A Projected Stochastic Gradient Method for Finite-Sum Problems with Linear Equality Constraints

Nataša Krklec Jerinkić[1], Benedetta Morini[2], Mahsa Yousefi[2]

[1] Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg D. Obradovića 4, Novi Sad, 21000, Serbia.
[2] Department of Industrial Engineering, University of Florence, Viale G.B. Morgagni 40, Florence, 50134, Italy.

Contributing authors: natasa.krklec@dmi.uns.ac.rs; benedetta.morini@unifi.it; mahsa.yousefi@unifi.it;

Abstract

A stochastic gradient method for finite-sum minimization subject to deterministic linear constraints is proposed and analyzed. The procedure presented adapts the projected gradient method on a convex set to the use of both a stochastic gradient and a possibly inexact projection map. Under standard assumptions in the field of stochastic gradient methods, we provide theoretical results in agreement with the theory for unconstrained problems. Numerical results are presented to show the practical behavior of the procedure.

Keywords: Constrained finite-sum minimization; stochastic gradient; exact and inexact projection.

MSC Classification: 90C30; 90C06; 90C53; 90C90; 65K05

1 Introduction

We consider the finite-sum optimization problem with linear equality constraints

$$\min_{x\in S} f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x), \quad S = \{x \in \mathbb{R}^n \mid Ax = b\}, \tag{1}$$

where the functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., N$, are continuously-differentiable, $A \in \mathbb{R}^{m \times n}$, $m < n$, is a full row-rank matrix and b is a vector in $\mathbb{R}^m$.

Problems of minimizing finite-sums are often encountered in applications, such as least-squares approximation and Machine Learning within training phase where parameters of a model function are optimized. This usually assumes a large number of data which corresponds to a large N in (1). The so-called Big Data setup motivates stochastic optimization approach since evaluating the whole function and/or its derivatives is too expensive to be treated by classical, deterministic methods. This raised a number of first-order stochastic strategies proposing different function and/or gradient approximations [1]. Moreover, analogously to deterministic optimization, second-order information can speed up convergence, and this yielded a new direction in stochastic optimization research based on appropriate Hessian. A special case of this approach leans on spectral methods and stochastic Barzilai-Borwein step-sizes, see e.g., [2–5].

Subsampling is a main-stream procedure for obtaining approximations to function and/or gradient evaluations; subsampling strategies range from the mini-batch approach where the sample size is usually small and fixed (e.g., [6]), to increasing sample size strategies where the full sample is eventually reached (e.g., [7]), with many variations in between ([3, 8, 9] to name just a few). Determining a suitable step-size sequence is also a key component in stochastic optimization field. In addition to prefixed constant step-size sequences and diminishing step-sizes, globalization strategies such as line search and trust region can be adapted to the stochastic framework (see e.g. [10–12]) but the majority of such approaches require at least approximate function evaluations.

Finite-sum problems can also come with constraints incorporating prior to knowledge and physical meaning [13–16]. In this work, we propose a stochastic projected gradient method for problem (1) based on the projected gradient method. Function evaluations are not required while a mini-batch approach is employed to compute stochastic gradients. Regarding the mini-batch strategy, the set of indices in the sum (1) is divided into r mini-batches which can be redefined possibly at each iteration. At any iteration k, each mini-batch can be selected with the same probability; a gradient estimate is calculated on the sampled mini-batch and by using an appropriate scaling that provides an unbiased estimate of the full gradient. In order to handle constraints, we allow the use of inexact projections, especially suited in the presence of a large number of constraints (e.g., [17]); specifically, inexact but controlled projections provide a nonmonotone decay of the infeasibility measure. The proposed algorithm is accompanied by a theoretical analysis that establishes convergence results, distinguishing constant and diminishing step-size sequences, in line with the existing literature. Convexity of the objective function is not required. The numerical behavior of the method is shown considering some step-size selections in agreement with the theory.

Outline of the paper. §2 provides the new method while §3 is devoted to its theoretical analysis. §4 provides a comparison with related contributions in the literature. §5 is dedicated to the experimental configuration while numerical results are presented in §6. §7 draws the main conclusions.

Notations. The symbol $\|\cdot\|$ indicates the Euclidean norm. $Pr(\cdot)$ and $\mathbb{E}[\cdot]$ represent the probability function and expected value, respectively.

2 Description of the method

In this section, we introduce our Projected Stochastic Gradient method for Linear Equality COnstrained problems, named the PSG_LECO method, to solve the problem (1). We start by describing two main tasks in the algorithm: the construction of a stochastic gradient and the computation of the projection of a point in $\mathbb{R}^n$ onto S.

At the k-th iteration, given $x_k \in \mathbb{R}^n$, a stochastic gradient g_k is computed as a mini-batch gradient using the following strategy. Let us define a partition $\{\mathcal{N}_k^i\}_{i=1}^r$ of $\mathcal{N} = \{1, \ldots, n\}$ into r disjoint mini-batches at iteration k, namely

$$\mathcal{N} = \bigcup_{i=1}^{r} \mathcal{N}_k^i, \quad \text{where} \quad \mathcal{N}_k^i \cap \mathcal{N}_k^j = \emptyset, \quad \text{for all } i \neq j. \tag{2}$$

This partition can be fixed or varying along the iterations. Then we let $g_k^{(i)}$ be the eligible mini-batch gradients associated with the partition in (2), i.e.,

$$g_k^{(i)} = \frac{r}{N} \sum_{j \in \mathcal{N}_k^i} \nabla f_j(x_k), \quad i = 1, \ldots, r, \tag{3}$$

and g_k be our stochastic gradient that corresponds to a uniformly and randomly selected mini-batch $\mathcal{N}_k^i$ from the partition. By

$$Pr\left(g_k = g_k^{(i)} \mid \mathcal{F}_k\right) = \frac{1}{r}, \quad i = 1, \ldots, r, \tag{4}$$

where $Pr(\cdot)$ represents the probability of the outcomes, and $\mathcal{F}_k$ is a σ-algebra generated by $x_0, \ldots, x_k$, i.e., by $g_0, \ldots, g_{k-1}$, we have

$$\mathbb{E}[g_k \,|\mathcal{F}_k] = \nabla f(x_k). \tag{5}$$

Regarding the projection map onto S, since $A \in \mathbb{R}^{m \times n}$ has full rank, the orthogonal projection $\pi_S(y)$ of any given point $y \in \mathbb{R}^n$ onto S takes the form

$$\begin{aligned} \pi_S(y) &= y - A^T \lambda(y), \\ \lambda(y) &= (AA^T)^{-1}(Ay - b). \end{aligned} \tag{6}$$

The computation of $\pi_S(y)$ is viable for moderate values of m as it depends on the solution of the linear system

$$AA^T \widetilde{\lambda}(y) = Ay - b. \tag{7}$$

More generally, it is advisable to allow for inexact projections of the form

$$\begin{aligned} \widetilde{\pi}_S(y) &= y - A^T \widetilde{\lambda}(y), \\ \widetilde{\lambda}(y) &= (AA^T)^{-1}(Ay - b + r(y)), \end{aligned} \tag{8}$$

and $r(y)$ denotes the residual vector in the solution of (7).

Algorithm 1 sketches the k-th iteration of our procedure. Step 1 indicates the hyper-parameters required for execution: two nonnegative sequences $\{\eta_k\}$ and $\{\mu_k\}$ that control inexactness of the projection onto S, a positive step-size related sequence $\{\alpha_k\}$, two positive scalars δ_ℓ, δ_u that define the projection $\pi_\delta(\cdot) = \max\{\delta_\ell, \min\{\cdot, \delta_u\}\}$ employed in the computation of the step-size.

Step 3 refers to the construction of the stochastic gradient g_k as described above. Steps 4 and 5 concern the computation of the iterate x_{k+1}. Specifically, first, the vector y_k is formed using the step-length Δ_k fixed in the previous iteration, then y_k is projected onto S and this gives rise to the new iterate x_{k+1}. Inequality (9) controls the accuracy in the calculation of the projection of y_k by means of scalars η_k and μ_k. We observe that if $\eta_k = \mu_k = 0$, $\forall k$, then the iterates x_k, $k \geq 1$, are feasible irrespective of x_0.

Steps 6 and 7 are devoted to the computation of the step-length to be used at the subsequent iteration. The choice of a (positive) scalar δ_k offers a variety of options and we discuss some possible adaptive choices in Section 5.

At the end of iteration k, Δ_{k+1} is formed by means of α_{k+1} and $\pi_\delta(\delta_k)$. Consequently, Δ_{k+1} is deterministic conditioning on x_{k+1} and this feature is crucial in the analysis of the procedure. The predetermined sequences $\{\alpha_k\}$ and $\{\mu_k\}$ affect the convergence properties, as shown in the subsequent theoretical analysis.

We conclude this section by giving more insight into the rule (9) in the PSG_LECO Algorithm. Inexact projections are convenient in case the dimension m is large as $\widetilde{\pi}_S(\cdot)$ can be computed using linear iterative solvers, such as the Conjugate Gradient method [18]. Since we accept every iterate without performing an acceptance test, it holds (see the next Lemma 1)

$$\|Ax_{k+1} - b\| = \|r(y_k)\|,$$

where $r(y_k)$ is given in (8). Thus, the inequality (9) implies an explicit relation between the infeasibility measure at x_k and x_{k+1}, i.e., $\|Ax_k - b\|$ and $\|Ax_{k+1} - b\|$, respectively. In addition to such characteristic, the inequality (9) is inspired by the standard control $\|r(y_k)\| \leq \chi_k \|Ay_k - b\|$, $\chi_k \in (0, 1)$, for the approximate solution of the linear system (7). In fact, by the definition of y_k, i.e. $y_k = x_k - \Delta_k g_k$, it holds

$$\|r(y_k)\| \leq \chi_k \|Ay_k - b\| \leq \chi_k \|Ax_k - b\| + \chi_k \Delta_k \|Ag_k\|.$$

Now, (9) rephrases the above inequality using the prescribed term μ_k instead of the random quantity $\chi_k \Delta_k \|Ag_k\|$.

