomization scheme affects the algorithms' efficiency. Third, it would be interesting to design new algorithms that make more efficient use of the sample gradients and sample constraints.

The aim of this paper is to gain a deeper understanding of constraint randomization and to develop faster algorithms that make smart use of random constraint projections. The existing method uses one random projection per iteration, which may induce large oscillation in the iterates and cause the convergence to be slow. For this reason, we propose new algorithms that sample multiple constraint supersets per iteration. Three feasibility update schemes are considered. We are particularly interested in obtaining tight theoretical guarantees of the algorithms' performance. In addition, we pay special attention to the efficiency of various feasibility update schemes.

The contributions of this paper are three-folded.

(i) We propose a new class of *random multi-constraint projection algorithms* that incrementally process the stochastic subgradients as well as sample constraint projections. It contains the existing algorithm (2) as a special case. In particular, we consider three algorithms with different feasibility update steps. The first algorithm averages over multiple random projections. The second algorithm projects the iterate onto the most distant set out of all the sample sets. The third algorithm constructs a new polyhedral set based on the sample projected points and projects the iterates onto the polyhedral set. Although the three methods use the sample projected points in different ways, all their iterations are easy to implement and computationally efficient.

(ii) We provide a unified analysis that establishes convergence and rate of convergence for the new

class of algorithms. We analyze their performances in two aspects: *convergence of feasibility error* and *convergence of optimality error*. The two error sequences are entangled together, which complicates the supermartingale analysis. We provide estimates of the convergence rates to illustrate the efficiency of various random feasibility update schemes. The convergence rate estimates are tight with respect to the number of (sub)gradient samples (up to constant factors). A summary of the convergence rate results is given in Table 1. They provide comprehensive rate estimates for random projection algorithms. Comparing the results of Table 1 with the convergence rates of SGD, we discover an interesting phenomenon: *Constraint randomization does not slow down the convergence of stochastic first-order methods (up to constant factors).* In other words, the random projection algorithms enjoy almost the same practical advantages and theoretical guarantees as the SGD method.

	Convex Optimization	σ-Strongly Convex Optimization
Optimality Error	$\mathcal{O}\left(\frac{DB/\sqrt{C}}{\sqrt{k}}\right)$	$\mathcal{O}\left(\frac{B^2/C}{\sigma^2}\cdot\frac{\log k+1}{k}\right)$
Feasibility Error	$\mathcal{O}\left(\frac{D^2}{C}\cdot\frac{1}{k}\right)$	$\mathcal{O}\left(\frac{B^2}{C^2\sigma^2}\cdot\frac{1}{k^2}\right)$

Table 1: Convergence rates of random multi-constraint projection algorithms. The number k is the iteration number as well as the number of sampled stochastic (sub)gradients. The constant B is a stochastic analog of the Lipschitz continuity. The parameter C is related to specific feasibility update schemes, whose best value is $C = M\eta/m$ in our analysis. Here m is the total number of constraints, M is the constraint sample size per iteration, and η is a regularity constant determined by the spatial distribution of all constraint sets.

(iii) We show that by sampling multiple constraints instead of sampling a single constraint, the rate of convergence (as well as the sample efficiency of stochastic gradients) is improved significantly. Consider the three feasibility update schemes: the averaging scheme, the max-distance-set scheme, and the polyhedral-set scheme. The theoretical and numerical results show that the polyhedral-set scheme has the best performance among the three.

The remainder of this paper is organized as follows. In Section 2, we propose the class of random multi-constraint projection algorithms with three feasibility update schemes. In Section 3, we show that the convergence of these algorithms is an interplay of a feasibility improvement process and an optimality improvement process, and we prove the almost sure convergence of all three algorithms. In Section 4, we study the rates of convergence of the feasibility error and the optimality error, respectively. In Section 5, we provide numerical experiments; and in Section 6, we draw the conclusions.

Notations All vectors are considered as column vectors. For a vector $x \in \Re^n$, we denote by x' its transpose, and denote by $\|x\| = \sqrt{x'x}$ its Euclidean norm. For a matrix $A \in \Re^{n\times n}$, we denote by $\|A\| = \max\{\|Ax\| \mid \|x\| = 1\}$ its induced Euclidean norm. For two sequences $\{a_k\}, \{b_k\}$, we

denote by $a_k = O(b_k)$ if there exists $c > 0$ such that $\|a_k\| \leq c\|b_k\|$ for all k. For a set $\mathcal{X} \subset \Re^n$ and vector $y \in \Re^n$, we denote by

$$\Pi_{\mathcal{X}}\{y\} = \text{argmin}_{x\in\mathcal{X}}\|y - x\|^2$$

the Euclidean projection of y onto $\mathcal{X}$, where the minimization is always uniquely attained if $\mathcal{X}$ is nonempty, convex and closed. For a function $f(x)$, we denote by $\nabla f(x)$ its gradient at $\mathcal{X}$ if f is differentiable, denote by $\partial f(x)$ its subdifferential at $\mathcal{X}$, and denote by $\widetilde{\nabla} f(x)$ some subgradient at $\mathcal{X}$ (to be specified in the context).

2 Random Multi-Constraint Projection Algorithms

In this section, we propose three random multi-projection algorithms, which are described in Algorithms 1, 2, and 3, respectively. They update using stochastic (sub)gradients and random projections onto a small subset of constraints. Suppose that we are given a *Sampling Oracle* ($\mathcal{SO}$) such that

- Given $x \in \Re^n$, the $\mathcal{SO}$ returns a random subgradient $g(x, v)$ of the objective function F.
- Given $x \in \Re^n$ and integer $M > 0$, the $\mathcal{SO}$ returns M random projected points $\{\Pi_{\mathcal{X}_{\omega_1}}(x), \ldots, \Pi_{\mathcal{X}_{\omega_M}}(x)\}$ where the sample sets are drawn uniformly without replacement.

Here v, ω are independent random variables, and $M \ll m$ is a positive integer. Note that the $\mathcal{SO}$ only returns the projected point $\Pi_{\mathcal{X}_\omega}(x)$, not the constraint set $\mathcal{X}_\omega$ itself. *We assume throughout that random variables generated by different calls to the $\mathcal{SO}$ are independent and identically distributed.*

The proposed algorithms update the iterates $\{x_k\}$ while interacting with the $\mathcal{SO}$. Each iteration alternates between two steps: an optimality update step (stochastic gradient descent), and a feasibility update step (random projections). The three algorithms considered in this paper use the same optimality update step, which is a straightforward stochastic gradient descent step. They differ from one another in their feasibility update steps. See Figure 1 for graphical illustrations of the three algorithms.

Algorithm 1 takes an average of multiple random projected points. It reduces to the known algorithm (2) when $M = 1$. As illustrated by Figure 1(a), the averaging step largely prevents the next iterate x_{k+1} from randomly jumping between distant constraint sets. While improving the stability of iterates, the averaging scheme is computationally efficient, as calculating each random projection still involves only a single set.

Algorithm 2 chooses the most distant set out of the sample constraints. By comparing the distances between the projected points and the original point, the algorithm can easily identify the most distant set out of all samples. Then it updates by setting the next iterate to be the most distant projected point. See Figure 1(b) for a graphical visualization of Algorithm 2. By projecting onto the most distant constraint, it guarantees larger improvement towards the feasible set than Algorithm 1. By using only the most distant set, we discard the less important information related to the other sets. We are interested how efficient the scheme is.

Algorithm 1: Random Averaging Projection Method

Input: $x_0 \in \mathcal{X}_0$, $\mathcal{SO}$, integer $M > 0$, stepsizes $\{\alpha_k\} \subset \Re^+$.

for $k = 0, 1, 2, \ldots$ **do**

(1) Sample a stochastic subgradient $g(x_k, v_{k+1})$ from the $\mathcal{SO}$ and update by

$$y_{k+1} = x_k - \alpha_k g(x_k, v_{k+1}) \ ;$$

(2) Sample M projections $\{\Pi_{\mathcal{X}_{\omega_{k+1,1}}} y_{k+1}, \ldots, \Pi_{\mathcal{X}_{\omega_{k+1,M}}} y_{k+1}\}$ from the $\mathcal{SO}$ and update by

$$x_{k+1} = \Pi_{\mathcal{X}_0} \left[\frac{1}{M} \sum_{i=1}^{M} \Pi_{\mathcal{X}_{\omega_{k+1,i}}} y_{k+1} \right] \ ;$$

Output: The sequences of iterates $\{x_k\}$ and $\{\widetilde{x}_k\}$ where $\widetilde{x}_k = \frac{1}{k+1} \sum_{t=0}^{k} x_t$.

Algorithm 2: Random Max-Set Projection Method

Input: $x_0 \in \mathcal{X}_0$, $\mathcal{SO}$, integer $M > 0$, stepsizes $\{\alpha_k\} \subset \Re^+$.

for $k = 0, 1, 2, \ldots$ **do**

(1) Sample a stochastic subgradient $g(x_k, v_{k+1})$ from the $\mathcal{SO}$ and update by

$$y_{k+1} = x_k - \alpha_k g(x_k, v_{k+1}) \ ;$$

(2) Sample M projections $\{\Pi_{\mathcal{X}_{\omega_{k+1,1}}} y_{k+1}, \ldots, \Pi_{\mathcal{X}_{\omega_{k+1,M}}} y_{k+1}\}$ from the $\mathcal{SO}$ and update by

$$x_{k+1} = \Pi_{\mathcal{X}_0} \left[\Pi_{\mathcal{X}_{\omega_{k+1,i^*}}} y_{k+1} \right] \ ;$$

where

$$i^* \in \text{argmax}_{i=1,\ldots,M} \| \Pi_{\mathcal{X}_{\omega_{k+1,i}}} y_{k+1} - y_{k+1} \| \ ;$$

Output: The sequences of iterates $\{x_k\}$ and $\{\widetilde{x}_k\}$ where $\widetilde{x}_k = \frac{1}{k+1} \sum_{t=0}^{k} x_t$.

Algorithm 3 utilizes the random projected points in a different way; see Figure 1(c) for an example. Given the random projected points and the corresponding normal hyperplanes, it constructs a new polyhedral set $\mathcal{P}_k$, which can be viewed as an intersection of linear cutting half spaces. This polyhedral set $\mathcal{P}_k$ is also a superset of the feasible set $\mathcal{X}$, and it is a better approximation to the unknown $\mathcal{X}$. Projection onto $\mathcal{P}_k$ involves minimizing a square function over a small set of linear inequalities. Such an extra projection step can be carried out efficiently, which, however, leads to a small computation overhead (if M is small). Theoretical and numerical results show that Algorithm 3 has the best performance among all three methods.

We left the stepsize choices unspecified in the statements of Algorithms 1,2,3. We will provide specific choices of stepsizes in the convergence rate analysis. The stepsizes will be chosen in order to minimize the optimality error bound, which is consistent with works like Rakhlin et al. [RSS12], Shamir and Zhang [SZ13] and many others.

Algorithm 3: Random Polyhedral-Set Projection Method

Input: $x_0 \in \mathcal{X}_0$, $\mathcal{SO}$, integer $M > 0$, stepsizes $\{\alpha_k\} \subset \Re^+$.

for $k = 0, 1, 2, \ldots$ **do**

(1) Sample a random subgradient $g(x_k, v_{k+1})$ and update the iterate by

$$y_{k+1} = x_k - \alpha_k g(x_k, v_{k+1}) \ ;$$

(2) Sample M random projections $\{\Pi_{\mathcal{X}_{\omega_{k+1,1}}} y_{k+1}, \ldots, \Pi_{\mathcal{X}_{\omega_{k+1,M}}} y_{k+1}\}$ from the $\mathcal{SO}$ and construct a system of linear inequities by letting

$$\mathcal{P}_{k+1} = \{x : a_i' x \leq b_i, \ \ \forall i = 1, \ldots, M\} \ ;$$

where $a_i = y_{k+1} - \Pi_{\mathcal{X}_{\omega_{k+1,i}}} y_{k+1}$, $b_i = a_i' \Pi_{\mathcal{X}_{\omega_{k+1,i}}} y_{k+1}$ for $i = 1, \ldots, M$.

