# LSOS: LINE-SEARCH SECOND-ORDER STOCHASTIC OPTIMIZATION METHODS*

DANIELA DI SERAFINO[†], NATAŠA KREJIĆ[‡], NATAŠA KRKLEC JERINKIĆ[‡], AND MARCO VIOLA[†]

**Abstract.** We develop a line-search second-order algorithmic framework for optimization problems in noisy environments, i.e., assuming that only noisy values are available for the objective function and its gradient and Hessian. In the general noisy case, almost sure convergence of the methods fitting into the framework is proved when line searches and suitably decaying step lengths are combined. When the objective function is a finite sum, such as in machine learning applications, our framework is specialized as a stochastic L-BFGS method with line search only, with almost sure convergence to the solution. In this case, linear convergence rate of the expected function error is also proved, along with a worst-case $\mathcal{O}(\log(\varepsilon^{-1}))$ complexity bound. Numerical experiments, including comparisons with state-of-the art first- and second-order stochastic optimization methods, show the efficiency of our approach.

**Key words.** Stochastic optimization, Newton-type methods, quasi-Newton methods, almost sure convergence, complexity bounds.

**AMS subject classifications.** 90C15, 65K05, 62L20.

## 1. Introduction.

We consider the problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \phi(\mathbf{x}), \tag{1.1}$$

where $\phi(\mathbf{x})$ is a twice continuously differentiable function in a noisy environment. This means that the values of $\phi(\mathbf{x})$, $\nabla\phi(\mathbf{x})$ and $\nabla^2\phi(\mathbf{x})$ are only accessible with some level of noise. There is a large class of problems of this type in many areas of engineering and of physical and social sciences [11, 26, 27, 38]. Typical applications are, e.g., model fitting, parameter estimation, experimental design, and performance evaluation. Furthermore, problem (1.1) is typical in the framework of statistical learning, where very large training sets make computations extremely expensive. In this case, it is common to work with subsamples of data, obtaining approximate function values – see, e.g., [7].

In the last few years there has been increasing interest toward stochastic optimization methods able to use second-order information, with the aim of improving accuracy and efficiency of first-order stochastic methods. We are interested in developing a family of line-search stochastic optimization methods where the search direction is obtained by exploiting (approximate) second-order information.

In order to provide motivations for our work and outline some techniques exploited in the sequel, we provide a quick overview of stochastic optimization methods. Their

*Submitted to the editors DATE.

**Funding:** research supported by the Executive Programme of Scientific and Technological Cooperation between the Italian Republic and the Republic of Serbia for years 2019-21 (Italian grant no. RS19MO05, Serbian grant no. 174030). D. di Serafino and M. Viola were also supported by the Istituto Nazionale di Alta Matematica - Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS) and by the V:ALERE Program of the University of Campania "L. Vanvitelli" (VAIN-HOPES Project). N. Krejić and N. Krklec Jerinkić were also supported by a Ministry of Education, Science and Technological Development grant.

[†]Department of Mathematics and Physics, University of Campania "Luigi Vanvitelli", viale A. Lincoln 5, 81100 Caserta, Italy (daniela.diserafino@unicampania.it, marco.viola@unicampania.it).

[‡]Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia (natasak@uns.ac.rs, natasa.krklec@dmi.uns.ac.rs).

roots can be found in the Stochastic Approximation (SA) method by Robbins and Monro [31], which can be interpreted as a stochastic variant of the gradient descent method. Convergence in probability of the SA method is ensured if the step-length sequence $\{\alpha_k\}$ (called gain sequence) is of harmonic type, i.e., it is non-summable but square-summable. Under suitable assumptions, the method converges in the mean square sense [31] and almost surely [37]. Its key property is the ability to avoid zigzagging due to noise when approaching the solution, thanks to the decay of the gain sequence. However, a significant drawback of the SA method is its slow convergence.

Asymptotically ideal step lengths for SA methods include the norm of the inverse Hessian at the solution [3]. Adjustments to the classical harmonic gain sequence, including adaptive step lengths based on changes in the sign of the difference between consecutive iterates, are analyzed in [13, 21] with the aim of speeding up the SA method. This idea is further developed in [39], where almost sure convergence of the method is proved. Adaptive step-length schemes are introduced also in [40], with the objective of reducing the dependence of the behavior of the method on user-defined parameters. The results in [5] are closely related to the SA method and concern methods based on search directions that are not necessarily noisy gradients, but some gradient-related directions. A hybrid approach that combines a line-search technique with SA is analyzed in [25] for noisy gradient directions and arbitrary descent directions. General descent directions are also considered in [24]. We also note that gradient approximations may need to be computed by using finite differences; an overview of finite-difference methods for stochastic optimization is given in [16]. Variance-reduction SA methods with line search for stochastic variational inequalities are considered in [19].