Finally, we note that inexact projections are allowed in [17] by imposing a control of the form $\|r(y_k)\| \leq \eta_k^{\text{IPAS}}$, where $r(y_k)$ is given in (8), the sequence $\{\eta_k^{\text{IPAS}}\}$ is prefixed and $\sum_{k=0}^{\infty} (\eta_k^{\text{IPAS}})^2 < \infty$. Contrary to IPAS, our upper bound on $\|r(y_k)\|$ is adaptive due to the presence of x_k. The assumptions on the sequences $\{\eta_k\}$ and $\{\mu_k\}$ required to ensure convergence to a stationary point of problem (1) are introduced in the next section.

Algorithm 1 PSG_LECO

Choose an initial iterate $x_0 \in \mathbb{R}^n$, a positive sequence $\{\alpha_k\}$, nonnegative sequences $\{\eta_k\}$ and $\{\mu_k\}$ with $\eta_k \in [0, \eta)$, $\eta < 1$, positive scalars $\delta_\ell < \delta_u < \infty$, and an initial step-size $\Delta_0 \in [\alpha_0\delta_\ell, \alpha_0\delta_u]$.
for $k = 0, 1, \ldots$ **do**
  Select uniformly at random a mini-batch $\mathcal{N}_k^i$ from the partition $\{\mathcal{N}_k^1, \ldots, \mathcal{N}_k^\tau\}$ as in (2), and set $g_k = g_k^{(i)}$ as in (3).
  Set $y_k = x_k - \Delta_k g_k$.
  Compute $x_{k+1} = \widetilde{\pi}_S(y_k)$ as in (7), where $r(y_k)$ satisfies

$$\|r(y_k)\| \leq \eta_k \|Ax_k - b\| + \mu_k. \tag{9}$$

  Choose a scalar δ_k.
  Compute $\Delta_{k+1} = \alpha_{k+1}\pi_\delta(\delta_k) = \alpha_{k+1} \max\{\delta_\ell, \min\{\delta_k, \delta_u\}\}$.
  Set $k = k + 1$.
end for

3 Theoretical Analysis

In this section, we analyze the theoretical properties of the PSG_LECO method. We make the following standard assumptions on the objective function.

Assumption 1. *The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on $\mathbb{R}^n$. The gradient of f is Lipschitz continuous with constant $L > 0$.*

Assumption 2. *The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is bounded from below by f^*,*

$$f(x) \geq f^*, \qquad \forall\ x \in \mathbb{R}^n. \tag{10}$$

The following result introduces an optimality measure $d(x)$ for problem (1) which will be used in our subsequent analysis.

Theorem 1. *Assume that f is continuously differentiable in an open set containing S and let*

$$d(x) = \pi_S(x - \nabla f(x)) - x. \tag{11}$$

Then, it holds $d(\bar{x}) = 0$, $\bar{x} \in S$, if and only if $\bar{x}$ is a stationary point for (1).

Proof. See [19, Lemma 2.1]. □

Under suitable assumptions, our analysis will provide the following results which are in accordance with the corresponding algorithms for unconstrained optimization: each limit point of the sequence $\{x_k\}$ is feasible; if $\alpha_k = \alpha$, for all k, then the expected sum of average-squared norms of $d(x_k)$ is bounded and decreases with α; if $\{\alpha_k\}$ is a diminishing sequence, then $\|d(x_k)\|$ cannot stay bounded away from zero almost surely.

The first step in our analysis is to characterize the infeasibility of the iterates generated by Algorithm 1. To this end, we introduce the measures

$$\widetilde{e}(y) = Ay - b, \qquad e(y) = \|\widetilde{e}(y)\|, \tag{12}$$

and make the following assumption on the sequences $\{\eta_k\}$ and $\{\mu_k\}$.

Assumption 3. *The nonnegative sequence* $\{\eta_k\}$ *satisfies* $\eta_k \leq \eta < 1$, $\forall k \geq 0$. *The nonnegative sequence* $\{\mu_k\}$ *converges to zero R-linearly.*

Lemma 1. *Suppose that Assumptions 1 and 3 hold. Then,*

a) For $y \in \mathbb{R}^n$, *it holds*

$$\widetilde{e}(\widetilde{\pi}_S(y)) = -r(y). \tag{13}$$

b) The iterate x_{k+1} *satisfies*

$$\widetilde{e}(x_{k+1}) = -r(y_k), \quad \textit{for all} \quad k \geq 0. \tag{14}$$

c) It holds

$$e(x_k) \leq \kappa_e q^k, \quad \textit{for all} \quad k \geq 0, \tag{15}$$

for some constant $\kappa_e > 0$ *and* $q \in (\eta, 1)$ *and every limit point of* $\{x_k\}$ *is feasible.*

d) It holds

$$e(x_k) \leq \kappa_e, \quad \textit{for all} \quad k \geq 0. \tag{16}$$

Proof. (a) From (8), (7), and (12), we get the expression in (13) as follows:

$$\widetilde{e}(\widetilde{\pi}_S(y)) = A\widetilde{\pi}_S(y) - b = A\left(y - A^T\tilde{\lambda}(y)\right) - b = -(AA^T\tilde{\lambda}(y) - Ay + b).$$

b) Equality (14) follows from (13) and Step 4 of Algorithm 1.

c) Equations (14), (9), (7) and Assumption 3 give

$$e(x_{k+1}) \leq \eta e(x_k) + \mu_k.$$

Therefore, for all k, there holds

$$e(x_k) \leq \eta^k e(x_0) + v_k, \qquad v_k = \sum_{i=1}^{k} \eta^{i-1}\mu_{k-i}. \tag{17}$$

Since $\{\mu_k\}$ converges to zero R-linearly and $\eta < 1$, $\{v_k\}$ converges to zero R-linearly which implies (15), see e.g., [20, Lemma 4.2]. Thus, $\lim_{k\to\infty} e(x_k) = 0$.

d) The claim trivially follows from (15). □

We note that feasibility is eventually enforced, i.e., $\lim_{k\to\infty} e(x_k) = 0$, whenever $\{\mu_k\}$ tends to zero. On the other hand, the summability of $\sum_{k=0}^{\infty} \mu_k$ is required to prove the main Theorem 2; this request motivates our Assumption 3 on $\{\mu_k\}$.

Now we proceed with an intermediate result on the step d_k taken at iteration k

$$d_k = x_{k+1} - x_k. \tag{18}$$

We denote

$$D = A^T(AA^T)^{-1}, \qquad P_A = I - DA, \qquad \kappa_D = \|D\|, \tag{19}$$

and introduce the following assumptions.

Assumption 4. *For some positive constant κ_∇, the sequence $\{x_k\}$ is either feasible or such that*

$$\|\nabla f(x_k)\| \leq \kappa_\nabla, \qquad \text{for all} \quad k \geq 0.$$

Note that bounded gradients are assumed only when the iterates are infeasible. We now consider the following assumption, which states that the conditional variance of the sampled gradient is uniformly bounded.

Assumption 5. *There exists a positive constant $\nu > 0$ such that*

$$\mathbb{E}\big[\|g_k - \nabla f(x_k)\|^2 \,|\mathcal{F}_k\big] \leq \nu. \tag{20}$$

Lemma 2. *Let $\{x_k\}$ be generated by Algorithm* PSG_LECO. *Suppose that Assumption 1 holds.*

a) The step d_k defined in (18) satisfies

$$\mathbb{E}\big[d_k \,|\mathcal{F}_k\big] = \Delta_k d(x_k) - (1-\Delta_k) D\widetilde{e}(x_k) - D\mathbb{E}\big[r(y_k)\,|\mathcal{F}_k\big], \tag{21}$$

where D is defined in (19).
b) Suppose further that Assumptions 3 and 4 hold. Then, $d(x_k)$ defined in (11) satisfies

$$d(x_k)^T \nabla f(x_k) \leq -\|d(x_k)\|^2 + \kappa_1 e(x_k), \tag{22}$$

for some positive constant κ_1.
c) Suppose further that Assumptions 3, 4 and 5 hold, and that $(1-\Delta_k)^2 \leq \Delta^*$, for all k and some positive Δ^*. Then,

$$\mathbb{E}\big[\|d_k\|^2 \,|\mathcal{F}_k\big] \leq \kappa_2 \Delta_k^2 + \kappa_3 q^{2k} + \kappa_4 \mu_k + 2\Delta_k^2 \|d(x_k)\|^2, \tag{23}$$

for some positive constants κ_2, κ_3, κ_4.

Proof. a) First, we note that (18), (19), (8) and (7) give

$$\begin{aligned} d_k &= \widetilde{\pi}_S(x_k - \Delta_k g_k) - x_k \\ &= x_k - \Delta_k g_k - D(Ax_k - \Delta_k A g_k - b + r(y_k)) - x_k \\ &= -\Delta_k P_A g_k - D\widetilde{e}(x_k) - Dr(y_k), \end{aligned} \tag{24}$$

and that (11), (6) give

$$d(x_k) = -P_A \nabla f(x_k) - D\widetilde{e}(x_k). \tag{25}$$

Hence, using (5) and that Δ_k is $\mathcal{F}_k$ -measurable, we have

$$\begin{aligned} \mathbb{E}\big[d_k \,|\mathcal{F}_k\big] &= -\Delta_k P_A \mathbb{E}\big[g_k \,|\mathcal{F}_k\big] - D\widetilde{e}(x_k) - D\mathbb{E}\big[r(y_k)\,|\mathcal{F}_k\big] \\ &= -\Delta_k P_A \nabla f(x_k) - D\widetilde{e}(x_k) - D\mathbb{E}\big[r(y_k)\,|\mathcal{F}_k\big], \end{aligned}$$

and (25) concludes the proof.

b) Using (25), $P_A D = 0$, $P_A^T = P_A$ and $P_A^2 = P_A$, it follows

$$\begin{aligned} d(x_k)^T d(x_k) &= \nabla f(x_k)^T P_A^2 \nabla f(x_k) + \|D\widetilde{e}(x_k)\|^2 \\ &= -\nabla f(x_k)^T d(x_k) - \nabla f(x_k)^T D\widetilde{e}(x_k) + \|D\widetilde{e}(x_k)\|^2. \end{aligned}$$