(3) Calculate x_{k+1} as

$$x_{k+1} = \Pi_{\mathcal{X}_0} \left[\text{argmin}_{x \in \mathcal{P}_{k+1}} \|x - y_{k+1}\|^2 \right] \ ;$$

Output: The sequences of iterates $\{x_k\}$ and $\{\widetilde{x}_k\}$ where $\widetilde{x}_k = \frac{1}{k+1} \sum_{t=0}^{k} x_t$.

3 Coupled Convergence Process

In this section, we study the convergence of random multi-constraint projection algorithms. Each iteration of the algorithms alternates between a feasibility update step and an optimality update step. Accordingly, the convergence process can be decomposed into two coupled stochastic processes: convergence towards feasibility and convergence towards optimality. More specifically, we provide two recursive bounds for the optimality error and the feasibility error, respectively. We show that, all three algorithms converge almost surely to an optimal solution of problem (1), as long as suitable stepsizes are used.

Throughout this paper, we make the following assumptions.

Assumption 1

(a) The objective function F is convex and continuous, the sets $\mathcal{X}_i$, $i = 1, \ldots, m$, are nonempty, closed and convex, and there exists at least one optimal solution x^* to problem (1).

(b) The sample subgradients are conditionally unbiased, i.e., for any $x \in \Re^n$ and $k \geq 0$,

$$\mathbf{E}[g(x, v_0)] \in \partial F(x). \tag{3}$$

(c) There exists a scalar $B > 0$ such that

$$\mathbf{E}[\|g(x, v_0)\|^2] \leq B^2, \qquad \forall \ x \in \mathcal{X}_0.$$

(d) There exists a constant $D > 0$ such that

$$\|x - \bar{x}\| \leq D, \qquad \forall \ x, \bar{x} \in \mathcal{X}_0.$$

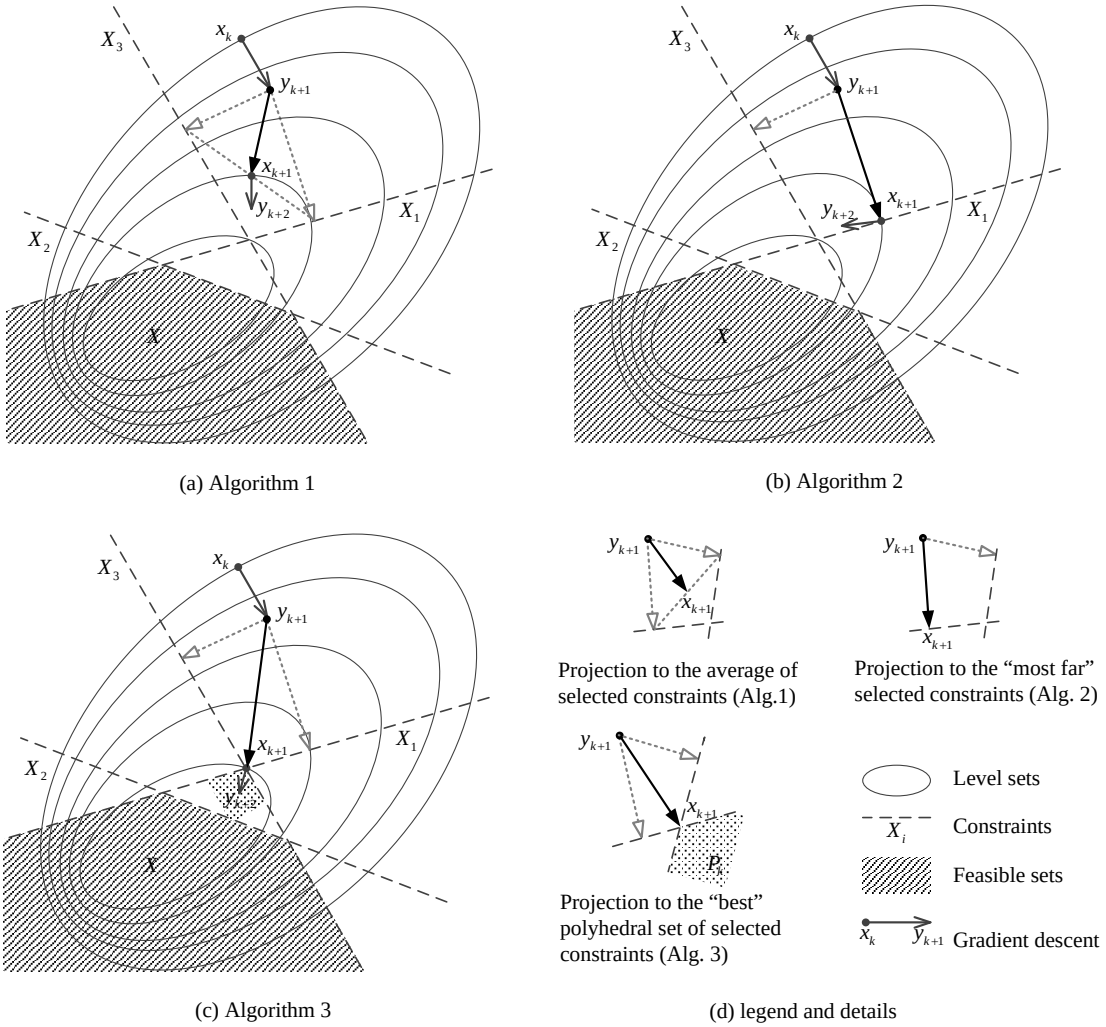


Figure 1: Graphical visualization of Algorithms 1,2,3. In this example, the intersection set is $X = X_1 \cap X_2 \cap X_3$ and the sample constraints at the current iteration are $\{X_2, X_3\}$.

Remarks on Assumption 1. The conditions are quite general for typical applications. They apply to both smooth and nonsmooth problems. We make a few remarks.

- The boundedness assumption (d) is commonly satisfied in practical problems where one wants to search for the optimal solution within a known bounded domain. The diameter D of the bounded domain $\mathcal{X}_0$ plays an important role in the convergence rate analysis (which is conventional for rate analysis of stochastic convex optimization; see for example [SZ13]).

- The condition (c) requires that the objective function has bounded gradient or subgradients over the compact domain (which implies that the objective is Lipschitz continuous over $\mathcal{X}_0$). The constant B can be viewed as a stochastic analog of the Lipschitz continuity constant.

- The conditions (b) and (c) require that the sample first-order information (either gradient or

subgradient) is unbiased and has bounded second moments. These conditions are very mild and are satisfied in the majority of sampling applications, as long as the underlying sampling distribution has reasonably light tails.

- One example that satisfies all conditions of Assumption 1 is the following stochastic least-square regression problem. It is very common in estimation and machine learning, which is given by

$$\text{minimize}_{x\in\mathcal{X}} \left\{ F(x) = \mathbf{E}_{A,b}\left[\|Ax-b\|^2\right] \right\}, \tag{4}$$

where the expectation is taken over pairs of random data points (A, b). A noisy gradient is given by $g(x, A, b) = 2A'(Ax - b)$. Then $F(x)$ satisfies the condition (a) as every realization of $\|Ax - b\|^2$ is convex. Condition (b) and (c) are also satisfied if the data pairs (A, b) are draw from a Gaussian distribution and the set $\mathcal{X}$ is compact.

Assumption 2 *There exists a constant scalar $\eta \in (0, 1)$ such that for all $x \in \Re^n$*

$$\eta\|x - \Pi_{\mathcal{X}}x\|^2 \leq \max_{i=0,\ldots,m} \|x - \Pi_{\mathcal{X}_i}x\|^2. \tag{5}$$

Remarks on Assumption 2.

- Assumption 2 is known as the *linear regularity condition*. It is related to crossing angles between the individual sets $\mathcal{X}_i$. Intuitively speaking, it requires that the sets $\mathcal{X}_i$ behave like linear sets where they intersect with one another. This property is automatically satisfied when $\mathcal{X}$ is a polyhedral set and $\mathcal{X}_i$ are linear half spaces. It has been studied by Deutsch and Hundal [DH08] in the context of the feasibility problem, where it is assumed in order to establish linear convergence of a cyclic projection method. The discussions in [Bau96] and [DH08] mention several cases where the linear regularity condition holds. These references point out that Assumption 2 is a mild restriction. Assumption 2 was also imposed in earlier works including Nedich [Ned10] and Wang and Bertsekas [WB15]. In fact, it is rare to find practical examples that do not satisfy this condition.[1]

- Assumption 2 is used to guarantee that projecting onto a random constraint set leads to sufficient decrease of the distance to feasibility. To see this, we let $\mathcal{X}_w \in \{X_1, \ldots, X_m\}$ be randomly generated by the $\mathcal{SO}$ according to a uniform distribution. Then we have

$$\mathbf{E}[\|z - \Pi_{\mathcal{X}_w}z\|^2] \geq \frac{\eta}{m}\|z - \Pi_{\mathcal{X}}z\|^2, \qquad \forall\ z \in \Re^n.$$

This lower bound will be used to establish a contractive property of the feasibility update step.

[1] We provide one such rare example: Suppose that we have two constraints, $\mathcal{X}_1 = \{(x, y) \mid y \geq x^2\}$ and $\mathcal{X}_2 = \{(x, y) \mid y \leq -x^2\}$, which intersect at the point $(0, 0)$. Let us consider the point $z = (x, 0)$. If we let x goes to zero, we have $\lim_{x\to 0} \frac{\max_{i=1,2}\|z-\Pi_{\mathcal{X}_i}z\|^2}{\|z-\Pi_{\mathcal{X}}z\|^2} \leq \lim_{x\to 0}\frac{x^4}{x^2} = 0$. So there does not exist $\eta > 0$ such that Assumption 2 is satisfied.

- We remark that it is possible to replace Assumption 2 with more general alternatives. In fact, our entire analysis applies under the following more general condition: There exists some $\rho > 0$ such that

$$\mathbf{E}[\|z - \Pi_{\mathcal{X}_w} z\|^2] \geq \rho \cdot \|z - \Pi_{\mathcal{X}} z\|^2, \qquad \forall\ z \in \Re^n.$$

  This condition requires that the $\mathcal{SO}$ finds "representable" constraint sets "on average". It can be satisfied even if there are an infinite number of constraints ($m = \infty$) and non-uniform sampling distributions are used; see also the discussion of Nedich [Ned10]. In this paper, we focus on Assumption 2 for simplicity and clarity of the presentation.