In the realm of machine learning, many stochastic versions of the gradient method have been developed. Starting from the basic stochastic and minibatch gradient methods – see, e.g., [7] and the references therein – variance reduction techniques for the gradient estimates have been developed, with the aim of improving convergence. Among them we mention SVRG [20], SAGA [12] and its version using Jacobian sketching [18], which will be considered in section 5. These methods have constant step lengths and get linear convergence in expectation.

Stochastic optimization methods exploiting search directions based on second-order information have been developed to get better theoretical and practical convergence properties, especially when badly-scaled problems are considered. Stochastic versions of Newton-type methods are discussed in [2, 6, 8, 9, 33, 34, 35, 36] and a variant of the adaptive cubic regularization scheme using a dynamic rule for building inexact Hessian information is proposed in [1]. Stochastic BFGS methods are analyzed, e.g., in [8, 10, 17, 28, 29, 30]. In particular, in [30] Moritz et al. propose a stochastic L-BFGS algorithm based on the same inverse Hessian approximation as in [10], but use SVRG instead of the standard stochastic gradient approximation. This algorithm, which applies a constant step length, has Q-linear rate of convergence of the expected value of the error in the objective function. A further modification to this L-BFGS scheme is proposed by Gower et al. in [17], where a stochastic block BFGS update is used, in which the vector pairs for updating the inverse Hessian are replaced by matrix pairs gathering directions and matrix-vector products between subsampled Hessians and those directions. The resulting algorithm uses constant step length and has Q-linear convergence rate of the expected value of objective function error, as in the previous case, but appears more efficient by numerical experiments.

**Our contribution.** We propose a Line-search Second-Order Stochastic (LSOS) algorithmic framework for stochastic optimization problems, where Newton and quasi-Newton directions in a rather broad meaning are used. Inexactness is allowed in the sense that the (approximate) Newton direction can be obtained as inexact solution of the corresponding system of linear equations. We focus on convex problems as they appear in a wide variety of applications, such as machine learning and least squares. Furthermore, many stochastic problems need regularization and hence become convex.

We prove almost sure convergence of the methods fitting into the LSOS framework and show by experiments the effectiveness of our approach when using Newton and inexact Newton directions affected by noise.

For finite-sum objective functions such as those arising in machine learning, we investigate the use of the stochastic L-BFGS Hessian approximations in [10] together with line searches and the SAGA variance reduction technique for the gradient estimates. The resulting algorithm has almost sure convergence to the solution, while for the efficient state-of-the-art stochastic L-BFGS methods in [17, 30] it has been proved only that the function error tends to zero in expectation. We also prove that the expected function error has linear convergence rate and provide a worst-case $\mathcal{O}(\log(\varepsilon^{-1}))$ complexity bound. Finally, numerical experiments show that our algorithm is competitive with the stochastic L-BFGS methods mentioned above.

**Notation.** $\mathbb{E}(x)$ denotes the expectation of a random variable $x$, $\mathbb{E}(x|y)$ the conditional expectation of $x$ given $y$, and $\mathrm{var}(x)$ the variance of $x$. $\|\cdot\|$ indicates either the Euclidean vector norm or the corresponding induced matrix norm, while $|\cdot|$ is the cardinality of a set. $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the sets of real non-negative and positive numbers, respectively. Vectors are written in boldface and subscripts indicate the elements of a sequence, e.g., $\{\mathbf{x}_k\}$. Throughout the paper $M_1, M_2, M_3, \ldots$ and $c_1, c_2, c_3, \ldots$ denote positive constants, without specifying their actual values. Other constants are defined when they are used. Finally, "a.s." abbreviates "almost sure/surely".

**Outline of the paper.** The rest of this article is organized as follows. In section 2, we define the general Stochastic Second-Order (SOS) framework with pre-defined step-length sequence, which is the basis for the family of algorithms proposed in this work, and we give preliminary assumptions and results used in the sequel. In section 3 we provide the convergence theory of the algorithms fitting into the SOS framework. In section 4 we introduce a SOS version named LSOS, which combines non-monotone line searches and (if needed) pre-defined step lengths in order to make the algorithm faster, and provide its convergence analysis. In section 5 we specialize LSOS for finite sum objective functions, obtaining a stochastic L-BFGS method with line search only, and in section 6 we provide its convergence theory, including convergence rate and complexity results. In section 7, numerical experiments on two classes of stochastic problems and comparisons with state-of-the art methods show the effectiveness of our approach. Concluding remarks are given in section 8.