Hence, using (12), (16) we obtain

$$\begin{aligned} \nabla f(x_k)^T d(x_k) &\leq -\|d(x_k)\|^2 + \|\nabla f(x_k)\| \, \kappa_D e(x_k) + \kappa_D^2 e(x_k)^2 \\ &\leq -\|d(x_k)\|^2 + \big(\kappa_\nabla \kappa_D + \kappa_D^2 \kappa_e\big) e(x_k). \end{aligned}$$

Thus, (22) holds with $\kappa_1 = \kappa_\nabla \kappa_D + \kappa_D^2 \kappa_e$.

c) Using (24), (25), $\|P_A\| = 1$, and (15) we obtain

$$\begin{aligned} \|d_k - \Delta_k d(x_k)\|^2 &= \| - \Delta_k P_A(g_k - \nabla f(x_k)) - (1-\Delta_k) D\widetilde{e}(x_k) - Dr(y_k)\|^2 \\ &\leq 2\Delta_k^2 \|g_k - \nabla f(x_k)\|^2 + 4\|(1-\Delta_k) D\widetilde{e}(x_k)\|^2 + 4\|Dr(y_k)\|^2 \\ &\leq 2\Delta_k^2 \|g_k - \nabla f(x_k)\|^2 + 4\kappa_D^2 \Delta^* e(x_k)^2 + 4\kappa_D^2 (\eta_k e(x_k) + \mu_k)^2 \\ &\leq 2\Delta_k^2 \|g_k - \nabla f(x_k)\|^2 + 4\kappa_D^2 (\Delta^* + \eta^2)\kappa_e^2 q^{2k} + 4\kappa_D^2 (2\eta\kappa_e q^k + \mu_k)\mu_k. \end{aligned}$$

Now, using this last inequality, Assumption 5 and the equation

$$\|d_k\|^2 = \|d_k - \Delta_k d(x_k) + \Delta_k d(x_k)\|^2 \leq 2\|d_k - \Delta_k d(x_k)\|^2 + 2\Delta_k^2 \|d(x_k)\|^2,$$

we obtain

$$\begin{aligned} \mathbb{E}\big[\|d_k\|^2 \,|\mathcal{F}_k\big] &\leq 2\mathbb{E}\big[\|d_k - \Delta_k d(x_k)\|^2 \,|\mathcal{F}_k\big] + 2\Delta_k^2 \mathbb{E}\big[\|d(x_k)\|^2 \,|\mathcal{F}_k\big] \\ &\leq 4\Delta_k^2 \mathbb{E}\big[\|g_k - \nabla f(x_k)\|^2 \,|\mathcal{F}_k\big] + 8\kappa_D^2 (\Delta^* + \eta^2)\kappa_e^2 q^{2k} \\ &\quad + 8\kappa_D^2 (2\eta\kappa_e q^k + \mu_k)\mu_k + 2\Delta_k^2 \|d(x_k)\|^2. \end{aligned} \tag{26}$$

Thus, the claim holds with $\kappa_2 = 4\nu$, $\kappa_3 = 8\kappa_D^2(\Delta^* + \eta^2)\kappa_e^2$, $\kappa_4 = 8\kappa_D^2(2\eta\kappa_e + \mu_{\max})$ where $\mu_{\max} = \max_k \{\mu_k\}$. □

For sake of completeness, the following lemma rephrases the results above when x_0 is feasible and the exact projection π_S is used.

Corollary 1. *Let $\{x_k\}$ be generated by Algorithm* PSG_LECO. *Suppose that Assumption 1 holds, $x_0 \in S$, $\eta_k = \mu_k = 0$ for all $k \geq 0$.*
a) The sequence $\{x_k\}$ is feasible.
b) The step d_k defined in (18) satisfies

$$\mathbb{E}\big[d_k \,|\mathcal{F}_k\big] = \Delta_k d(x_k). \tag{27}$$

c) The direction $d(x_k)$ defined in (11) satisfies

$$d(x_k)^T \nabla f(x_k) \leq -\|d(x_k)\|^2. \tag{28}$$

d) Suppose further that Assumption 5 holds and that $(1-\Delta_k)^2 \le \Delta^*$ *for all* k *and some positive* Δ^**. Then,*

$$\mathbb{E}\big[\|d_k\|^2 \,|\mathcal{F}_k\big] \le \kappa_2 \Delta_k^2 + 2\Delta_k^2 \|d(x_k)\|^2, \qquad \textit{for some} \quad \kappa_2 > 0. \tag{29}$$

Proof. Item (a) follows from (17), while items (b)–(d) follow from Lemma 2. □

By the following theorem, we now analyze the behavior of the PSG_LECO method under the assumption that the sequence $\{\alpha_k\}$ is diminishing.

Theorem 2. *Suppose that Assumptions 1–5 hold, and that*

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty. \tag{30}$$

Then, almost surely,

$$\liminf_{k\to\infty} \|d(x_k)\| = 0, \tag{31}$$

and the sequence of function values $\{f(x_k)\}$ *converges.*

Proof. Assumption 1 implies that

$$f(x) \le f(y) + \nabla f(y)^T (x-y) + \frac{L}{2}\|x-y\|^2, \qquad \forall\, x, y \in \mathbb{R}^n. \tag{32}$$

Hence, by (18) we have

$$f(x_{k+1}) \le f(x_k) + \nabla f(x_k)^T d_k + \frac{L}{2}\|d_k\|^2.$$

Taking the conditional expectation with respect to the σ-algebra $\mathcal{F}_k$, we obtain

$$\mathbb{E}\big[f(x_{k+1}) \,|\mathcal{F}_k\big] \le f(x_k) + \nabla f(x_k)^T \mathbb{E}\big[d_k \,|\mathcal{F}_k\big] + \frac{L}{2}\,\mathbb{E}\big[\|d_k\|^2 \,|\mathcal{F}_k\big].$$

Conditions (30) imply $(1-\Delta_k)^2 \le \Delta^*$ for some $\Delta^* > 0$. Now, by (21)–(23), we have

$$\begin{aligned}
\mathbb{E}\big[f(x_{k+1}) \,|\mathcal{F}_k\big] &\le f(x_k) - \Delta_k \|d(x_k)\|^2 + \kappa_1 \Delta_k e(x_k) - (1-\Delta_k)\nabla f(x_k)^T D\widetilde{e}(x_k) \\
&\quad - \mathbb{E}\big[\nabla f(x_k)^T D r(y_k) \,|\mathcal{F}_k\big] + \frac{L}{2}(\kappa_2 \Delta_k^2 + \kappa_3 q^{2k} + \kappa_4 \mu_k + 2\Delta_k^2 \|d(x_k)\|^2) \\
&\le f(x_k) - \Delta_k \|d(x_k)\|^2 + \kappa_1 \Delta_k e(x_k) + \sqrt{\Delta^*}\|\nabla f(x_k)\| \kappa_D e(x_k) \\
&\quad + \|\nabla f(x_k)\| \kappa_D \mathbb{E}\big[\|r(y_k)\| \,|\mathcal{F}_k\big] + \frac{L}{2}(\kappa_2 \Delta_k^2 + \kappa_3 q^{2k} + \kappa_4 \mu_k + 2\Delta_k^2 \|d(x_k)\|^2).
\end{aligned}$$

If $\{x_k\}$ is feasible, then $r(y_k) = 0$. Otherwise, the upper bound $\eta e(x_k) + \mu_k$ on $\|r(y_k)\|$ given in (9) is $\mathcal{F}_k$-measurable, and we obtain the following inequality that

includes exact and inexact projections,

$$\begin{aligned}\mathbb{E}\big[f(x_{k+1})\,|\mathcal{F}_k\big] &\leq f(x_k) - \Delta_k\|d(x_k)\|^2 + \big((1+\sqrt{\Delta^*})\kappa_1 + \kappa_\nabla\kappa_D(\sqrt{\Delta^*}+\eta)\big)e(x_k) \\ &\quad + \kappa_\nabla\kappa_D\mu_k + \frac{L}{2}(\kappa_2\Delta_k^2 + \kappa_3 q^{2k} + \kappa_4\mu_k + 2\Delta_k^2\|d(x_k)\|^2) \\ &\leq f(x_k) - \Delta_k(1 - L\Delta_k)\|d(x_k)\|^2 + \kappa_5 q^k + \kappa_6\mu_k + \frac{L}{2}\kappa_2\Delta_k^2, \qquad (33)\end{aligned}$$

with $\kappa_5 = \Big(\kappa_1(1+\sqrt{\Delta^*}) + \kappa_\nabla\kappa_D(\sqrt{\Delta^*}+\eta)\Big)\kappa_e + \frac{L}{2}\kappa_3$, $\kappa_6 = \Big(\kappa_\nabla\kappa_D + \frac{L}{2}\kappa_4\Big)$ and the last inequality obtained using (15).

By construction Δ_k satisfies

$$\alpha_k\delta_\ell \leq \Delta_k \leq \alpha_k\delta_u, \qquad (34)$$

hence

$$\mathbb{E}\big[u_{k+1}\,|\mathcal{F}_k\big] \leq u_k - \alpha_k\big(\delta_\ell - L\alpha_k\delta_u^2\big)\|d(x_k)\|^2 + \kappa_5 q^k + \kappa_6\mu_k + \frac{L}{2}\kappa_2\alpha_k^2\delta_u^2, \qquad (35)$$

where $u_k = f(x_k) - f^*$ from (10). Since $\{\alpha_k\}$ is diminishing, let $\bar{k}$ be the index such that $\big(\delta_\ell - L\alpha_k\delta_u^2\big) \geq \frac{\delta_\ell}{2}$ for all $k \geq \bar{k}$. Hence, for $k \geq \bar{k}$

$$\mathbb{E}\big[u_{k+1} \mid \mathcal{F}_k\big] \leq u_k - \alpha_k\frac{\delta_\ell}{2}\|d(x_k)\|^2 + \kappa_5 q^k + \kappa_6\mu_k + \frac{L}{2}\kappa_2\alpha_k^2\delta_u^2.$$

Since $\sum_{k=0}^{\infty}\alpha_k^2$, $\sum_{k=0}^{\infty}q^k$ and $\sum_{k=0}^{\infty}\mu_k$ are summable, the Robbins–Siegmund supermartingale convergence Theorem [21] gives that

$$\sum_{k=\bar{k}+1}^{\infty}\alpha_k\|d(x_k)\|^2 < \infty,$$

almost surely. Now, assume by contradiction that $\|d(x_k)\| \geq c > 0$ for all $k > \bar{k}$. Thus

$$\sum_{k=\bar{k}+1}^{\infty}\alpha_k\|d(x_k)\|^2 \geq c^2\sum_{k=\bar{k}+1}^{\infty}\alpha_k,$$

which contradicts (30). Therefore, (31) holds almost surely. Finally, from the Robbins–Siegmund supermartingale convergence theorem, we also conclude that $\lim_{k\to\infty}u_k$ exists and is finite almost surely. □

The theorem above implies that almost surely there exists a subsequence $\{\|d(x_k)\|\}_{k\in K}$ of $\{\|d(x_k)\|\}$ convergent to zero; if $\{x_k\}_{k\in K}$ admits limit points, then such points are stationary. Now we analyze the convergence of the PSG_LECO using constant step-lengths and show that the expected sum of average-squared norms of $d(x_k)$ is bounded and decreases with α.