Let us summarize some basic notations and facts before analyzing the algorithms. We denote by $\widetilde{\nabla} F(x)$ the particular subgradient given by

$$\widetilde{\nabla} F(x) = \mathbf{E}[g(x, v)] \in \partial F(x).$$

For a vector $x \in \Re^n$, we denote its distance to a set Y by $\mathrm{d}(x, Y) = \|x - \Pi_Y x\|$, and denote its distance to the feasible set $\mathcal{X}$ for short by

$$\mathrm{d}(x) = \mathrm{d}(x, \mathcal{X}) = \|x - \Pi_{\mathcal{X}} x\|.$$

We define $\mathscr{F}_k$ to be the filtration generated by random variables that are revealed up to the kth iteration, i.e.,

$$\mathscr{F}_k = \left\{ v_t, \{\omega_{t,i}\}_{i=1}^M, x_t, y_t \;\middle|\; t = 1, 2, \ldots, k \right\}.$$

Note that the random variables indexed by k belong to $\mathscr{F}_k$, such as x_k. Random variables indexed by $k+1$ belong to $\mathscr{F}_{k+1}$, such as the random variables $\{v_{k+1}, \omega_{k+1,1}, \ldots \omega_{k+1,M}\}$ sampled in the $(k+1)$th iteration of the algorithm and the resulting iterates x_{k+1}, y_{k+1}.

In what follows, we study the convergence properties of Algorithms 1, 2, and 3 using a unified analysis. The analysis is developed through Theorems 1-5, which apply to *all three* algorithms. Note that the results of Theorems 1-5 involve a parameter C, which takes different values for different algorithms (to be specified in Theorem 1).

3.1 Error Decomposition and Almost Sure Convergence

There are two types of errors associated with the iterates $\{x_k\}$: the optimality error and the feasibility error. The feasibility error is usually positive because the random projection algorithms can not guarantee $\{x_k\}$ to be feasible. Let us estimate the conditional expected optimality errors and feasibility errors, i.e.,

$$\left\{ \mathbf{E}\left[\|x_{k+1} - x^*\|^2 \mid \mathscr{F}_k \right] \right\}, \qquad \left\{ \mathbf{E}[\mathrm{d}^2(x_{k+1}) \mid \mathscr{F}_k] \right\},$$

where x^* is an arbitrary optimal solution to problem (1). Our first main result establishes two recursive bounds for the optimality and feasibility errors, respectively.

Theorem 1 (Error Decomposition) *Let Assumptions 1 and 2 hold, let x^* be an arbitrary optimal solution to problem* (1)*, and let the sequence $\{x_k\}$ be generated by any of Algorithms 1, 2, or 3. Then for all $k \geq 0$, with probability 1 that*

$$\begin{aligned}\mathbf{E}[\|x_{k+1} - x^*\|^2 \mid \mathscr{F}_k] \leq & \|x_k - x^*\|^2 + 2B^2\alpha_k^2 - 2\alpha_k(F(x_k) - F(x^*)) - \frac{C}{2}\mathrm{d}^2(x_k) \\ \leq & \|x_k - x^*\|^2 + \left(2 + \frac{2}{C}\right) B^2\alpha_k^2 - 2\alpha_k(F(\Pi_{\mathcal{X}} x_k) - F(x^*)),\end{aligned} \tag{6}$$

and

$$\mathbf{E}[\mathrm{d}^2(x_{k+1})|\mathscr{F}_k] \leq \Big(1 - \frac{C}{4}\Big)\mathrm{d}^2(x_k) + \Big(2 + \frac{4}{C}\Big)B^2\alpha_k^2. \tag{7}$$

where $C = \frac{\eta}{m}$ in the case of Algorithm 1, and $C = \frac{M\eta}{m}$ in the case of Algorithms 2 or 3.

Theorem 1 gives recursive relations for the feasibility error and optimality error, respectively. The error recursions suggest that the errors "shrink" in the long run, although they do not necessarily decrease in each iteration. But when the stepsize α_k goes to zero, the optimality error decreases at a speed determined by stepsize α_k. The feasibility error decreases according to a geometric contraction with some additive error proportional to α_k^2. Indeed, the two processes $\{\|x_k - x^*\|^2\}$ and $\{\mathrm{d}^2(x_k)\}$ are entangled with each other. However, the recursive error bounds given in Theorem 1 are almost decoupled from each other. This makes it possible to study the two convergence processes separately. The proof of Theorem 1 is deferred to Section 3.2.

The second main result establishes the almost sure convergence of Algorithms 1-3. Due to the random noise, almost sure convergence of stochastic algorithms usually require that stepsizes diminish at a rate neither too slow nor too fast.

Theorem 2 (Almost Sure Convergence) *Let Assumptions 1 and 2 hold, let $\{\widetilde{x}_k\}$ be the averaged iterates generated by any of Algorithms 1, 2, or 3, and let the stepsize $\{\alpha_k\}$ satisfy*

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \qquad \sum_{k=0}^{\infty} \alpha_k^2 < \infty.$$

Then both $\{\widetilde{x}_k\}$ and $\{x_k\}$ converge almost surely to a random point in the set of optimal solutions of problem (1).

In Theorem 2, we only require a general condition on the stepsize α_k. A typical choice of the stepsize is $\alpha_k = \alpha_0 k^{-a}$ for an arbitrary $a \in (1/2, 1]$ and $\alpha_0 > 0$. Specific choices of the stepsize and the corresponding convergence rates are to be specificed in Theorems 4 and 5. The proof of Theorem 2 relies on the recursive error bounds derived in Theorem 1. The key to the proof is a coupled supermartingale convergence argument that has been used in [RS85, WB16]. We defer the formal proof to the next section.

3.2 Proofs of Theorems 1 and 2

In the rest of this section, we develop the proofs of Theorem 1 and Theorem 2 through a series of lemmas. For readers who are not concerned with the technical details, this part can be safely skipped.

Lemma 1 (Recursive Error Bounds) *Let Assumptions 1 and 2 hold, and let x^* be an arbitrary optimal solution to problem (1). Suppose that $\{x_k\}$ is the sequence of iterates generated by any of Algorithms 1, 2, or 3. Let e_{k+1} be defined as*

$$e_{k+1} = \begin{cases} \frac{1}{M}\sum_{i=1}^{M} d^2(y_{k+1}, X_{\omega_{k+1,i}}) & \text{if } \{x_k\} \text{ is generated by Algorithm 1,} \\ \max_{i=1,\ldots,M} d^2(y_{k+1}, X_{\omega_{k+1,i}}) & \text{if } \{x_k\} \text{ is generated by Algorithm 2,} \\ d^2(y_{k+1}, \mathcal{P}_{k+1}) & \text{if } \{x_k\} \text{ is generated by Algorithm 3.} \end{cases}$$

(a) With probability 1, for all $k \geq 0$,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k g(x_k, v_{k+1})'(x_k - x^*) - e_{k+1} + \alpha_k^2 \|g(x_k, v_{k+1})\|^2. \quad (8)$$

(b) With probability 1, for all $k \geq 0$ and $\epsilon > 0$,

$$\mathrm{d}^2(x_{k+1}) \leq (1+\epsilon)\mathrm{d}^2(x_k) + (1+1/\epsilon)\alpha_k^2 \|g(x_k, v_{k+1})\|^2 - e_{k+1}. \quad (9)$$

Proof. Note that $e_{k+1} \in \mathscr{F}_{k+1}$. In each case of the three algorithms, the iterate x_{k+1} takes the following form

$$x_{k+1} = \Pi_{\mathcal{X}_0}\left[\sum_{i=1}^{M} \xi_i \Pi_{Y_{k+1,i}} y_{k+1}\right],$$

where $\xi \geq 0$ for all i and $\sum_{i=1}^{M} \xi_i = 1$. In the case of Algorithm 1, we have $\xi_i = 1/M$ and $Y_{k+1,i} = X_{\omega_{k+1,i}}$ for all $i = 1, \ldots, M$. In the case of Algorithm 2, we have $\xi_i = 1$ if $i = i^*$ and $\xi_i = 0$ otherwise, and we have $Y_{k+1,i} = X_{\omega_{k+1,i}}$ for $i = 1, \ldots, M$. In the case of Algorithm 3, we have $\xi_i = 1/M$ such that $Y_{k+1,i} = \mathcal{P}_{k+1}$ for all $i = 1, \ldots, M$. In each of the three cases, $Y_{k+1,i}$ is a convex set and $\mathcal{X} \subset Y_{k+1,i}$ for all $i = 1, \ldots, M$. Moreover, we can verify that

$$\sum_{i=1}^{M} \xi_i \|y_{k+1} - \Pi_{Y_{k+1,i}} y_{k+1}\|^2 = e_{k+1}, \quad (10)$$

in each case of the three algorithms.

(a) By the nonexpansiveness of the projection $\Pi_{\mathcal{X}_0}$ and the fact $x^* \in \mathcal{X} \subset \mathcal{X}_0$, we have

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \left\| \Pi_{\mathcal{X}_0}\left[\sum_{i=1}^{M} \xi_i \Pi_{Y_{k+1,i}} y_{k+1}\right] - x^* \right\| \\ &\leq \left\| \sum_{i=1}^{M} \xi_i \Pi_{Y_{k+1,i}} y_{k+1} - x^* \right\|^2 \\ &= \|y_{k+1} - x^*\|^2 + \left\| \sum_{i=1}^{M} \xi_i \Pi_{Y_{k+1,i}} y_{k+1} - y_{k+1} \right\|^2 - 2\sum_{i=1}^{M} \xi_i (y_{k+1} - x^*)'(y_{k+1} - \Pi_{Y_{k+1,i}} y_{k+1}). \end{aligned}$$

Consider the third term on the right side. Since $x^* \in \mathcal{X} \subset Y_{k+1,i}$, we have

$$\begin{aligned}(y_{k+1}-x^*)'(y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1}) &= (y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1})'(y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1})\\ &\quad + (\Pi_{Y_{k+1,i}}y_{k+1}-x^*)'(y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1})\\ &\geq (y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1})'(y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1})+0\\ &= \|y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1}\|^2,\end{aligned}$$

where the inequality uses $(\Pi_{Y_{k+1,i}}y_{k+1}-x^*)'(y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1}) \geq 0$, which is a property of the projection $\Pi_{Y_{k+1,i}}$ onto the convex set $Y_{k+1,i}$. It follows that

$$\begin{aligned}\|x_{k+1}-x^*\|^2 &\leq \|y_{k+1}-x^*\|^2 + \left\|\sum_{i=1}^M \xi_i \Pi_{Y_{k+1,i}}y_{k+1}-y_{k+1}\right\|^2 - 2\sum_{i=1}^M \xi_i \left\|y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1}\right\|^2\\ &\leq \|y_{k+1}-x^*\|^2 + \sum_{i=1}^M \xi_i \left\|y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1}\right\|^2 - 2\sum_{i=1}^M \xi_i \left\|y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1}\right\|^2\\ &= \|y_{k+1}-x^*\|^2 - \sum_{i=1}^M \xi_i \left\|y_{k+1}-\Pi_{Y_{k+1,i}}y_{k+1}\right\|^2\\ &= \|y_{k+1}-x^*\|^2 - e_{k+1},\end{aligned}$$

where the second inequality uses the Jensen inequality that

$$\left\|\sum_{i=1}^M \xi_i \Pi_{Y_{k+1,i}}y_{k+1}-y_{k+1}\right\|^2 \leq \sum_{i=1}^M \xi_i \|(\Pi_{Y_{k+1,i}}y_{k+1}-y_{k+1})\|^2,$$

and the last equality uses Eq. (10). By using the definition of y_{k+1} (which is identical for all three algorithms), we have

$$\|y_{k+1}-x^*\|^2 = \|x_k-x^*-\alpha_k g(x_k,v_{k+1})\|^2 = \|x_k-x^*\|^2-2\alpha_k g(x_k,v_{k+1})'(x_k-x^*)+\alpha_k^2\|g(x_k,v_{k+1})\|^2.$$

Combining the preceding relations, we obtain (8).