## 2. Preliminaries.

We assume that for problem (1.1) we can only compute

$$
\begin{array}{rcl}
f(\mathbf{x}) & = & \phi(\mathbf{x}) + \varepsilon_f(\mathbf{x}), \\
\mathbf{g}(\mathbf{x}) & = & \nabla\phi(\mathbf{x}) + \boldsymbol{\varepsilon}_g(\mathbf{x}), \\
B(\mathbf{x}) & = & \nabla^2\phi(\mathbf{x}) + \varepsilon_B(\mathbf{x}),
\end{array} \tag{2.1}
$$

with $\varepsilon_f(\mathbf{x})$ being a random number, $\boldsymbol{\varepsilon}_g(\mathbf{x})$ a random vector and $\varepsilon_B(\mathbf{x})$ a symmetric random matrix. The general algorithmic scheme we analyze in this paper is given in

Algorithm 2.1.

**Algorithm 2.1** Second-Order Stochastic (SOS) method

1: given $\mathbf{x}^0 \in \mathbb{R}^n$ and $\{\alpha_k\} \subset \mathbb{R}_+$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: compute $\mathbf{d}_k \in \mathbb{R}^n$
4: set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
5: **end for**

For now we assume that $\{\alpha_k\}$ is given and it satisfies the conditions stated in Assumption 2.2 below. We also assume that $f(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$ and $B(\mathbf{x})$ are available for any $\mathbf{x} \in \mathbb{R}^n$. Although here we do not specify how $\mathbf{d}_k$ is obtained, we call the algorithm "Second-Order" because in the next sections we will compute $\mathbf{d}_k$ by exploiting noisy second-order information about $\phi(\mathbf{x})$.

We make the following assumptions.

Assumption 2.1. *The function $\phi$ is strongly convex and has Lipschitz-continuous gradient.*

If Assumption 2.1 holds, then there exists a unique $\mathbf{x}_* \in \mathbb{R}^n$ that solves (1.1), with $\nabla\phi(\mathbf{x}_*) = \mathbf{0}$. Furthermore, for some positive constants $\mu$ and $L$ and any $\mathbf{x} \in \mathbb{R}^n$ we have

$$\mu I \preceq \nabla^2 \phi(\mathbf{x}) \preceq LI,$$

where $I$ is the identity matrix, and

$$\frac{\mu}{2}\|\mathbf{x} - \mathbf{x}_*\|^2 \leq \phi(\mathbf{x}) - \phi(\mathbf{x}_*) \leq \frac{L}{2}\|\nabla\phi(\mathbf{x})\|^2. \tag{2.2}$$

Assumption 2.2. *The gain sequence $\{\alpha_k\}$ satisfies*

$$\alpha_k > 0 \text{ for all } k, \quad \sum_k \alpha_k = \infty, \quad \sum_k \alpha_k^2 < \infty.$$

This is a standard assumption for SA methods.

Henceforth we denote $\mathcal{F}_k$ the $\sigma$-algebra generated by $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k$.

Assumption 2.3. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 2.1. The gradient noise $\boldsymbol{\varepsilon}_g(\mathbf{x})$ is such that*

$$\mathbb{E}(\boldsymbol{\varepsilon}_g(\mathbf{x})|\mathcal{F}_k) = 0 \text{ and } \mathbb{E}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|^2|\mathcal{F}_k) \leq M_1.$$

In other words, we assume that the expected gradient noise is zero and the variance of gradient errors,

$$\operatorname{var}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|\,|\mathcal{F}_k) = \mathbb{E}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|^2\,|\mathcal{F}_k) - \mathbb{E}^2(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|\,|\mathcal{F}_k), \tag{2.3}$$

is bounded. From (2.3) and Assumption 2.3 it also follows that

$$\mathbb{E}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|^2|\mathcal{F}_k) \geq \mathbb{E}^2(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\||\mathcal{F}_k)$$

and hence

$$\mathbb{E}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|\,|\mathcal{F}_k) \leq \sqrt{\mathbb{E}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|^2|\mathcal{F}_k)} \leq \sqrt{M_1} := M_2.$$