Theorem 3. *Suppose that Assumptions 1–5 hold, $\eta_k \leq \eta < 1, \forall k$. If $\alpha_k = \alpha$, $\forall k \geq 0$, with α such that*

$$0 < \alpha \leq \frac{\delta_\ell}{2L\delta_u^2}, \tag{36}$$

then the iterates generated by PSG_LECO *method satisfy*

$$\lim_{K\to\infty} \mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K} \|d(x_k)\|^2\right] \leq \frac{\alpha L\kappa_2\delta_u^2}{\delta_\ell}, \tag{37}$$

where κ_2 is a scalar in (23).

Proof. Condition (36) implies $(1-\Delta_k)^2 \leq \Delta^*$, $\forall k$, and some positive Δ^*. Applying (34) and (36) in (33) and taking total expectation, we obtain the following

$$\mathbb{E}\big[f(x_{k+1})\big] \leq \mathbb{E}\big[f(x_k)\big] - \frac{1}{2}\alpha\delta_\ell\mathbb{E}\big[\|d(x_k)\|^2\big] + \kappa_5 q^k + \kappa_6\mu_k + \frac{L}{2}\kappa_2\alpha^2\delta_u^2,$$

and consequently

$$\mathbb{E}\big[\|d(x_k)\|^2\big] \leq \frac{2}{\alpha\delta_\ell}\left(\mathbb{E}\big[f(x_k)\big] - \mathbb{E}\big[f(x_{k+1})\big]\right) + \frac{2\kappa_5}{\alpha\delta_\ell}q^k + \frac{2\kappa_6}{\alpha\delta_\ell}\mu_k + \frac{\alpha L\kappa_2\delta_u^2}{\delta_\ell}.$$

Summing both sides of this inequality for $k = 0, \ldots, K$, and dividing by K, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K}\|d(x_k)\|^2\right] &\leq \frac{2}{\alpha\delta_\ell K}\left(f(x_0) - \mathbb{E}\big[f(x_{K+1})\big]\right) + \frac{2\kappa_5}{\alpha\delta_\ell K}\sum_{k=0}^{K} q^k \\
&\quad + \frac{2\kappa_6}{\alpha\delta_\ell K}\sum_{k=0}^{K}\mu_k + \frac{\alpha L\kappa_2\delta_u^2}{\delta_\ell} \\
&\leq \frac{2}{\alpha\delta_\ell K}(f(x_0) - f^*) + \frac{2\kappa_5}{\alpha\delta_\ell K}\sum_{k=0}^{K} q^k \\
&\quad + \frac{2\kappa_6}{\alpha\delta_\ell K}\sum_{k=0}^{K}\mu_k + \frac{\alpha L\kappa_2\delta_u^2}{\delta_\ell}.
\end{aligned}$$

Since $\sum_{k=0}^{\infty} q^k$ and $\sum_{k=0}^{\infty}\mu_k$ are summable, the claim follows. □

4 Related Work

The extension of methods with random models for the unconstrained setting to the setting of deterministic equality and inequality constrained problems is a recent area of research which is drawing much interest, see [13–17, 22–24]. Focusing on papers [13, 15–17] for deterministic equality constrained optimization, we sketch their main features.

The work [17] introduces an Inexact Projection with Additional Sampling gradient method (IPAS) for weighted-sum minimization with linear equality constraints; we refer to the end of §2 for the description of the control imposed on $\|r(y_k)\|$.

Despite inexact projections in IPAS and in PSG_LECO being controlled differently, the sequence $\{e(x_k)\}$ of infeasibility measures behaves similarly (cf. our Lemma 1 and [17, Lemma 4.2]).

IPAS employs stochastic estimates of the objective function. Specifically, functions and gradients are approximated by sampling and the step-size is determined by a non-monotone line-search rule over an approximate objective function. Additional sampling is used to decide whether a trial point should be accepted, as well as to decide if the sample size needs to be increased. Although this provides an adaptive batch size strategy, the additional sampling combined with the line search yields additional costs and a more elaborated algorithm with respect to the scheme proposed in this paper. In principle, PSG_LECO also allows a variable sample size scheme, but it lacks a precise rule for its guidance.

Papers [13, 15, 16] present objective function-free procedures in the class of Sequential Quadratic Programming (SQP) algorithms. The procedures in [15, 16] employ stochastic gradients of the objective function and prescribed approximations of the Hessian of the objective function and/or a Lagrangian function; the step-size selection is adaptive and is based on estimated Lipschitz constants. In particular, in [15] the search direction results from the use of a merit function with ℓ_1-norm penalty function; it is computed solving a quadratic optimization problem based on a local quadratic minimizer of the objective function and a local affine model of the constraint. The choice of the step-size is inspired by a line search strategy and is based on a rule using Lipschitz constant estimates. The hyper-parameters of the algorithm are: a sequence of Lipschitz constant estimates of the objective function, a sequence of Lipschitz constant estimates of the constraint function, and a sequence to control the step-size. Assuming unbiased stochastic gradients and condition (20), the analysis shows that the generated sequence achieves stationarity and feasibility in expectation. For the linearly constrained case considered here, the algorithms in [13, 15, 16] generate feasible iterates.

In [16] the search direction is decomposed into a normal step and a tangential step. Exact projections are supposed to be computable and consequently the normal step has a closed form. On the other hand, the tangential step solves a trust-region problem which employs a basis for the null space of the Jacobian of the constraints. The adaptive trust-region radius is computed using estimates of the Lipschitz constants of both the objective function and the constraint functions.The hyper-parameters of the algorithm are: a sequence of Lipschitz constant estimates of the objective function, a sequence of Lipschitz constant estimates of the constraint function, and two sequences of positive scalars to control the trust-region radius. Assuming that the stochastic gradient is unbiased and condition (20), the theoretical analysis shows that KKT residuals converge to zero almost surely.

Finally, the recent work [13] extends stochastic momentum methods for unconstrained optimization to the stochastic SQP setting and proposes two algorithms: a projected stochastic heavy-ball SQP method and a projected stochastic Adam SQP

method. These methods require exact projections, use projected stochastic gradient estimates in the momentum terms, and involve two predefined step-size-related sequences. Assuming unbiased stochastic projected gradients, condition (20), and boundedness of $\{\|\nabla f(x_k)\|\}$, their convergence behavior is shown to be analogous to that of the corresponding unconstrained methods. In contrast, PSG_LECO allows inexact projections.

5 Experimental Configuration

All runs were performed in MATLAB R2025a on a workstation equipped with an Intel Ultra 9 processor, 128 GB DDR5 RAM, dual 2 TB SSDs, and an NVIDIA RTX PRO 4000 Blackwell GPU (24 GB), without using GPU acceleration. First, we describe the test problems and our numerical implementation. Second, we show the performance of PSG_LECO Algorithm. Finally, we compare our procedure with two algorithms recently proposed.

5.1 Test problems

We applied PSG_LECO in the solution of six problems from the literature. The complete description is given below.

Equality-constrained logistic regression problems

We used MUSHROOMS, MNIST, and DIABETES datasets from LIBSVM[1] [25]. For MUSHROOMS, we mapped labels $1, 2$ to $+1, -1$ respectively, and used the feature matrix as returned. For the multi-class dataset MNIST, we restricted the dataset to a binary one with digits $0, 8$ and mapped 0 to $+1$ and 8 to -1; all features were scaled to $[0, 1]$. For DIABETES, we remapped labels $0, 1$ to $-1, +1$ respectively, and scaled the features to $[-1, 1]$. Given $\{(z_i, y_i)\}_{i=1}^N$ with $z_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$, we define the objective function in (1) [2]

$$f(x) \;=\; \frac{1}{N}\sum_{i=1}^{N} \log\bigl(1 + \mathrm{e}^{-y_i z_i^\top x}\bigr). \tag{38}$$

The constraint matrix $A \in \mathbb{R}^{m\times n}$ and the vector $b \in \mathbb{R}^m$ were generated using a fixed random seed (`rng(0)`); as for n, m and N, see Table 1. We used the initial iterate $x_0 = A^T(AA^T)^{-1}b$ for the runs with exact projection and $x_0 = 0$ for runs with inexact projection.

Equality-constrained* Cutest *problems

We considered the problems HUESTIS, DTOC1L, and HS50 subject to linear constraints. Following [17], let $\widetilde{f}(x) : \mathbb{R}^n \to \mathbb{R}$ denote the objective function in the CUTEst collection. We define a finite-sum objective function for (1) as

$$f(x) = \widetilde{f}(x) + \sum_{i=1}^{N} \xi_i^2 \,\|x\|_2^2, \tag{39}$$

[1] We used the files `Mnist.mat` and `diabetes.txt` from the LIBSVM Library.
[2] In our runs we used the stable formulation of the logistic regression.

Table 1: Test problems.