(b) We consider the squared distance between x_{k+1} and the feasible set $\mathcal{X}$. By using the nonexpansiveness of projection, property of distance function and the Jensen inequality, we obtain

$$\begin{aligned}\mathrm{d}^2(x_{k+1}) &= \mathrm{d}^2\left(\Pi_{\mathcal{X}_0}\left[\sum_{i=1}^M \xi_i \Pi_{Y_{k+1,i}}y_{k+1}\right]\right) \leq \mathrm{d}^2\left(\sum_{i=1}^M \xi_i \Pi_{Y_{k+1,i}}y_{k+1}\right)\\ &\leq \left\|\left(\sum_{i=1}^M \xi_i \Pi_{Y_{k+1,i}}y_{k+1}\right)-\Pi_{\mathcal{X}}y_{k+1}\right\|^2 = \left\|\sum_{i=1}^M \xi_i(\Pi_{Y_{k+1,i}}y_{k+1}-\Pi_{\mathcal{X}}y_{k+1})\right\|^2 \qquad (11)\\ &\leq \sum_{i=1}^M \xi_i \|\Pi_{Y_{k+1,i}}y_{k+1}-\Pi_{\mathcal{X}}y_{k+1}\|^2.\end{aligned}$$

For each i, by using the property of projection $\Pi_{Y_{k+1,i}}$, we have $(\Pi_{Y_{k+1,i}}y_{k+1}-\Pi_{\mathcal{X}}y_{k+1})'(\Pi_{Y_{k+1,i}}y_{k+1}-y_{k+1}) \leq 0$. As a result, we have

$$\|\Pi_{Y_{k+1,i}}y_{k+1}-\Pi_{\mathcal{X}}y_{k+1}\|^2 \leq \|y_{k+1}-\Pi_{\mathcal{X}}y_{k+1}\|^2 - \|\Pi_{Y_{k+1,i}}y_{k+1}-y_{k+1}\|^2.$$

Then by using the basic inequality $\|a+b\|^2 = \|a\|^2 + \|b\|^2 + 2a'b \leq (1+1/\epsilon)\|a\|^2 + (1+\epsilon)\|b\|^2$ for any $\epsilon > 0$, we obtain

$$\begin{aligned}\|y_{k+1} - \Pi_{\mathcal{X}} y_{k+1}\|^2 \leq & \|y_{k+1} - \Pi_{\mathcal{X}} x_k\|^2 \\ = & \|y_{k+1} - x_k + x_k - \Pi_{\mathcal{X}} x_k\|^2 \\ \leq & (1+1/\epsilon)\|y_{k+1} - x_k\|^2 + (1+\epsilon)\|x_k - \Pi_{\mathcal{X}} x_k\|^2 \\ = & (1+1/\epsilon)\alpha_k^2 \|g(x_k, v_{k+1})\|^2 + (1+\epsilon)\mathrm{d}^2(x_k).\end{aligned}$$

where ϵ is an arbitrary scalar. Combining the preceding two relations, we obtain

$$\|\Pi_{Y_{k+1,i}} y_{k+1} - \Pi_{\mathcal{X}} y_{k+1}\|^2 \leq (1+\epsilon)\mathrm{d}^2(x_k) + (1+1/\epsilon)\alpha_k^2 \|g(x_k, v_{k+1})\|^2 - \|\Pi_{Y_{k+1,i}} y_{k+1} - y_{k+1}\|^2.$$

Applying this to (11), we obtain

$$\mathrm{d}^2(x_{k+1}) \leq (1+\epsilon)\mathrm{d}^2(x_k) + (1+1/\epsilon)\alpha_k^2 \|g(x_k, v_{k+1})\|^2 - \sum_{i=1}^{M} \xi_i \|\Pi_{Y_{k+1,i}} y_{k+1} - y_{k+1}\|^2.$$

Finally, we apply (10) to the preceding inequality and obtain (9). □

Lemma 2 *Under the assumptions of Theorem 1, there exists a constant $C \in (0,1)$ such that*

$$\mathbf{E}\left[e_{k+1} \mid \mathscr{F}_k\right] \geq C \cdot \mathbf{E}\left[\mathrm{d}^2(y_{k+1}) \mid \mathscr{F}_k\right],$$

where $C = \frac{\eta}{m}$ in the case of Algorithm 1, and $C = \frac{M\eta}{m}$ in the case of Algorithms 2 or 3.

Proof. Note the definition of e_{k+1} given by Eq. (10). In the case of Algorithm 1, each $X_{\omega_{k+1,i}}$ has probability $1/m$ to be the max-distance set to y_{k+1} which achieves the maximum in the linear regularity condition of Assumption 2. As a result, we have

$$\mathbf{E}\left[e_{k+1} \mid \mathscr{F}_k\right] = \frac{1}{M}\sum_{i=1}^{M} \mathbf{E}\left[d^2(y_{k+1}, X_{\omega_{k+1,i}}) \mid \mathscr{F}_k\right] \geq \frac{\eta}{m}\mathbf{E}\left[\mathrm{d}^2(y_{k+1}) \mid \mathscr{F}_k\right].$$

In the case of Algorithm 2, the M sets are sampled according to a uniform distribution without replacement. As a result, the max-distance set within the samples has probability M/m to be the max-distance set to y_{k+1} which achieves the maximum in the linear regularity condition of Assumption 2. So we have

$$\mathbf{E}\left[e_{k+1} \mid \mathscr{F}_k\right] = \mathbf{E}\left[\max_{i=1,\ldots,M} d^2(y_{k+1}, X_{\omega_{k+1,i}}) \mid \mathscr{F}_k\right] \geq \frac{M\eta}{m}\mathbf{E}\left[\mathrm{d}^2(y_{k+1}) \mid \mathscr{F}_k\right].$$

In the case of Algorithm 3, we have $\mathcal{P}_k \subset X_{\omega_{k+1,i}}$ for all i, therefore

$$\mathrm{d}^2(y_{k+1}, \mathcal{P}_k) \geq \max_{i=1,\ldots,M} d^2(y_{k+1}, X_{\omega_{k+1,i}}).$$

Thus by using the result of the second case, we have

$$\begin{aligned}\mathbf{E}\left[e_{k+1} \mid \mathscr{F}_k\right] &= \mathbf{E}\left[\mathrm{d}^2(y_{k+1}, \mathcal{P}_k) \mid \mathscr{F}_k\right] \\ &\geq \mathbf{E}\left[\max_{i=1,\ldots,M} d^2(y_{k+1}, X_{\omega_{k+1,i}}) \mid \mathscr{F}_k\right] \\ &\geq \frac{M\eta}{m}\mathbf{E}\left[\mathrm{d}^2(y_{k+1}) \mid \mathscr{F}_k\right].\end{aligned}$$

Note that the above inequality is tight when y_{k+1} violates only one constraint. $\square$

We are ready to develop the main proofs of Theorems 1 and 2.

Proof of Theorem 1. (a) Applying Lemma 1 and taking conditional expectation on both sides of (8), we have

$$\begin{aligned}\mathbf{E}[\|x_{k+1}-x^*\|^2|\mathscr{F}_k] \leq & \|x_k-x^*\|^2 - 2\alpha_k \mathbf{E}[g(x_k,v_{k+1})|\mathscr{F}_k]'(x_k-x^*) \\ & - \mathbf{E}\left[e_{k+1}|\mathscr{F}_k\right] + \alpha_k^2 \mathbf{E}[\|g(x_k,v_{k+1})\|^2|\mathscr{F}_k].\end{aligned}$$

According to Assumption 1(b), the second term becomes

$$2\alpha_k \mathbf{E}[g(x_k,v_{k+1})|\mathscr{F}_k]'(x_k-x^*) = 2\alpha_k \tilde{\nabla} F(x_k)'(x_k-x^*) \geq 2\alpha_k(F(x_k)-F(x^*)),$$

where the inequality uses the convexity of F and the property of subgradients. According to Assumption 1(c) and using $x_k \in \mathcal{X}_0$, the fourth term can be bounded by

$$\mathbf{E}[\|g(x_k,v_{k+1})\|^2|\mathscr{F}_k] \leq B^2.$$

Applying Lemma 2 and using the following fact:

$$\begin{aligned}\|y-\Pi_S y\|^2 \leq \|y-\Pi_S x\|^2 &\leq \|y-x+x-\Pi_S x\|^2 \\ &\leq 2\|x-y\|^2 + 2\|x-\Pi_S x\|^2, \qquad \forall x,y \in \Re^n, S \subset \Re^n \text{ convex},\end{aligned}$$

we have

$$\begin{aligned}\mathbf{E}\left[e_{k+1}|\mathscr{F}_k\right] &\geq C\mathbf{E}\left[\mathrm{d}^2(y_{k+1})|\mathscr{F}_k\right] \\ &\geq C\mathbf{E}\left[\frac{1}{2}\mathrm{d}^2(x_k) - \|y_{k+1}-x_k\|^2 \;\middle|\; \mathscr{F}_k\right] \\ &\geq \frac{C}{2}\mathrm{d}^2(x_k) - \alpha_k^2 C B^2 \\ &\geq \frac{C}{2}\mathrm{d}^2(x_k) - \alpha_k^2 B^2,\end{aligned} \tag{12}$$

where we have used $C \leq 1$ in the last step. (Since $\eta < 1$ and $M < m$, it is easy to verify all three possible values of C are bounded by 1.) Combining the preceding inequalities, we obtain for all $k \geq 0$, with probability 1 that

$$\mathbf{E}[\|x_{k+1}-x^*\|^2|\mathscr{F}_k] \leq \|x_k-x^*\|^2 + 2\alpha_k^2 B^2 - 2\alpha_k(F(x_k)-F(x^*)) - \frac{C}{2}\mathrm{d}^2(x_k). \tag{13}$$

By using the convexity of $F(x)$ and the property of subgradients, we have

$$\begin{aligned}F(x_k)-F(x^*) =& F(\Pi_{\mathcal{X}}x_k) - F(x^*) + F(x_k) - F(\Pi_{\mathcal{X}}x_k) \\ \geq& F(\Pi_{\mathcal{X}}x_k) - F(x^*) + \widetilde{\nabla} F(\Pi_{\mathcal{X}}x_k)'(x_k - \Pi_{\mathcal{X}}x_k) \\ \geq& F(\Pi_{\mathcal{X}}x_k) - F(x^*) - \|\widetilde{\nabla} F(\Pi_{\mathcal{X}}x_k)\|\|x_k - \Pi_{\mathcal{X}}x_k\| \\ \geq& F(\Pi_{\mathcal{X}}x_k) - F(x^*) - B\mathrm{d}(x_k),\end{aligned} \tag{14}$$

where the last step is based on

$$\|\widetilde{\nabla} F(x)\| = \|\mathbf{E}[g(x, v_{k+1})|\mathscr{F}_k]\| \leq \sqrt{\mathbf{E}[\|g(x, v_{k+1})\|^2|\mathscr{F}_k]} \leq B, \qquad \forall\, x \in \Re^n.$$

Then we obtain

$$\begin{aligned}\mathbf{E}[\|x_{k+1} - x^*\|^2|\mathscr{F}_k] \leq & \|x_k - x^*\|^2 + 2\alpha_k^2 B^2 - 2\alpha_k\big(F(\Pi_{\mathcal{X}} x_k) - F(x^*)\big) \\ & + 2\alpha_k B \mathrm{d}(x_k) - \frac{C}{2}\mathrm{d}^2(x_k) \\ \leq & \|x_k - x^*\|^2 + \left(2 + \frac{2}{C}\right) B^2 \alpha_k^2 - 2\alpha_k\big(F(\Pi_{\mathcal{X}} x_k) - F(x^*)\big),\end{aligned}$$

where the second inequality uses the basic quadratic inequality

$$2\alpha_k B \mathrm{d}(x_k) - \frac{C}{2}\mathrm{d}^2(x_k) \leq \frac{2}{C} B^2 \alpha_k^2.$$