We observe that Assumptions 2.1 and 2.3 imply

$$\|\nabla\phi(\mathbf{x})\|^2 + \mathbb{E}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|^2|\mathcal{F}_k) \le L^2\|\mathbf{x}-\mathbf{x}_*\|^2 + M_1 \le c_1(1+\|\mathbf{x}-\mathbf{x}_*\|^2), \tag{2.4}$$

with $c_1 = \max\{L^2, M_1\}$. Moreover, (2.4) and Assumption 2.3 imply

$$\mathbb{E}(\|\mathbf{g}(\mathbf{x})\|^2|\mathcal{F}_k) \le c_1(1+\|\mathbf{x}-\mathbf{x}_*\|^2), \tag{2.5}$$

which can be proved as follows:

$$\begin{aligned}
\mathbb{E}(\|\mathbf{g}(\mathbf{x})\|^2|\mathcal{F}_k) &= \mathbb{E}(\|\nabla\phi(\mathbf{x}) + \boldsymbol{\varepsilon}_g(\mathbf{x})\|^2|\mathcal{F}_k) \\
&= \mathbb{E}(\|\nabla\phi(\mathbf{x})\|^2 + 2\nabla\phi(\mathbf{x})^\top\boldsymbol{\varepsilon}_g(\mathbf{x}) + \|\boldsymbol{\varepsilon}_g(\mathbf{x})\|^2|\mathcal{F}_k) \\
&= \|\nabla\phi(\mathbf{x})\|^2 + 2\nabla\phi(\mathbf{x})^\top\mathbb{E}(\boldsymbol{\varepsilon}_g(\mathbf{x})|\mathcal{F}_k) + \mathbb{E}(\|\boldsymbol{\varepsilon}_g(\mathbf{x})\|^2|\mathcal{F}_k) \\
&\le c_1(1+\|\mathbf{x}-\mathbf{x}_*\|^2),
\end{aligned}$$

where the last inequality comes from $\mathbb{E}(\boldsymbol{\varepsilon}_g(\mathbf{x})|\mathcal{F}_k) = 0$.

The following theorem (see [31]) will be used in Section 3.

THEOREM 2.4. *Let $U_k, \beta_k, \xi_k, \rho_k \ge 0$ be $\mathcal{F}_k$-measurable random variables such that*

$$\mathbb{E}(U_{k+1}|\mathcal{F}_k) \le (1+\beta_k)U_k + \xi_k - \rho_k, \quad k = 1, 2, \ldots.$$

*If $\sum_k \beta_k < \infty$ and $\sum_k \xi_k < \infty$, then $U_k \to U$ a.s. and $\sum_k \rho_k < \infty$ a.s..*

**3. Convergence theory of Algorithm SOS.** The assumptions stated in the previous section generally form a common set of assumptions for SA and related methods. Actually, Assumption 2.1 is different from the commonly used assumption that for some symmetric positive definite matrix $B$ and for all $\eta \in (0,1)$, we have

$$\inf_{\eta < \|\mathbf{x}-\mathbf{x}_*\| < \frac{1}{\eta}} (\mathbf{x}-\mathbf{x}_*)^\top B \nabla\phi(\mathbf{x}) > 0.$$

However, the restriction to strongly convex problems allows us to prove a more general convergence result.

While the SA method uses the negative gradient direction, in [24] general descent directions have been considered such that for all $k$

$$\mathbf{g}(\mathbf{x}_k)^\top \mathbf{d}_k < 0, \tag{3.1}$$

$$\begin{aligned}
(\mathbf{x}_k - \mathbf{x}_*)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) &\le -c_3\|\mathbf{x}_k - \mathbf{x}_*\| \ \text{ a.s.}, \\
\|\mathbf{d}_k\| &\le c_4\|\mathbf{g}(\mathbf{x}_k)\| \ \text{ a.s.}.
\end{aligned} \tag{3.2}$$

Here we relax (3.1) and (3.2) so that the direction $\mathbf{d}_k$ need neither be a descent direction nor satisfy (3.2). This relaxation allows us to extend the set of directions covered by the theoretical analysis presenter further on. At each iteration, we allow a deviation from a descent direction proportional to $\delta_k$, where $\{\delta_k\}$ is a predefined sequence of positive numbers that converges to zero with almost arbitrary rate. More precisely, the following condition must hold:

$$\sum_k \alpha_k \delta_k < \infty. \tag{3.3}$$

Thus, a possible choice could be $\delta_k = \nu^k$, where $\nu \in (0,1)$, regardless of the choice of the gain sequence. On the other hand, if we choose the standard gain sequence

$\alpha_k = 1/k$, then $\delta_k = 1/k^\epsilon$, with arbitrary small $\epsilon > 0$, is a suitable choice. Roughly speaking, the set of feasible directions is rather wide while we are far away from the solution, and the descent condition is enforced as we progress towards the solution. More precisely, we make the following assumptions on the search directions.