	MNIST	MUSHROOMS	DIABETES	DTOC1L	HUESTIS	HS50
N	11774	8124	768	10000	10000	10000
n	780	112	8	58	10	5
m	$\lfloor 0.5n \rfloor$	$\lfloor 0.5n \rfloor$	$\lfloor 0.5n \rfloor$	36	2	3

where each ξ_i is an independent random sample drawn from a Gaussian distribution, i.e., $\xi_i \sim \mathcal{N}(0, \sigma^2)$, with $\sigma = 0.1$ and fixed random seed (`rng(0)`). The constraint matrix $A \in \mathbb{R}^{m \times n}$ and the initial feasible point x_0 are given in the collection. The values of n, m and N are given in Table 1.

5.2 Algorithmic Implementation

For the computation of the stochastic gradients, the partition $\{\mathcal{N}_k^i\}_{i=1}^r$ in (2) was kept fixed along the iterations. Unless explicitly stated, the mini-batch size was set to 64 for DIABETES and to 256 for the other datasets. We built ten independent random partitions, each of which created by using an independent random seed[3]. For each of such partitions, the mini-batches were selected randomly and independently along the iteration.

Regarding the selection of the step-sizes in PSG_LECO, we set $\delta_\ell = 10^{-3}$, $\delta_u = 10^2$. We tested three strategies denoted **S1**, **S2** and **S3**. They differ in the choice of the scaling parameter α_k and/or of the parameter δ_k. In **S1**, we considered a fixed sequence $\{\alpha_k\}$, $\alpha_k = \alpha$ for all $k \geq 0$, while the scalar δ_k was computed with a Barzilai-Borwein approach due to the potential of such step-sizes in a stochastic environment, see e.g., [5, 26]. Our requirement that Δ_k is fully determined at x_k motivated the use of the retarded Barzilai-Borwein steps [26, 27], defined as

$$\delta_k = \left| \frac{d_{k-\widetilde{q}}^T d_{k-\widetilde{q}}}{d_{k-\widetilde{q}}^T z_{k-\widetilde{q}}} \right|,$$

where $\widetilde{q} = \min\{q, k-1\}$ with $q \geq 1$, and $d_{t-1} = x_t - x_{t-1}$, $z_{t-1} = g_t - g_{t-1}$ with $t \geq 1$; see [26, Eq. (2.3)]. Setting $q = 1$ yields

$$\delta_k = \left| \frac{d_{k-1}^T d_{k-1}}{d_{k-1}^T z_{k-1}} \right|. \tag{40}$$

Following guidelines on stochastic Barzilai-Borwein step-lengths, it is advisable to compute z_t using stochastic gradients with the same mini-batch; since this would require an additional gradient evaluation at each iteration, we updated δ_k using (40) every 20 iterations. The strategy **S1** is adaptive due to the form of δ_k, and the sequence $\{\Delta_k\}$ is not supposed to be decreasing.

[3] If N is not a multiple of the number of partitions in (3), one mini-batch in (2) has smaller cardinality.

Table 2: Strategies for computing the step-length Δ_{k+1}.

Strategy	α_{k+1}	δ_k
S1	$\alpha > 0$	δ_k in (40)
S2	α_{k+1} in (41)-(42)	δ_k in (40)
S3	α_{k+1} in (41)–(42)	$\delta_k = 1$

In **S2**, we considered δ_k as in (40), and the diminishing parameter α_{k+1} of the form

$$\alpha_{k+1} = \frac{a}{a+k} c_k(\gamma_0, \gamma_1), \tag{41}$$

$$c_k(\gamma_0, \gamma_1) = \gamma_1 + 0.5(\gamma_0 - \gamma_1)\Big(1 + \cos\Big(\frac{k\pi}{k_{\max}}\Big)\Big), \tag{42}$$

with $\gamma_0, \gamma_1 > 0$ and $k_{\max}$ representing the maximum number of iterations allowed, [28, 29]. The cosine-decay step rule (42) implies $c_0(\gamma_0, \gamma_1) = \gamma_0$, $c_{k_{\max}}(\gamma_0, \gamma_1) = \gamma_1$. In fact, the strategy **S2** is still adaptive because of δ_k, but now Δ_k is driven in intervals of diminishing length, i.e., from $[\gamma_0\delta_\ell, \gamma_0\delta_u]$ to $[\frac{a\gamma_1}{a+k_{\max}}\delta_\ell, \frac{a\gamma_1}{a+k_{\max}}\delta_u]$ as k reaches $k_{\max}$.

In the strategy **S3** we fixed $\delta_k = 1$ for all $k \geq 0$, and set α_{k+1} as in (41)-(42). Taking into account that $\pi_\delta(1) = 1$ for reasonable values of δ_ℓ, δ_u, we have $\Delta_k = \alpha_k$ and decreasing step-sizes for increasing values of k.

Table 2 summarizes the three strategies **S1**-**S3**. In our experiments we set $a = 1000$ in (41) and $\gamma_1 = 10^{-5}$ in (42). The parameter α in **S1** and the parameter γ_0 in **S2** and **S3** were varied; we tuned α and γ_0 exploring a set of prefixed values as shown in §6.1. The initial step-size is set to $\Delta_0 = \alpha\delta_\ell$ in **S1** and $\Delta_0 = \gamma_0\delta_\ell$ in **S2** and **S3**.

The exact projection π_S in (6) was computed by the Cholesky factorization of AA^T. Exploiting this factorization, we precomputed the matrix $D = A^T(AA^T)^{-1}$ once prior to the iterative process. Afterwards, the projection computation required products of the matrix D times a vector.

The evaluation of the inexact projection $\widetilde{\pi}_S$ in (8) was performed applying the Conjugate Gradient Method (Matlab built-in function `pcg`) to (7) with stopping criterion (9). We set $\eta_k = \eta > 0$ for all $k \geq 0$, and $\mu_k = \mu_0\rho^k$, for all $k \geq 0$, $\mu_0 > 0$ and $\rho \in (0, 1)$; the ablation study on η, μ_0 and ρ is given in §6.1.2. The stopping criteria of `pcg` are the relative residual condition $\|r(y_k)\|/\|Ay_k - b\| < \tau$ and the maximum iteration limit (ITER). We set ITER $= m$ and let

$$\tau = \max\left\{10^{-10}, \min\left\{\frac{\eta_k\|Ax_k - b\| + \mu_k}{\|Ay_k - b\|}, 10^{-3}\right\}\right\}.$$

The thresholds above avoid too small and too large values of $\|r(y_k)\|$. Moreover, to save computations, we skipped projection when $\|Ay_k - b\| \leq 10^{-12}$.

If not explicitly stated, each run consists of $k_{\max} = 10^4$ iterations. For each configuration of the parameters involved, the evaluation metrics were averaged over ten independent runs to construct the mean performance values. In the generated plots, the y-axis displays these metrics on a base-10 logarithmic scale.

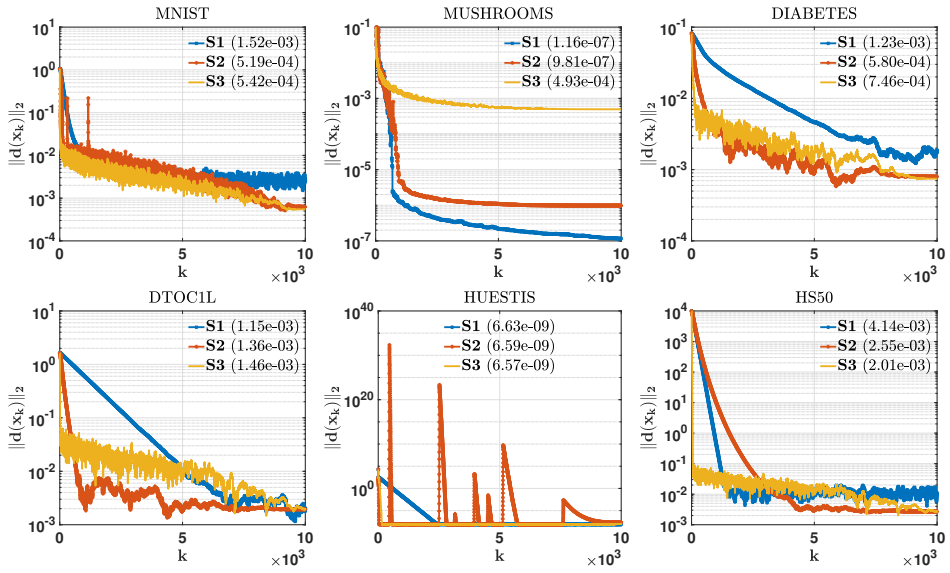

Fig. 1: The average values of $\|d(x_k)\|$ obtained by PSG_LECO using the exact projection and step -size strategies **S1-S3** with tuned α and γ_0.

6 Numerical Results

In §6.1 we analyze the performance of our algorithm implemented with the step-size strategies **S1-S3** and run for varying hyper-parameters, including the batch-size. In §6.2 we analyze the performance of PSG_LECO when the projection is calculated inexactly. In § 6.3 and § 6.4, we compare the performance of PSG_LECO with the Projected Stochastic Adam SQP method (SQP_ADAM) proposed in [13] and with the IPAS algorithm proposed in [17], respectively.

6.1 Performance of PSG_LECO and Ablation Study

6.1.1 Step-Size Selection

PSG_LECO Algorithm is an adaptation of the Stochastic Gradient method to the linearly constrained case and inherits its dependence from the step-size selection from both a theoretical and numerical point of view. Hence, our numerical analysis of PSG_LECO method was first conducted using the exact projection π_S in (6) and focusing on the strategy for choosing the step-size and the parameters therein: the parameter α in strategy **S1**, and the parameter γ_0 in strategies **S2** and **S3**. In principle, in the strategies **S1** and **S2** the frequency of the update (40) of δ_k may play a role, but, as briefly shown at the end of this section, we experimentally verified that it marginally affects the performance; for this reason, the discussion below refers to the update (40) of δ_k every 20 iterations, with the batch-size as specified in §5.2.

Table 3: The minimum average value of $\|d(x_k)\|$ obtained by PSG_LECO over all the datasets, using the exact projection.