(b) We apply a similar analysis to Lemma 1 Eq. (9). Taking conditional expectation on both sides of (9) and applying (12), we obtain

$$\mathbf{E}[\mathrm{d}^2(x_{k+1})|\mathscr{F}_k] \leq (1+\epsilon)\mathrm{d}^2(x_k) + (2 + 1/\epsilon)\alpha_k^2 B^2 - \frac{C}{2}\mathrm{d}^2(x_k).$$

where ϵ is an arbitrary positive scalar. Letting $\epsilon = C/4$, we have

$$\mathbf{E}[\mathrm{d}^2(x_{k+1})|\mathscr{F}_k] \leq \Big(1 - \frac{C}{4}\Big)\mathrm{d}^2(x_k) + \Big(2 + \frac{4}{C}\Big)\alpha_k^2 B^2.$$

□

Proof of Theorem 2. The analysis follows from Theorem 1. Now we have obtained Eq. (6) and Eq. (7). By applying the *Coupled Supermartingale Convergence Theorem* ([WB16] Theorem 1), we obtain that $\{x_k\}$ converges with probability 1 to a random point in the set of optimal solutions of problem (1). Since $\widetilde{x}_k = \frac{1}{k+1}\sum_{t=0}^{k} x_t$, we obtain that $\{\widetilde{x}_k\}$ has the same pathwise convergence property as that of $\{x_k\}$. □

4 Convergence Rate Analysis

In this section, we analyze the rate of convergence of the random multi-constraint projection algorithms. We provide convergence rate results for both the feasibility error and the optimality error. In particular, we study the case of convex objectives and the case of strongly convex objectives separately.

4.1 Convergence Rate of Feasibility Error

Let us study the feasibility error associated with the iterates generated by the algorithms. We consider the expected squared distance from the iterate to the feasible set $\mathcal{X}$, i.e.,

$$\mathbf{E}[\mathrm{d}^2(x_k)] = \mathbf{E}[\|x_k - \Pi_{\mathcal{X}} x_k\|^2].$$

We have shown that this feasibility error decreases to zero according to a geometric contraction with an additive error. The additive error is due to the gradient descent step and dominates the convergence of feasibility error. We give a finite-time feasibility error bound in the following theorem.

Theorem 3 (Feasibility Error Bound) *Let Assumptions 1 and 2 hold, let the sequence $\{x_k\}$ be generated by any of Algorithms 1, 2, or 3. If there exists $\bar{k} \geq 0$ such that $\alpha_{k+1}^2 \geq \big(1 - \frac{C}{8}\big)\alpha_k^2$ for all $k \geq \bar{k}$. Then the feasibility error satisfies for all $k \geq \bar{k}$ that*

$$\begin{aligned} \mathbf{E}[\mathrm{d}^2(x_k)] &\leq 4B^2\Big(\frac{2}{C} + \frac{4}{C^2}\Big)\left(2\alpha_k^2 + \frac{C}{4}\left(\sum_{t=0}^{\bar{k}} \alpha_t^2\right)\Big(1 - \frac{C}{4}\Big)^{k-\bar{k}}\right) + \mathrm{d}^2(x_0)\Big(1 - \frac{C}{4}\Big)^k \\ &= 8B^2\Big(\frac{2}{C} + \frac{4}{C^2}\Big)\mathcal{O}(\alpha_k^2), \end{aligned} \tag{15}$$

where $C = \frac{\eta}{m}$ in the case of Algorithm 1, and $C = \frac{M\eta}{m}$ in the case of Algorithms 2 or 3.

Lemma 3 *Let $\{\delta_k\}$ and $\{\alpha_k\}$ be two sequences of nonnegative scalars such that*

$$\delta_{k+1} \leq (1 - \beta)\delta_k + N\alpha_k^2, \qquad \forall\ k \geq 0,$$

where $\beta \in (0, 1)$ and $N \geq 0$ are constants. If there exists $\bar{k} \geq 0$ such that $\alpha_{k+1}^2 \geq \big(1 - \frac{\beta}{2}\big)\alpha_k^2$ for all $k \geq \bar{k}$, we have

$$\delta_k \leq \frac{2N}{\beta}\alpha_k^2 + \delta_0(1 - \beta)^k + \left(N\sum_{t=0}^{\bar{k}} \alpha_t^2\right)(1 - \beta)^{k-\bar{k}}.$$

Proof. For $k \geq \bar{k}$, we have

$$\begin{aligned} \delta_{k+1} \leq& (1 - \beta)\delta_k + N\alpha_k^2 \\ =& (1 - \beta)\delta_k + \frac{2}{\beta}\Big(1 - \frac{\beta}{2}\Big)N\alpha_k^2 - \frac{2}{\beta}(1 - \beta)N\alpha_k^2 \\ \leq& (1 - \beta)\delta_k + \frac{2}{\beta}N\alpha_{k+1}^2 - \frac{2}{\beta}(1 - \beta)N\alpha_k^2. \end{aligned}$$

As a result, we have

$$\delta_{k+1} - \frac{2N}{\beta}\alpha_{k+1}^2 \leq (1 - \beta)\Big(\delta_k - \frac{2N}{\beta}\alpha_k^2\Big).$$

Applying the preceding relation inductively, we obtain for all $k \geq \bar{k}$ that

$$\delta_k \leq \frac{2N}{\beta}\alpha_k^2 + \Big(\delta_{\bar{k}} - \frac{2N}{\beta}\alpha_{\bar{k}}^2\Big)(1 - \beta)^{k-\bar{k}}.$$

Moreover, we have

$$\delta_{\bar{k}} \leq \delta_0(1 - \beta)^{\bar{k}} + N\sum_{t=0}^{\bar{k}} \alpha_t^2$$

Combining the preceding two inequalities, we complete the proof. □

Proof of Theorem 3. We follow the proof of Theorem 1. We take expectation over (7) and obtain

$$\mathbf{E}[\mathrm{d}^2(x_{k+1})] \leq \Big(1 - \frac{C}{4}\Big)\mathbf{E}[\mathrm{d}^2(x_k)] + \Big(2 + \frac{4}{C}\Big)B^2\alpha_k^2.$$

We apply Lemma 3 to the preceding inequality and obtain (15). □

Remark. When the stepsize α_k takes the form $\alpha_k = \alpha_0 k^{-a}$ where $a \in (0, 1]$, the stepsize assumption $\alpha_{k+1}^2 \geq (1 - \frac{C}{8})\alpha_k^2$ is satisfied for all k sufficiently large. Then we may use Theorem 3 and obtain

$$\mathbf{E}[\mathrm{d}^2(x_k)] \leq \mathcal{O}\left(\frac{B^2(1+C)\alpha_0^2}{C^2} \cdot k^{-2a}\right) + \mathrm{d}^2(x_0)\Big(1 - \frac{C}{4}\Big)^k.$$

As $k \to \infty$, the feasibility error is dominated by the error induced by the optimality update step, so that the distance to the feasible region satisfies

$$\mathrm{d}(\widetilde{x}_k) = \mathcal{O}(k^{-a}),$$

for k sufficiently large, with high probability.

According to Theorem 3, we learn that the constant $C \in (0, 1]$ plays a key role in the convergence rate of the feasibility error. A larger value of C leads to faster convergence to the feasible set $\mathcal{X}$. As suggested above, the constant C is a useful metric that quantifies the efficiency of the feasibility update steps. It is jointly determined by the feasibility update scheme, the spatial layout of the constraints, as well as the sampling oracle.

4.2 Convergence Rate of Optimality Error

Now let us analyze the optimality errors associated with $\{x_k\}$ and $\{\widetilde{x}_k\}$. The traditional error metric is the objective error $F(\widetilde{x}_k) - F(x^*)$ or the expected objective error $\mathbf{E}[F(\widetilde{x}_k) - F(x^*)]$. However, due to the constraint randomization of our algorithms, the iterates are not guaranteed to be feasible. When $\widetilde{x}_k$ is infeasible, the objective error $F(\widetilde{x}_k) - F(x^*)$ could take negative value, so it is not a valid error metric. In order to quantify the optimality error separately from the feasibility error, we will focus on the projected averaged iterates $\{\Pi_{\mathcal{X}}\widetilde{x}_k\}$.

For simplicity of analysis, we focus on stepsizes taking the form $\alpha_k = \alpha_0 \cdot k^{-a}$. In the next theorem, we consider the general convex objectives and provide estimates of the optimality error after k iterations.

Theorem 4 (Optimality Error for Convex Objectives) *Let Assumptions 1 and 2 hold, let $\{\widetilde{x}_k\}$ be the sequence of averaged iterates generated by any of Algorithms 1, 2, or 3, and let the stepsize be $\alpha_k = \alpha_0 k^{-a}$ for some $\alpha_0 > 0$ and $a \in (0, 1]$. Then for all $k \geq 0$,*

$$\mathbf{E}[F(\Pi_{\mathcal{X}}\widetilde{x}_k) - F(x^*)] \leq \begin{cases} \dfrac{D^2}{2\alpha_0}k^{a-1} + \left(1 + \dfrac{1}{C}\right)B^2\dfrac{\alpha_0}{1-a}k^{-a}, & a \in (0, 1), \\ \dfrac{D^2}{2\alpha_0}k^{a-1} + \left(1 + \dfrac{1}{C}\right)B^2\alpha_0\left(\dfrac{\log k + 1}{k}\right), & a = 1. \end{cases} \tag{16}$$

Proof of Theorem 4. We follow the proof of Theorem 1 and take expectation on both sides of (6). We obtain

$$\mathbf{E}[\|x_{k+1}-x^*\|^2] \leq \mathbf{E}[\|x_k-x^*\|^2] + 2B^2\alpha_k^2 - 2\alpha_k\mathbf{E}\left[F(x_k)-F(x^*)\right] - \frac{C}{2}\mathbf{E}\left[\mathrm{d}^2(x_k)\right], \tag{17}$$

Denote for simplicity that $E_k = \mathbf{E}[\|x_k-x^*\|^2]$ and $A = 2B^2$. We rearrange the preceding inequality and obtain

$$2\mathbf{E}[F(x_k)-F(x^*)] \leq \frac{1}{\alpha_k}\Big(E_k - E_{k+1}\Big) + A\alpha_k - \frac{C}{2\alpha_k}\mathbf{E}\left[\mathrm{d}^2(x_k)\right].$$

Summing the preceding inequalities from 0 to k, we obtain

$$\begin{aligned}
2\sum_{t=0}^{k}\mathbf{E}[F(x_t)-F(x^*)] &\leq \sum_{t=0}^{k}\left(\frac{1}{\alpha_t}\Big(E_t-E_{t+1}\Big) + A\alpha_t - \frac{C}{2\alpha_t}\mathbf{E}\left[\mathrm{d}^2(x_t)\right]\right)\\
&= \sum_{t=1}^{k}\left(\frac{1}{\alpha_t}-\frac{1}{\alpha_{t-1}}\right)E_t + \frac{1}{\alpha_0}E_0 - \frac{1}{\alpha_{k+1}}E_{k+1} + A\sum_{t=0}^{k}\alpha_t - \sum_{t=0}^{k}\frac{C}{2\alpha_t}\mathbf{E}\left[\mathrm{d}^2(x_t)\right]\\
&\leq \sum_{t=1}^{k}(\frac{1}{\alpha_t}-\frac{1}{\alpha_{t-1}})D^2 + \frac{1}{\alpha_0}D^2 + A\sum_{t=0}^{k}\alpha_t - \sum_{t=0}^{k}\frac{C}{2\alpha_t}\mathbf{E}\left[\mathrm{d}^2(x_t)\right]\\
&= \frac{D^2}{\alpha_k} + A\sum_{t=0}^{k}\alpha_t - \sum_{t=0}^{k}\frac{C}{2\alpha_t}\mathbf{E}\left[\mathrm{d}^2(x_t)\right],
\end{aligned}$$

where the second inequality uses the fact $\mathbf{E}[\|x_k-x^*\|^2] \leq D^2$ for all k (from Assumption 1(d) and $x_k \in \mathcal{X}_0$). By using the same analysis as in Eq. (14), we obtain

$$\begin{aligned}
\mathbf{E}[F(\Pi_{\mathcal{X}}\widetilde{x}_k)-F(x^*)] &\leq \mathbf{E}\left[F(\widetilde{x}_k)-F(x^*)\right] + B\mathbf{E}\left[\mathrm{d}(\widetilde{x}_k)\right]\\
&\leq \frac{1}{k+1}\sum_{t=0}^{k}\mathbf{E}[F(x_t)-F(x^*)] + B\mathbf{E}\left[\mathrm{d}(\widetilde{x}_k)\right]\\
&\leq \frac{D^2}{2(k+1)\alpha_k} + \frac{A}{2(k+1)}\sum_{t=0}^{k}\alpha_t + \left(B\mathbf{E}\left[\mathrm{d}(\widetilde{x}_k)\right] - \frac{C}{4(k+1)}\sum_{t=0}^{k}\frac{1}{\alpha_t}\mathbf{E}\left[\mathrm{d}^2(x_t)\right]\right),
\end{aligned}$$

where the second inequality is obtained by applying the convexity of F to the relation $\widetilde{x}_k = \frac{1}{k+1}\sum_{t=0}^{k}x_t$. By using the Cauchy-Schwarz inequality, we have

$$\left(\sum_{t=0}^{k}\frac{1}{\alpha_t}\mathrm{d}^2(x_t)\right)\times\left(\sum_{t=0}^{k}\alpha_t\right) \geq \left(\sum_{t=0}^{k}\sqrt{\frac{1}{\alpha_t}\mathrm{d}^2(x_t)\times\alpha_t}\right)^2 = \left(\sum_{t=0}^{k}\mathrm{d}(x_t)\right)^2,$$

By using the convexity of $\mathrm{d}^2(\cdot)$ [BV04], we further have

$$\sum_{t=0}^{k}\frac{1}{\alpha_t}\mathrm{d}^2(x_t) \geq \frac{(\sum_{t=0}^{k}\mathrm{d}(x_t))^2}{\sum_{t=0}^{k}\alpha_t} \geq \frac{(k+1)^2\mathrm{d}(\widetilde{x}_k)^2}{\sum_{t=0}^{k}\alpha_t}.$$

Then by using the preceding relation and a basic quadratic inequality, we have

$$B\mathbf{E}\left[\mathrm{d}(\widetilde{x}_k)\right] - \frac{C}{4(k+1)}\sum_{t=0}^{k}\frac{1}{\alpha_t}\mathbf{E}\left[\mathrm{d}^2(x_t)\right] \leq B\mathbf{E}\left[\mathrm{d}(\widetilde{x}_k)\right] - \frac{C(k+1)}{4}\frac{\mathbf{E}\left[\mathrm{d}(\widetilde{x}_k)^2\right]}{\sum_{t=0}^{k}\alpha_t} \leq \frac{B^2}{C(k+1)}\sum_{t=0}^{k}\alpha_t.$$

By combining the preceding relations, we obtain

$$\mathbf{E}[F(\Pi_{\mathcal{X}}\widetilde{x}_k) - F(x^*)] \leq \frac{D^2}{2(k+1)\alpha_k} + (\frac{A}{2} + \frac{B^2}{C})\frac{1}{k+1}\sum_{t=0}^{k}\alpha_t.$$

Letting $\alpha_k = \alpha_0 k^{-a}$, we have

$$\sum_{t=0}^{k} t^{-a} \leq 1 + \int_{x=1}^{k} x^{-a}\mathrm{d}x \leq \begin{cases} \frac{1}{1-a}k^{1-a}, & a \in (0,1), \\ 1 + \log k, & a = 1. \end{cases}$$

Then we have

$$\mathbf{E}[F(\Pi_{\mathcal{X}}\widetilde{x}_t) - F(x^*)] \leq \begin{cases} \frac{D^2}{2\alpha_0}k^{a-1} + \frac{\alpha_0}{2(1-a)}(\frac{A}{2} + \frac{B^2}{C})\big(k^{-a}\big), & a \in (0,1), \\ \frac{D^2}{2\alpha_0}k^{a-1} + \frac{\alpha_0}{2}(\frac{A}{2} + \frac{B^2}{C})\Big(\frac{\log k + 1}{k}\Big), & \alpha = 1. \end{cases}$$

□

Remark. In order to minimize the error bound, the best stepsize choice is

$$\alpha_k = \gamma \cdot \sqrt{C} \cdot \frac{D}{B} \cdot \frac{1}{\sqrt{k}},$$

where γ is some tuning parameter. The corresponding optimality error bound becomes

$$\mathbf{E}[F(\widetilde{x}_k) - F(x^*)] \leq \mathcal{O}\left(\frac{BD/\sqrt{C}}{\sqrt{k}}\right).$$

Note that when α_k are chosen in this way, the stepsize condition $\sum_{k=0}^{\infty}\alpha_k < \infty$ required by Theorem 2 does not hold. In other words, when the stepsize are optimized for the expected optimization error, the iterates happen to *not* converge almost surely.

Next we consider the case of strongly convex objectives. We say a function f is σ-strongly convex if there exists a constant $\sigma > 0$ such that

$$f(x) \geq f(y) + \widetilde{\nabla} f(y)'(x-y) + \frac{\sigma}{2}\|x-y\|^2, \qquad \forall\, x, y \in \Re^n,$$

where $\widetilde{\nabla} f(y)$ is an arbitrary subgradient of f at y. As a result, we also have

$$(x-y)'(\widetilde{\nabla} f(x) - \widetilde{\nabla} f(y)) \geq \sigma\|x-y\|^2, \qquad \forall\, x, y \in \Re^n.$$

When the objective function F is strongly convex, there exists a unique solution x^* to the constrained optimization problem. As a result, we may use the expected squared distance to x^* as a metric of optimality error. In the next theorem, we analyze the convergence rate in terms of the expected squared solution error $\mathbf{E}[\|\widetilde{x}_k - x^*\|^2]$.

Theorem 5 (Optimality Error for Strongly Convex Objectives) *Suppose that F is σ-strongly convex. Let Assumptions 1 and 2 hold, let the sequence $\{x_k\}$ be generated by any of Algorithms 1, 2, or 3. Let the stepsize be*

$$\alpha_k = \frac{1}{2\sigma(k+1)},$$

then

$$\mathbf{E}[\|x_k - x^*\|^2] \leq \frac{(2+\frac{2}{C})B^2}{4\sigma^2} \cdot \frac{1+\log(k)}{k}. \tag{18}$$

The proof of Theorem 5 uses the following lemma. We recall it for clarity.

Lemma 4 (Lemma 2.1 of [NB01]) *Let $\{\delta_k\}$ be a sequence of nonnegative scalars such that for all $k \geq 0$*

$$\delta_{k+1} \leq (1 - \frac{p}{k+1})\delta_k + \frac{d}{(k+1)^2},$$

for some $p > 0$ and $d > 0$. Then

$$\delta_k \leq \begin{cases} \frac{1}{(k+1)^p}\left(\delta_0 + \frac{2^p d(2-p)}{1-p}\right) & \text{if } p < 1, \\ \frac{d(\log(k)+1)}{k} & \text{if } p = 1, \\ \frac{1}{(p-1)(k+1)}\left(d + \frac{(p-1)\delta_0 - d}{(k+1)^{p-1}}\right) & \text{if } p > 1. \end{cases}$$

Proof of Theorem 5. We recall Eq. (8) which follows from Lemma 1:

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k g(x_k, v_{k+1})'(x_k - x^*) - e_{k+1} + \alpha_k^2 \|g(x_k, v_{k+1})\|^2.$$

Let $h_k = g(x_k, v_{k+1}) - \widetilde{\nabla} F(x_k)$. We analyze the second term and obtain

$$\begin{aligned} g(x_k, v_{k+1})'(x_k - x^*) =& \widetilde{\nabla} F(x_k)'(x_k - x^*) + (g(x_k, v_{k+1}) - \widetilde{\nabla} F(x_k))'(x_k - x^*) \\ =& \widetilde{\nabla} F(x_k)'(x_k - x^*) + h_k'(x_k - x^*) \\ =& (\widetilde{\nabla} F(x_k) - \widetilde{\nabla} F(x^*))'(x_k - x^*) + \widetilde{\nabla} F(x^*)'(x_k - x^*) + h_k'(x_k - x^*), \end{aligned}$$

where the first term can be bounded using the strong convexity assumption as follows

$$(\widetilde{\nabla} F(x_k) - \widetilde{\nabla} F(x^*))'(x_k - x^*) \geq \sigma \|x_k - x^*\|^2,$$

and the second term can be bounded using the optimality of x^* and Assumption 1(c) as follows

$$\begin{aligned} \widetilde{\nabla} F(x^*)'(x_k - x^*) \geq& \widetilde{\nabla} F(x^*)'(x_k - \Pi_{\mathcal{X}} x_k) \\ \geq& -\|\widetilde{\nabla} F(x^*)\| \|x_k - \Pi_{\mathcal{X}} x_k\| \\ \geq& -\mathbf{E}[\|g(x^*, v_{k+1})\| | \mathscr{F}_k] \mathrm{d}(x_k) \\ \geq& -B\mathrm{d}(x_k), \end{aligned}$$

where the first inequality uses the fact $\widetilde{\nabla} F(x^*)'(x - x^*) \geq 0$ for all $x \in \mathcal{X}$. We combine the preceding relations and obtain

$$\begin{aligned} \|x_{k+1} - x^*\|^2 \leq& (1 - 2\alpha_k\sigma)\|x_k - x^*\|^2 - 2\alpha_k h_k'(x_k - x^*) + 2B\alpha_k \mathrm{d}(x_k) \\ & - e_{k+1} + \alpha_k^2 \|g(x_k, v_{k+1})\|^2. \end{aligned} \tag{19}$$

Taking conditional expectation on both sides of (19), and using $\mathbf{E}[h_k|\mathscr{F}_k] = 0$ and (12), we obtain

$$\mathbf{E}[\|x_{k+1} - x^*\|^2|\mathscr{F}_k] \leq (1 - 2\alpha_k\sigma)\|x_k - x^*\|^2 + 2B\alpha_k \mathrm{d}(x_k) - \frac{C}{2}\mathrm{d}^2(x_k) + 2B^2\alpha_k^2.$$

Taking expectation on both sides and using the fact

$$2B\alpha_k \mathrm{d}(x_k) - \frac{C}{2}\mathrm{d}^2(x_k) \leq \frac{2B^2}{C}\alpha_k^2,$$

we have

$$\mathbf{E}[\|x_{k+1} - x^*\|^2] \leq (1 - 2\alpha_k\sigma)\mathbf{E}[\|x_k - x^*\|^2] + \Big(2B^2 + \frac{2B^2}{C}\Big)\alpha_k^2. \tag{20}$$

By applying Lemma 4 to the preceding inequality, we obtain the error bound. □

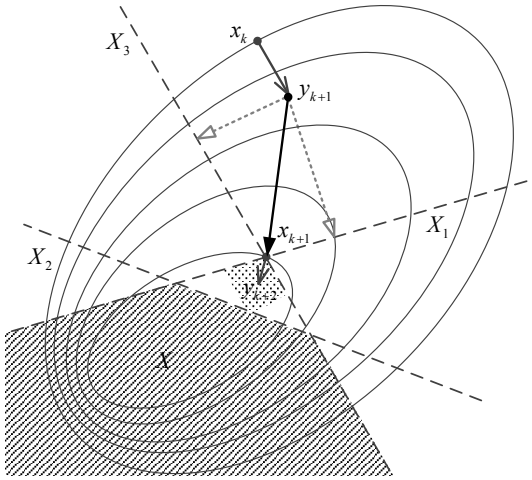


Figure 2: Projecting onto the polyhedral set $\mathcal{P}_k = X_1 \cap X_3$ reduces the feasibility error much more than projecting onto the max-distance set X_3. The improvement becomes more significant when the crossing angle between X_1 and X_3 becomes smaller.