	MNIST			**MUSHROOMS**		
α, γ_0	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
10^{-3}	7.326426e-03	2.083257e-01	2.788223e-01	1.013773e-02	9.433663e-02	2.964358e-01
10^{-2}	1.518623e-03	3.853655e-03	4.195787e-02	1.099648e-03	6.204044e-03	6.876800e-02
10^{-1}	2.034988e-03	7.589910e-04	6.799109e-03	1.036599e-04	6.369529e-04	1.472482e-02
1	3.859859e-03	5.192025e-04	1.444848e-03	7.020857e-06	3.794461e-05	2.981289e-03
10	4.092225e-03	5.560996e-04	5.421119e-04	1.158238e-07	9.810904e-07	4.932504e-04
	DIABETES			**DTOC1L**		
α, γ_0	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
10^{-3}	1.228244e-03	1.921677e-02	7.151869e-02	1.154305e-03	2.907515e-01	1.458438e-03
10^{-2}	1.638480e-03	5.799938e-04	3.518885e-02	2.698684e-03	1.361814e-03	1.806219e-03
10^{-1}	4.407748e-03	8.521068e-04	2.688412e-03	5.849365e-03	1.747942e-03	1.608953e+00
1	2.432084e-02	1.463002e-03	7.456083e-04	3.145822e-02	1.817416e-03	1.315883e+00
10	7.998332e-02	2.392200e-03	9.455264e-04	1.273853e+00	1.273853e+00	1.614671e+00
	HUESTIS			**HS50**		
α, γ_0	**S1**	**S2**	**S3**	**S1**	**S2**	**S3**
10^{-3}	1.389308e-02	5.864945e+01	6.663500e-09	4.170339e-01	1.755844e+03	2.008578e-03
10^{-2}	6.625383e-09	1.024091e-05	6.571084e-09	4.142960e-03	2.547671e-03	9.893982e+03
10^{-1}	6.572476e-09	6.383179e-09	3.253096e+02	8.063442e-03	2.399365e-03	9.711535e+03
1	6.833632e-09	6.590809e-09	2.686514e+02	4.337829e-02	2.503668e-03	7.920287e+03
10	2.064706e+02	2.064706e+02	2.979305e+02	9.914297e+03	1.078018e+04	1.672371e+05

We conducted our experiments by varying the hyper-parameters α and γ_0 within the range $\mathcal{T}_1 = \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. The remaining hyper-parameters in PSG_LECO were specified in §5.2. In Table 3, we display the smallest average value of $\|d(x_k)\|$ obtained for varying strategies and parameters. The performance of PSG_LECO depends on the step-size rule and below we discuss the results obtained.

For values of α in the set $\{10^{-2}, 10^{-1}, 1\}$, the strategy **S1** achieves low values for optimality measure and the value of α influences the performance only in the solution of the problem MUSHROOMS. In general, using the smallest value $\alpha = 10^{-3}$ is not effective. Referring to a failure in the case when the smallest achieved value of $\|d(x_k)\|$ is higher than 10^{-1}, we observe failures for the value $\alpha = 10$ in problems DTOC1L, HUESTIS and HS50. The strategy **S2** shows failures in the solution of the problems DTOC1L, HUESTIS and HS50 when $\gamma_0 = 10^{-3}, 10$. Excluding failures, out of the complete runs, the accuracy achieved by PSG_LECO with **S2** is higher or comparable to that of PSG_LECO with **S1** except for problem MUSHROOMS. Further, employing **S2** outperforms **S3** in terms of optimality. The strategy **S3** appears to be the less robust among the three tested. Failures occur for values γ_0 in the set $\{10^{-2}, 10^{-1}, 1, 10\}$ in the solution of DTOC1L, HUESTIS and HS50.

The results obtained suggest that PSG_LECO Algorithm coupled with the strategy **S2** attains the highest accuracy, and that the adaptive choice for δ_k positively affects the performance of the method. In order to provide more insight into PSG_LECO Algorithm, in Figure 1 we plot the average values of $\|d(x_k)\|$ versus the iterations obtained with the strategies **S1**-**S3**; the results displayed correspond to the tuned parameters which gave comparatively better performance in terms of optimality in the experimental study presented above and are reported in Table 4.

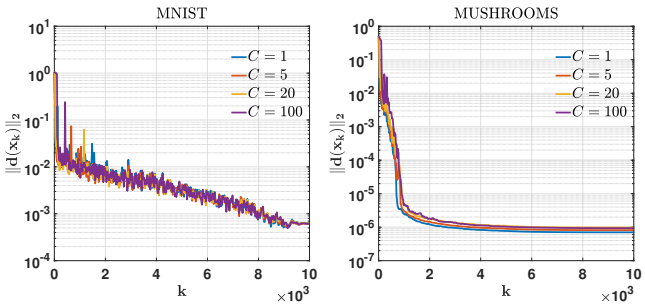

Fig. 2: Average value of $\|d(x_k)\|$ vs. iterations obtained by PSG_LECO with **S2** and exact projection for varying C.

Table 4: Selected parameter values for different step-size strategies across datasets.

Strategy (α/γ_0)	MNIST	MUSHROOMS	DIABETES	DTOC1L	HUESTIS	HS50
S1 (α)	10^{-2}	10	10^{-3}	10^{-3}	1	10^{-2}
S2 (γ_0)	1	10	10^{-2}	10^{-2}	1	10^{-2}
S3 (γ_0)	10	10	1	10^{-3}	10^{-2}	10^{-3}

Figure 1 displays the trend of $\|d(x_k)\|$ through the iterative procedure for the three strategies with selected parameter values in Table 4; the attained minimum value of $\|d(x_k)\|$ is reported in parentheses in the legend. We observe that Algorithm PSG_LECO coupled with the **S3** strategy generally exhibits good progress toward optimality during the initial phase of the execution and shows a behavior comparable to **S2** in the final stage, except on the MUSHROOMS dataset. On the other hand, the **S2** strategy consistently achieves better or comparable optimality measures against **S3** in the middle stage of the execution, except for HUESTIS where some awkward peaks in $\|d(x_k)\|$ occurred, though recovered at the end of the process. Except for MUSHROOMS, the performance of **S2** is overall superior to the performance of **S1** and this fact can be attributed to the incorporation of the diminishing α_{k+1} in **S2**. Summarizing the results obtained, a tuned implementation of Algorithm PSG_LECO combined with the **S2** strategy is effective on our test problems.

We conclude this analysis, showing the impact of the BB step-length update period, denoted by C, of δ_k in (40). The results presented above correspond to $C = 20$. We tested the values $C \in \mathcal{T}_2 = \{1, 5, 20, 100\}$ in the strategy **S2** with γ_0 as in Table 4. Figure 2 illustrates the behavior of the optimality measure versus the iteration count. For the sake of readability, we display one point every 20 iterations. The behavior of the optimality measure is only slightly influenced by the value of C, although $C = 100$ appears to be less effective. On the other hand, the cost in terms of stochastic gradient evaluations depends on C since the smaller C the higher the cost required to evaluate z_{k-1} in (40); thus $C = 20$ represents a good tradeoff between efficiency and cost.

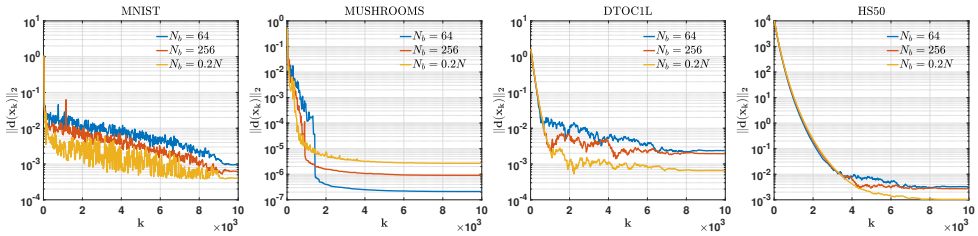

Fig. 3: Average value of $\|d(x_k)\|$ vs. iterations obtained by PSG_LECO with strategy **S2** and exact projection for varying N_b.

6.1.2 Mini-Batch Size Selection

The results presented above correspond to batch sizes, denoted as N_b, equal to 64 for DIABETES and to 256 for the remaining datasets, as stated in §5.2. Now, we investigate the influence of N_b by testing values in the set $\mathcal{T}_3 = \{64, 256, 0.2N\}$. Figure 3 shows the optimality measure versus iterations for PSG_LECO with **S2** and γ_0 from Table 4; values are plotted every 20 iterations for readability.

The initial behavior of the optimality measure is similar for all choices of N_b. The smallest optimality values obtained with $N_b = 64$ and $N_b = 256$ are comparable, except for MUSHROOMS, where $N_b = 64$ is particularly effective. Choosing $N_b = 0.2N$ provides some advantages, except on MUSHROOMS. Overall, with a tuned step-size selection, the choice of N_b is not crucial for the effectiveness and efficiency of PSG_LECO.

6.2 Inexact Projection

We now consider PSG_LECO with the inexact projection $\widetilde{\pi}_S$ in (8). The evaluation of $\widetilde{\pi}_S$ is performed as discussed in §5.2. We tested PSG_LECO Algorithm coupled with the strategy **S2** with γ_0 as in Table 4 over MNIST and MUSHROOM due to their relatively larger values of m. We let $\eta \in \mathcal{T}_4 = \{0.2, 0.5, 0.7, 0.9\}$, $\rho \in \mathcal{T}_5 = \{0.90, 0.95, 0.98, 0.99\}$, $\mu_0 = 0.1$, and studied the potential effect of allowing looser linear solver accuracy in the initial phase of the iterations.

Figure 4 shows that $\|d(x_k)\|$ is generally insensitive to η and ρ. Slightly higher values of $\|d(x_k)\|$ are observed in the first phase of the iterative process for the large values of ρ, but $\mu_k = \mu_0 \rho^k$ is supposed to vanish eventually; thus changes in the values of ρ as well as η become negligible due to the specific form of the upper bound in (9).

Figure 5 illustrates the optimality measure versus execution time (in seconds), given the prefixed number of iterations $k_{\max} = 10^4$. It refers to $\eta_k = \eta = 0.5$, for all $k \geq 0$, and concerns the parameter setting $\rho \in \mathcal{T}_5$ and $\mu_0 \in \mathcal{T}_6 = \{10^{-1}, 1, 10, 100\}$. We note that PSG_LECO is quite insensitive to the choice of μ_0 in ρ, although large values of μ_0 and ρ yield some gain in terms of timings.

Finally, in Figure 6, we show both the optimality measure $\|d(x_k)\|$ and the infeasibility measure $\|Ax_k - b\|$ versus the iterations for PSG_LECO with strategies **S1**–**S3** and parameters as in Table 3; the lowest achieved values are reported in the corresponding legends. The results refer to $\eta_k = \eta = 0.5$, for all $k \geq 0$, $\mu_0 = 10^{-1}$ and $\rho = 0.95$.

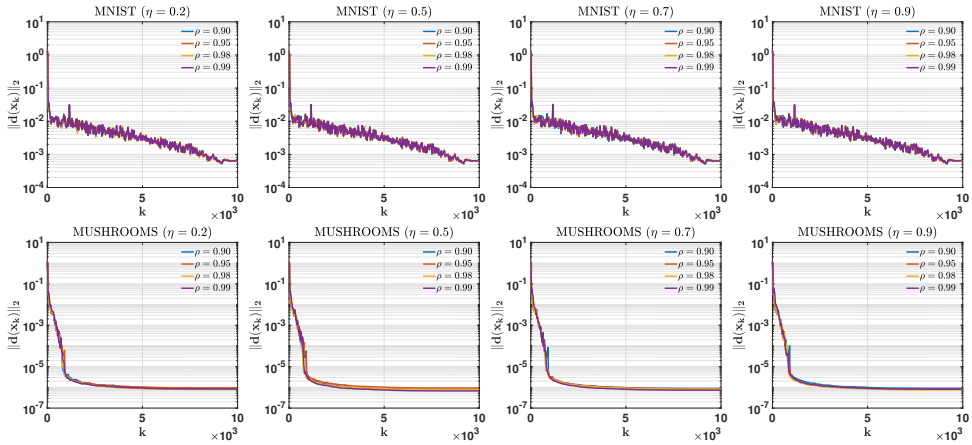

Fig. 4: Average values of $\|d(x_k)\|$ vs. iterations obtained by PSG_LECO with strategy **S2**, inexact projection, $\mu_0 = 0.1$ and varying η, ρ.

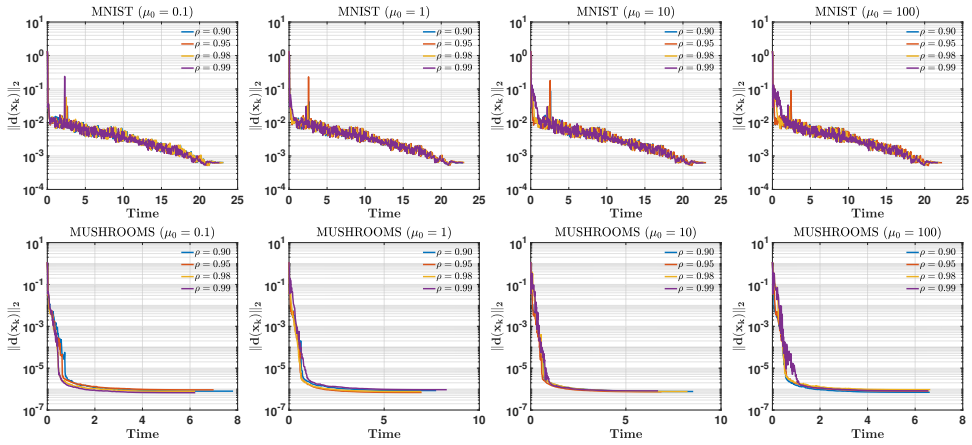

Fig. 5: Average value of $\|d(x_k)\|$ vs. CPU time obtained by PSG_LECO with strategy **S2** and the inexact projection for $\eta = 0.5$ and varying μ_0, ρ.

We note that the infeasibility decreases fast and that the decrease of optimality is not affected by the inexact projection.

6.3 Comparison of PSG_LECO with SQP_ADAM

In this section we present results from the numerical comparison of PSG_LECO with the procedure SQP_ADAM [13]. Both algorithms PSG_LECO and SQP_ADAM are first-order objective function-free procedures, and require a stochastic gradient at each iteration. As for the feasibility, SQP_ADAM was designed with exact projection. It employs a constant step-size $\bar{\alpha}$ and tuning is required. In [13], SQP_ADAM was run with tuned step-sizes of order 10^{-4} to 10^{-2} for a batch size corresponding to $0.2N$.

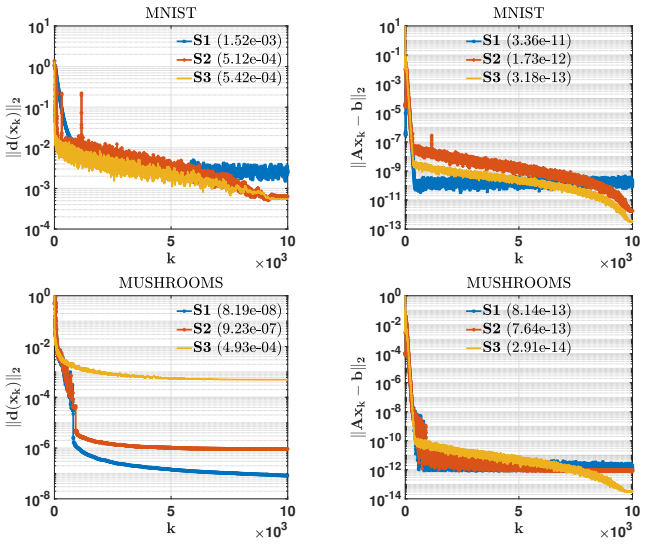

Fig. 6: Average value of $\|d(x_k)\|$ and $\|Ax_k - b\|$ vs. iterations obtained by PSG_LECO with the strategies **S1-S3** and inexact projection.

Table 5: Step-size hyperparameters for PSG_LECO and SQP_ADAM for varying N_b.

Dataset	**Methods**	$N_b = 64$	$N_b = 256$	$N_b = 0.2N$
Mnist	**SQP_ADAM** ($\bar{\alpha}$)	0.001	0.001	0.001
	PSG_LECO (**S2**, γ_0)	1	1	1
Mushrooms	**SQP_ADAM** ($\bar{\alpha}$)	1	1	1
	PSG_LECO (**S2**, γ_0)	1	10	10

We implemented PSG_LECO with exact projection and the strategy **S2** where the update of δ_k in (40) was performed every $C = 20$ iterations with an additional gradient evaluation is required for such an update. Focusing on MNIST and MUSHROOMS problems, PSG_LECO and SQP_ADAM were tested with γ_0 and $\bar{\alpha}$, respectively, from the set $\mathcal{T}_7 = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ and with batch size N_k from $\mathcal{T}_3 = \{64, 256, 0.2N\}$. The best-performing values of $\bar{\alpha}$ and γ_0 are reported in Table 5, and Figure 7 presents a comparison of the average optimality measure versus the iterations obtained with such parameters and exact projection; the value of $\|d(x_k)\|$ is displayed every 20 iterations for readability.

We note that both PSG_LECO and SQP_ADAM are reliable. In most cases, PSG_LECO shows a faster decrease of the optimality measure in the initial phase of the process and is competitive with SQP_ADAM. PSG_LECO achieves lower values of $\|d(x_k)\|$ on MNIST, a significantly lower optimality measure for $N_b = 64$ on MUSHROOMS, and satisfactory optimality measures for the remaining mini-batch sizes ($N_b = 256$ and $0.2N$) on MUSHROOMS though SQP_ADAM is more accurate. This suggests the selection **S2** of the step-size compares well with the scaling and momentum formula in SQP_ADAM.

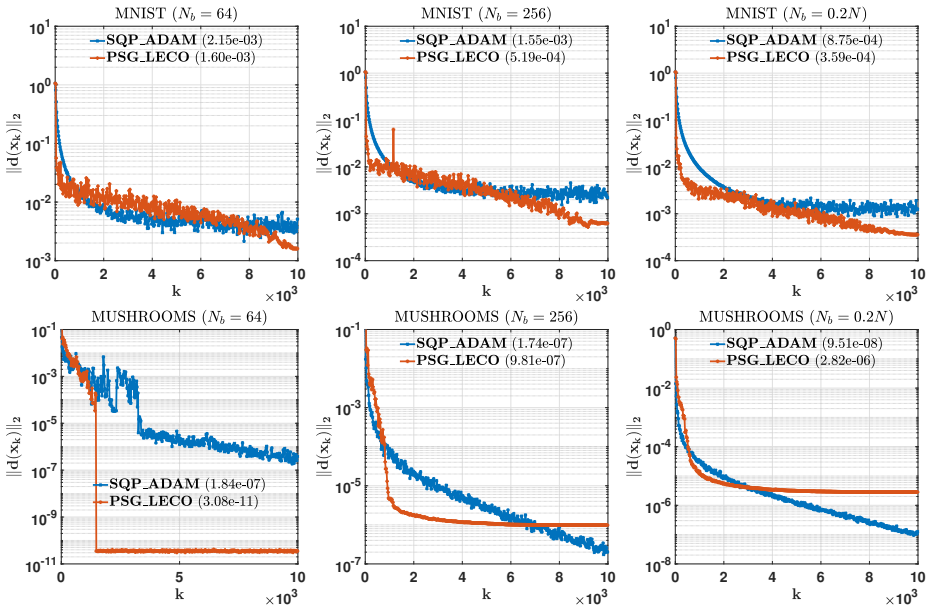

Fig. 7: Average values of $\|d(x_k)\|$ vs. iterations obtained by SQP_ADAM Algorithm and PSG_LECO Algorithm with strategy **S2** and the exact projection.

6.4 Comparison of PSG_LECO with IPAS

We conclude our numerical validation by comparing our method with Algorithm IPAS [17]. Both methods use inexact projections; unlike PSG_LECO, IPAS uses a stochastic line search procedure, and thus function evaluations, that monitors decrease at each iteration and dynamically adjusts the batch size. To balance computational cost and optimality progress, the batch size in IPAS is increased only when the line search test fails. This algorithm was tested using its authors' code with the hyperparameter settings from [17]. PSG_LECO was tested using the strategy **S2** with γ_0 as in Table 4, $\eta = 0.5$, $\mu_0 = 0.1$, $\rho = 0.95$ in (9), and choosing N_b from $\mathcal{T}_3 = \{64, 256, 0.2N\}$. Both methods used the same initial point $x_0 = 0$.