4.3 Summary and Discussions

In what follows, we summarize the convergence rate results of Theorems 3,4,5. In the case of convex objectives, the stepsize chosen to minimize the optimality error bound is given by

$$\alpha_k \propto \sqrt{C} \cdot \frac{D}{B} \cdot \frac{1}{\sqrt{k}}.$$

Then by using Theorems 3 and 4, we obtain the following optimality and feasibility error bounds

$$\mathbf{E}[F(\widetilde{x}_k) - F(x^*)] = \mathcal{O}\left(\frac{BD/\sqrt{C}}{\sqrt{k}}\right), \qquad \mathbf{E}[\mathrm{d}^2(\tilde{x}_k)] = \mathcal{O}\left(\frac{D^2}{C} \cdot \frac{\log k}{k}\right).$$

In the case of strongly convex objectives, the stepsize is chosen to be

$$\alpha_k = \frac{1}{2\sigma(k+1)},$$

Then by using Theorems 3 and 5, we obtain the following optimality and feasibility error bounds

$$\mathbf{E}[\|x_k - x^*\|^2] \leq \mathcal{O}\left(\frac{B^2/C}{\sigma^2} \cdot \frac{1+\log(k)}{k}\right), \qquad \mathbf{E}[\mathrm{d}^2(x_k)] = \mathcal{O}\left(\frac{B^2}{C^2\sigma^2} \cdot \frac{1}{k}\right).$$

We report the preceding error bounds in Table 1 (see Section 1).

Now let us compare Algorithms 1, 2, and 3. The convergence rates of the three algorithm differ only in the parameter C. Recall that $C = \frac{\eta}{m}$ in the case of Algorithm 1, and $C = \frac{M\eta}{m}$ in the case of Algorithms 2 or 3. It implies that Algorithms 2 and 3 exhibit faster convergence than Algorithm 1.

Let us compare polyhedral-set projection (Algorithm 3) with the max-set projection scheme (Algorithm 2). According to our theoretical analysis, they have the same error bounds. However, the convergence rate estimates for Algorithm 3 are quite conservative. More specifically, the polyhedral-set projection scheme always performs better or equally well with the max-set scheme. This is because the polyhedral set is a subset of the max-distance set, thus it is a better approximation to $\mathcal{X}$. We illustrate this point using a simple geometric example, which is given by Figure 2. Numerical results also validate that Algorithm 3 converges faster than the two other algorithms.

5 Numerical Results

In this section, we conduct two numerical experiments to justify our algorithms. In the first experiment, we test the algorithms on problems with ball-like constraints. The results show that the max-distance-set scheme (Algorithm 2) and the polyhedral-set scheme (Algorithm 3) largely outperform the averaging scheme (Algorithm 1) and the baseline algorithm (2). They also show that the polyhedral-set scheme performs significantly better than all other schemes when the constraints have an "irregular" spatial distribution. In the second experiment, we apply the algorithms to a support vector machine problem. The results match very well with the error bounds predicted by theorems in Section 4. Throughout the experiments, the polyhedral-set scheme (Algorithm 3) demonstrates the best convergence properties in terms of optimization error, feasibility error, iteration efficiency, as well as sample efficiency.

5.1 Experiments on Ball-Like Constraints

We consider the following online regression problem

$$\begin{array}{ll} \min & \mathbf{E}\left[\|Y - X^T\beta\|^2\right] \\ \text{s.t.} & \beta \in \mathcal{X}_0 \cap (\cap_{i=1}^m \mathcal{X}_i), \end{array}$$

where $X \in \mathbb{R}^d$ is a Gaussian random variable with distribution $N(0, I_d)$ and $Y = X^T\beta^* + \eta$ with $\eta \sim N(0, 10)$ and some randomly generated β^*, $\mathcal{X}_0$ is a big Euclidean ball which contains the optimal solution in its interior. We consider two different sets of constraints $\cap_{i=1}^m \mathcal{X}_i$:

(i) A system of linear constraints given by cutting planes of a sphere; as illustrated in Figure 3. The feasible set $\mathcal{X}$ is approximately a ball centered at 0.

(ii) A system of linear constraints given by cutting planes of two spheres that nearly "touch" each other; as illustrated in Figure 4. The feasible set $\mathcal{X}$ is approximately the intersection of two balls with radius 61, centered at $(-60, 0)$ and $(60, 0)$ respectively.

The constraints in (i) are "regular" in the sense that constraint supersets (i.e., linear halfspaces) cross one another at large angles, corresponding to a large constant in the linear regularity condition (Assumption 2) and a large feasibility improvement constant C (Lemma 2). In contrast, the constraints in (ii) have an "irregular" distribution in the sense that some constraints cross one

another at very small angles, corresponding to a small constant in the linear regularity condition and a small value of C.

We apply the proposed stochastic algorithms to recover the optimal solution β_{OPT} from random samples of (X_i, Y_i) and sample constraint cutting planes. We test Algorithm 1, 2, 3 with $M = 5$ and the baseline algorithm (2) with $M = 1$ (M is the number of sample constraints used per iteration.) We let the stepsize be $1/(k + 10)$ and test each algorithm and parameter setting for 500 trial runs. Figures 3 and 4 plot the trajectories of the mean optimality error $\|\beta_t - \beta_{\mathrm{OPT}}\|_2^2$ and the mean feasibility error $\|\beta_t - \Pi_{\mathcal{X}}\beta_t\|_2^2$, for experiments (i) and (ii) respectively.

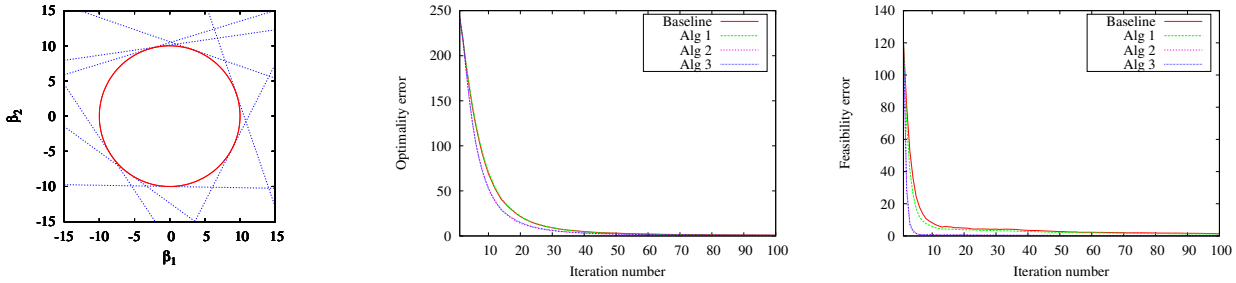


Figure 3: Convergence of random multi-constraint projection algorithms in experiment (i) ($d = 2, M = 5$ and $m = 300$.) Left: Illustration of the ball constraint. Middle: Mean optimality error. Right: Mean feasibility error.

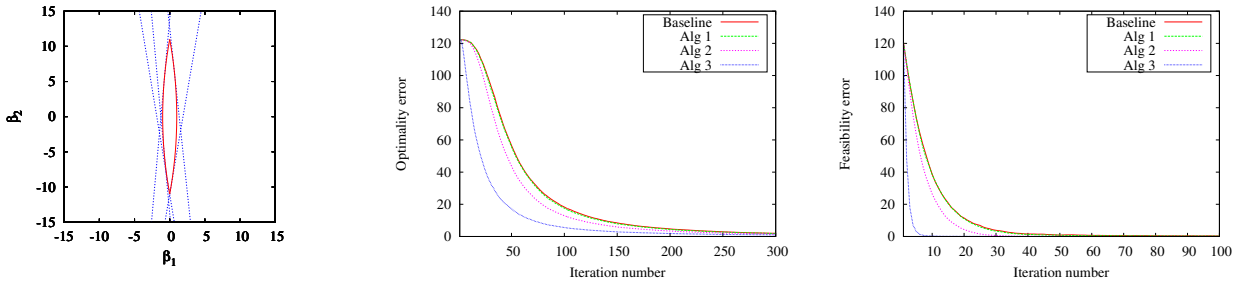


Figure 4: Convergence of random multi-constraint projection algorithms in experiment (ii) ($d = 2, M = 5$ and $m = 300$.) Left: The intersection of two balls. Middle: Mean optimality error. Right: Mean feasibility error.

In experiment (i), Algorithms 2 and 3 demonstrate similar performances in terms of both the optimality error and the feasibility error, outperforming Algorithm 1 and the baseline. This is consistent with our theory given in Section 4. The reason why Algorithms 2 and 3 perform similarly is due to the ball constraint. When using the polyhedral-set projection scheme, only one cutting plane out of many samples would be active after the projection, making it identical with the max-distance-set projection scheme. As a result, Algorithm 2 and Algorithm 3 exhibit similar performances on the ball constraint. This is an example where our worst-case error bounds are tight.

In experiment (ii), Algorithm 3 demonstrates a significantly better performance than the rest of

the algorithms. This is because the constraint sets are more irregular. Consider the situation where the sample constraints contain one cutting plane from the left sphere and one from the right sphere. According to the analysis of Section 4.3, the polyhedral-set algorithm enjoys a large improvement factor, as long as the two sample cutting planes intersect at a sharp angle. This is an example where Algorithm 3 substantially outperforms the worst-case error bounds.