For IPAS, we report in Figure 8 the evolution of the mini-batch size versus iterations, and observe that N_b remains considerably smaller than N. In Figure 9, we display the values of $\|d(x_k)\|$ vs. iterations (top) and CPU time (bottom), plotting one value every 20 iterations for readability. From the first row, obtained with the fixed maximum number of iterations $k_{\max} = 10^4$, we observe that PSG_LECO achieves a significantly faster decrease and lower optimality values compared to IPAS. Analogous conclusions can be drawn from the second row that displays results in term of 165 seconds for MNIST and 30 seconds for MUSHROOMS; the latter results were obtained setting the maximum number of iterations to $k_{\max} = 7 \cdot 10^4$ for MNIST and $k_{\max} = 12 \cdot 10^4$ for MUSHROOM and plotting the results for the previously specified amount of cpu time which the largest time budget common to all the experiments reported.

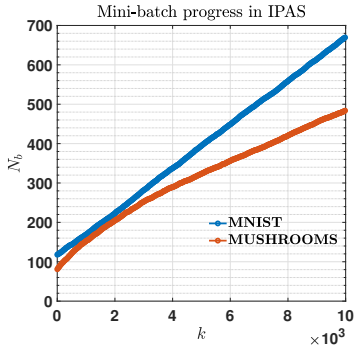

Fig. 8: The progress of adaptive mini-batch in IPAS.

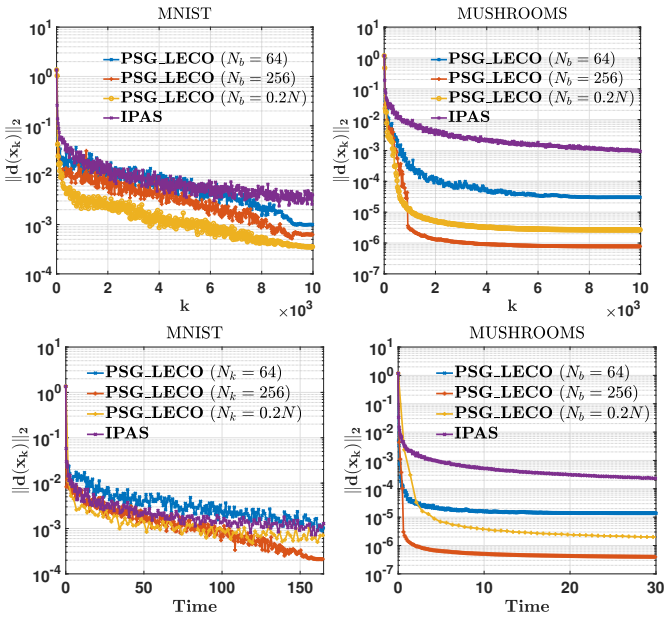

Fig. 9: Average values of $\|d(x_k)\|$ versus iterations (top) and CPU time (bottom) obtained by IPAS and PSG_LECO using strategy **S2**.

The bottom panels show results obtained within a computational budget of 80 seconds for MNIST and 30 seconds for MUSHROOMS. For each N_b, ten seeds may result in different numbers of iterations within the same computational time. To enable a meaningful averaging across seeds, for each run the computational time was recorded

through the iterations. Then, a uniform time grid consisting of 10^4 points was considered and the average value of the optimality at such points was computed using values obtained by linear interpolation. The plots confirm that PSG_LECO compares well with IPAS; namely, the performance of such a method does not suffer from the absence of a test for the acceptance of the iterates.

7 Conclusions

In this work, we proposed a projected stochastic gradient method for minimizing a function subject to deterministic linear constraints. Our method involves a projection map that can be evaluated either exactly or inexactly. Theoretical properties depend on the choice of a step-size related sequence and are equivalent to those established in the unconstrained setting. Numerical illustration of our procedure was presented to show its effectiveness.

Acknowledgments

Natasa Krklec Jerinkić was supported by the Science Fund of the Republic of Serbia, GRANT No 7359, Project LASCADO.
Benedetta Morini and Mahsa Yousefi are members of the INdAM Research Group GNCS. The research that led to the present paper was partially supported by INDAM-GNCS through Progetti di Ricerca 2026.
The authors wish to thank Qi Wang and Yulang Zhu for providing the code for SQP_ADAM, and Luka Rutešić for providing the code for IPAS.

Data availability

The datasets utilized in this research are publicly accessible and commonly employed benchmarks in the field of machine learning and numerical optimization, see [25, 30].

Declarations

Conflict of interest

The authors have no relevant financial or non-financial interests to disclose.

References

[1] Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Review **60**(2), 223–311 (2018)

[2] Tan, C., Ma, S., Dai, Y.-H., Qian, Y.: Barzilai-borwein step size for stochastic gradient descent. Advances in Neural Information Processing Systems **29**, 685–693 (2016)

[3] Krklec Jerinkić, N., Ruggiero, V., Trombini, I.: Spectral stochastic gradient method with additional sampling for finite and infinite sums. Computational Optimization and Applications **91(2)**, 717–758 (2025)

[4] Bellavia, S., Krejić, N., N., K.J., Raydan, M.: Slises: Subsampled line search spectral gradient method for finite sums. Optimization Methods and Software **91(2)**, 1–26 (2024)

[5] Bellavia, S., Morini, B., Yousefi, M.: Fully stochastic trust-region methods with Barzilai-Borwein steplengths. Journal of Computational and Applied Mathematics **476**, 117059 (2026)

[6] Robbins, H., Monro, S.: A stochastic approximation method. SIAM J. Optim **21**, 1109–1140 (1951)

[7] Friedlander, M.P., Schmidt, M.: Hybrid deterministic-stochastic methods for data fitting. SIAM Journal on Scientific Computing **34(3)**, 1380–1405 (2012)

[8] Franchini, G., Porta, F., Ruggiero, V., Trombini, I., Zanni, L.: A stochastic gradient method with variance control and variable learning rate for deep learning. Journal of Computational and Applied Mathematics **451**, 116083 (2024)

[9] Bastin, F., Cirillo, C., Toint, P.L.: An adaptive Monte Carlo algorithm for computing mixed logit estimators. Computational Management Science **3(1)**, 55–79 (2006)

[10] Bellavia, S., Krejić, N., Morini, B., Rebegoldi, S.: A stochastic first-order trust-region method with inexact restoration for finite-sum minimization. Computational Optimization and Applications **84**, 53–84 (2023)

[11] Paquette, C., Scheinberg, K.: A stochastic line search method with expected complexity analysis. SIAM Journal on Optimization **30(1)**, 349–376 (2020)

[12] Curtis, F.E., Scheinberg, K., Shi, R.: A stochastic trust region algorithm based on careful step normalization. Informs Journal on Optimization **1(3)**, 200–220 (2019)

[13] Wang, Q., Piermarini, C., Zhu, Y., Curtis, F.E.: Projected stochastic momentum methods for nonlinear equality-constrained optimization for machine learning. arXiv preprint arXiv:2601.11795 (2026)

[14] Curtis, F., Robinson, D., Zhou, B.: Sequential quadratic optimization for stochastic optimization with deterministic nonlinear inequality and equality constraints. SIAM Journal on Optimization **34**, 3592–3622 (2024)

[15] Berahas, A.S., Curtis, F.E., Robinson, D., Zhou, B.: Sequential quadratic optimization for nonlinear equality constrained stochastic optimization. SIAM

Journal on Optimization **31**(3), 1352–1379 (2021)

[16] Fang, Y., Na, S., Mahoney, M.W., Kolar, M.: Fully stochastic trust-region sequential quadratic programming for equality-constrained optimization problems. SIAM Journal on Optimization **34**(2), 2000–2037 (2024)

[17] Krejić, N., Krklec Jerinkić, N., Rapajić, S., Rutešić, L.: IPAS: An adaptive sample size method for weighted finite sum problems with linear equality constraints. arXiv preprint arXiv:2504.19629 (2025)

[18] Hestenes M., S.E.: Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards **49**, 409–436 (1952)

[19] Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM Journal on Optimization **10**(4), 1196–1211 (2000)

[20] Krejić, N., Krklec Jerinkić, N.: Non-monotone line search methods with variable sample size. Numerical Algorithms **68**, 711–739 (2015)

[21] Robbins, H., Siegmund, D.: A convergence theorem for non negative almost supermartingales and some applications. In: Optimizing Methods in Statistics, pp. 233–257. Elsevier, New York (1971)

[22] Bellavia, S., Gratton, S., Morini, B., Toint, P.L.: Fast stochastic second-order Adagrad for nonconvex bound-constrained optimization. arXiv:2505.06374 (2025)

[23] Krejić, N., Krklec Jerinkić, N., Ostojić, T., Vučićević, N.: AS-BOX: Additional sampling method for weighted sum problems with box constraints. Numerical Algorithms (2025)

[24] Krejić, N., Krklec Jerinkić, N., Ostojić, T., Vučićević, N.: Aspen: An additional sampling penalty method for finite-sum optimization problems with nonlinear equality constraints. arXiv preprint arXiv:2508.02299 (2025)

[25] Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) **2**(3), 1–27 (2011)

[26] Dai, Y.H., Fletcher, R.: Projected Barziali-Borwein methods for large box-constrained quadratic programming. Numer. Math. **100**, 21–47 (2005)

[27] Friedlander, A., Martiínez, J.M., Molina, B., Raydan, M.: Gradient methods with retard and generalizations. SIAM J. Numer. Anal. **36**, 275–289 (1999)

[28] Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)

[29] MathWorks: Cosine learning rate schedule. Deep Learning Toolbox Documentation, available at: https://it.mathworks.com/help/deeplearning/ref/cosinelearnrate.html (2024)

[30] Gratton, S., Toint, P.L.: S2MPJ and CUTEst optimization problems for Matlab, Python and Julia. Optimization Methods and Software, 1–33 (2025)