5.2 Experiments on Support Vector Machine

The support vector machine problem is to find the best linear hyperplane that separates labeled training data $(x_i, y_i)_{i=1}^m$ into two classes. When the data are separable, it can be formulated as the minimization problem

$$\begin{aligned} \min_{\beta \in \Re^d} \quad & \frac{1}{2}\|\beta\|^2 \\ \text{s.t.} \quad & y_i \cdot x_i' \beta \geq 1, \\ & i = 1, \ldots, m. \end{aligned}$$

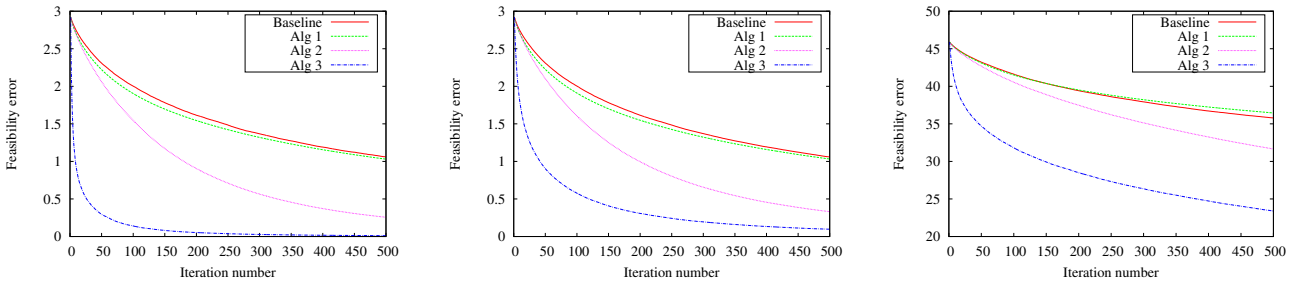
We introduce another ball constraint $\mathcal{X}_0 = \{\|\beta\| \leq R\}$ to ensure the boundedness of solution. The ball is very large, so it actually does not affect the algorithm at all.

In our experiment, we generate m pairs of data points $(x_1, y_1), \ldots, (x_m, y_m)$, with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. The data points $\{x_i\}_{i=1}^m$ are generated randomly from a mixture of Gaussian distribution. The labels $\{y_i\}_{i=1}^m$ are generated such that the data are separable and the optimization problem is feasible. In the support vector machine problem, each constraint naturally corresponds to a single data point. The proposed stochastic algorithms with random feasibility updates can be viewed as online training methods for the classification problem. The algorithms use the samples one by one without storing them.

We test Algorithms 1, 2, 3 and the baseline algorithm (2) on instances of the support vector machine problem with varying dimension d and data size (constraint size) m. We also test the algorithms with various values of M (the number of sample constraints used per iteration). We let $\beta_0 = 0$ and let the stepsize be $1/(k + 10)$ where k is the iteration number. For each algorithm and each parameter setting, we generate 500 trail runs. In Figure 5 and Figure 6, we plot the convergence of mean feasibility error and the percentage of constraints violation, respectively. Note that the mean optimality error is not reported here, as it is very similar to the mean feasibility error. In Figure 7, we illustrate the efficiency constants of various algorithms in terms of both the iteration efficiency and the sample efficiency.

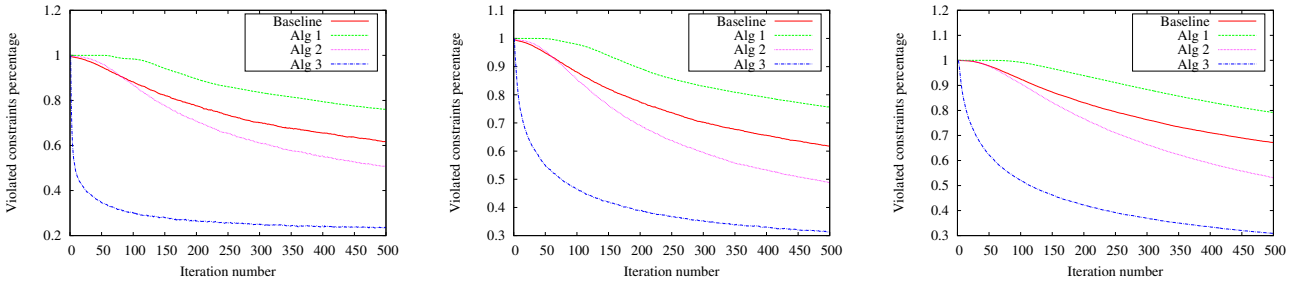
According to Figure 5, when M increases, the convergence rate of Algorithm 3 improves while that of Algorithm 1 and 2 remain largely unchanged. It hints that Algorithm 3 is more efficient in utilizing additional sample constraints. Figure 6 illustrates the algorithms' ability to identify the feasible region. From the figure, we can see that Algorithm 3 works far better than the other algorithms. Interestingly, we find that Algorithm 1 could be worse than the baseline algorithm.

Figure 7 highlights the efficiency factors of various algorithms. It shows that the inverse feasibility error is linear to the iteration number and the total sample budget. It verifies our theory



(a) $d = 100, m = 200, M = 30$ (b) $d = 100, m = 200, M = 10$ (c) $d = 100, m = 2000, M = 10$

Figure 5: Convergence of mean feasibility errors.



(a) $d = 100, m = 200, M = 30$ (b) $d = 100, m = 200, M = 10$ (c) $d = 100, m = 2000, M = 10$

Figure 6: Percentage of constraint violation as the algorithms proceed.

that the feasibility error $\|\beta_t - \Pi_{\mathcal{X}}\beta_t\|_2^2$ decreases on the order of $\mathcal{O}\left(\frac{1}{k^2}\right)$ (Theorem 3). The slopes in Figure 7 correspond to the constant hidden in the error bounds, therefore they are metrics of efficiency for the tested algorithms.

The right plot of Figure 7 is particularly interesting. It plots the inverse error against the total number of sample constraints used so far. It shows that Algorithm 3 works uniformly better than Algorithm 2, and that Algorithm 2 works uniformly better than Algorithm 1, when the total sample size is fixed (even if M varies). It indicates that Algorithm 3 is more efficient in utilizing the samples and therefore has a better sample complexity (in addition to better iteration complexity). Let us focus on Algorithm 3 with different values of M. We observe that the sample efficiency increases as M increases. It suggests that for a limited number of constraints, the algorithm is more efficient if we increase the samples used per iteration rather than increasing the number of iterations. This verifies our conjecture in Section 4.3.

6 Conclusion

We have proposed a class of random algorithms that involve stochastic (sub)gradients and random multi-constraint projections. We provide almost sure convergence and rate of convergence analysis for these algorithms. In particular, we have provided rate of convergence in terms of the optimality error and the feasibility error for a variety of feasibility update schemes, in the cases of general

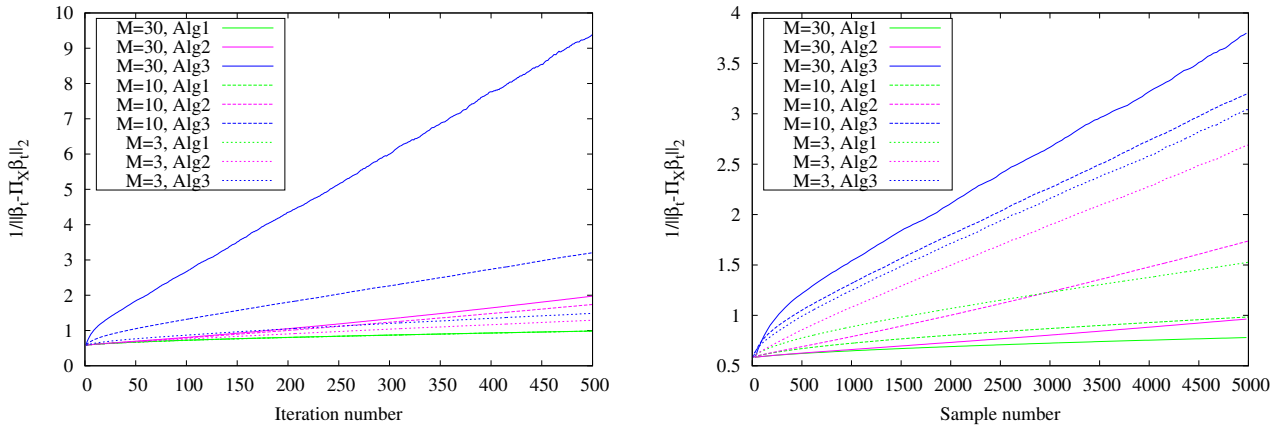


Figure 7: The inverse feasibility error is linearly related to the iteration number and the total constraint sample size. Left: The slope is a metric of iteration efficiency. Right: The slope is a metric of sample efficiency.

convex objectives and strongly convex objectives. See Table 1 for a summary of the convergence results. These results suggest that, by using random projection in replacement of expensive exact projection, stochastic gradient methods remain efficient.

Acknowledgements

We thank the anonymous reviewers for the constructive feedbacks and the improvement on the results.

References

[ABRW12] Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Trans. Inform. Theory*, 58(5):3235–3249, 2012.

[Bau96] Heinz H. Bauschke. *Projection Algorithms and Monotone Operators*. PhD thesis, Simon Fraser University, 1996.

[BBL97] Heinz H. Bauschke, Jonathan M. Borwein, and Adrian S. Lewis. The method of cyclic projections for closed convex sets in Hilbert space. *Contemp. Math.*, 204:1–38, 1997.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[CS08] Andrzej Cegielski and Agnieszka Suchocka. Relaxed alternating projection methods. *SIAM J. Optim.*, 19(3):1093–1106, 2008.

[DH06a] Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm. I. Angles between convex sets. *J. Approx. Theory*, 142(1):36–55, 2006.

[DH06b] Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm. II. Norms of nonlinear operators. *J. Approx. Theory*, 142(1):56–82, 2006.

[DH08] Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm III: Regularity of convex sets. *J. Approx. Theory*, 155(2):155–184, 2008.

[GL12] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM J. Optim.*, 22(4):1469–1492, 2012.

[GL13] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: Shrinking procedures and optimal algorithms. *SIAM J. Optim.*, 23(4):2061–2089, 2013.

[GPR67] LG Gubin, BT Polyak, and EV Raik. The method of projections for finding the common point of convex sets. *USSR Comput. Math. Math. Phys.*, 7(6):1–24, 1967.

[Hal62] Israel Halperin. The product of projection operators. *Acta Sci. Math.*, 23(1):96–99, 1962.

[LL10] Dennis Leventhal and Adrian S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Math. Oper. Res.*, 35(3):641–654, 2010.

[LM08] Adrian S. Lewis and Jérôme Malick. Alternating projections on manifolds. *Math. Oper. Res.*, 33(1):216–234, 2008.

[MB11] Eric Moulines and Francis R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems 24*, pages 451–459. 2011.

[NB01] Angelia Nedić and Dimitri Bertsekas. Convergence rate of incremental subgradient algorithms. In *Stochastic optimization: algorithms and applications*, volume 54 of *Appl. Optim.*, pages 223–264. 2001.

[Ned10] Angelia Nedić. Random projection algorithms for convex set intersection problems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 7655–7660, 2010.

[Ned11] Angelia Nedić. Random algorithms for convex minimization problems. *Math. Program.*, 129(2, Ser. B):225–253, 2011.

[Neu50] John von Neumann. Functional operators. *Ann. of Math.*, (22), 1950.

[NJLS08] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2008.

[NY83] Arkadi Nemirovskii and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

[RNV09] S. Sundhar Ram, Angelia Nedić, and Venugopal V. Veeravalli. Incremental stochastic subgradient algorithms for convex optimization. *SIAM J. Optim.*, 20(2):691–717, 2009.

[RS85] Herbert Robbins and David Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Herbert Robbins Selected Papers*, pages 111–135. 1985.

[RSS12] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Conference on Machine Learning*, ICML '12, pages 449–456, 2012.

[SZ13] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, ICML'13, pages I–71–I–79, 2013.

[Tse90] Paul Tseng. Successive projection under a quasi-cyclic order. Technical report, DTIC Document, 1990.

[WB15] Mengdi Wang and Dimitri P. Bertsekas. Incremental constraint projection methods for variational inequalities. *Math. Program.*, 150(2, Ser. A):321–363, 2015.

[WB16] Mengdi Wang and Dimitri P. Bertsekas. Stochastic first-order methods with random constraint projection. *SIAM J. Optim.*, 26(1):681–717, 2016.