ASSUMPTION 3.1. *The direction* $\mathbf{d}_k$ *satisfies*

$$\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \le \delta_k c_2 - c_3\|\nabla\phi(\mathbf{x}_k)\|^2.$$

ASSUMPTION 3.2. *The direction* $\mathbf{d}_k$ *satisfies*

$$\|\mathbf{d}_k\| \le c_4\|\mathbf{g}(\mathbf{x}_k)\| \ \ a.s..$$

We observe that Assumptions 3.1 and 3.2 can be seen as a stochastic version of well-known sufficient conditions that guarantee gradient-related directions in the deterministic setting [4, p. 36], i.e.,

$$\nabla\phi(\mathbf{x}_k)^\top \mathbf{d}_k \le -q_1 \|\nabla\phi(\mathbf{x}_k)\|^{p_1}, \qquad \|\mathbf{d}_k\| \le q_2 \|\nabla\phi(\mathbf{x}_k)\|^{p_2}$$

for $q_1, q_2 > 0$ and $p_1, p_2 \ge 0$.

In the following theorem we prove almost sure convergence for the general Algorithm SOS.

THEOREM 3.3. *Let Assumptions 2.1 to 2.3 and Assumptions 3.1 and 3.2 hold, and let* $\{\mathbf{x}_k\}$ *be generated by Algorithm 2.1. Assume also that (3.3) holds. Then* $\mathbf{x}_k \to \mathbf{x}_*$ *a.s..*

*Proof.* Since $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ we have, by Assumption 2.1 and the descent lemma [4, Proposition A24],

$$\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*) \le \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) + \alpha_k \nabla\phi(\mathbf{x}_k)^\top \mathbf{d}_k + \frac{L}{2}\alpha_k^2\|\mathbf{d}_k\|^2.$$

Therefore, by Assumption 3.2,

$$\begin{aligned}
\mathbb{E}(\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*)\,|\mathcal{F}_k) &\le \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) + \alpha_k \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \\
&\quad + \frac{L}{2}\alpha_k^2\, c_4^2\, \mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k) \\
&= \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) + \alpha_k \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \\
&\quad + \alpha_k^2\, c_5\, \mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k)
\end{aligned}$$

where $c_5 = Lc_4^2/2$. From (2.5) (arising from Assumptions 2.1 and 2.3) and Assumption 3.1 it follows that

$$\begin{aligned}
\mathbb{E}(\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*)|\mathcal{F}_k) &\le \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) + \alpha_k^2 c_1 c_5 \left(1 + \|\mathbf{x}_k - \mathbf{x}_*\|^2\right) \\
&\quad + \alpha_k \left(\delta_k c_2 - c_3\|\nabla\phi(\mathbf{x}_k)\|^2\right).
\end{aligned}$$

Since (2.2) holds, we have

$$\begin{aligned}
\mathbb{E}(\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*)|\mathcal{F}_k) &\le (1 + \alpha_k^2 c_6)(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) + \alpha_k^2 c_1 c_5 \\
&\quad + \alpha_k \delta_k c_2 - \alpha_k c_3 \frac{2}{L}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)),
\end{aligned}$$

with $c_6 = 2c_1c_5/\mu$. Taking $\beta_k = \alpha_k^2 c_6$, $U_k = \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)$, $\xi_k = \alpha_k^2 c_1 c_5 + \alpha_k \delta_k c_2$ and $\rho_k = 2\alpha_k c_3/L\,(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*))$, we have

$$\sum_k \beta_k < \infty, \quad \sum_k \xi_k < \infty$$

because of Assumption 2.2 and (3.3), and $U_k \geq 0$ as $\mathbf{x}_*$ is the solution of (1.1). Therefore, by Theorem 2.4 we conclude that $\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)$ converges a.s. and $\sum_k \rho_k < \infty$ a.s.. Hence, we have

$$0 = \lim_{k\to\infty} \rho_k = \lim_{k\to\infty} \alpha_k c_3 \frac{2}{L} \left(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)\right) \quad a.s..$$

There are two possibilities for the sequence $\{\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)\}$: either there exists an infinite set $\mathcal{K} \subset \mathbb{N}$ such that

$$\lim_{k\in\mathcal{K}, k\to\infty} \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) = 0 \text{ a.s.}$$

or there exists $\varepsilon > 0$ such that

$$\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) \geq \varepsilon \text{ a.s. for all } k \text{ sufficiently large.} \tag{3.4}$$

If $\mathcal{K}$ exists, then we have that the whole sequence $\{\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)\}$ converges to zero a.s., and then $\mathbf{x}_k \to \mathbf{x}_*$ a.s. because of the continuity of $\phi$. On the other hand, if (3.4) holds, then

$$\sum_k \rho_k = \sum_k \alpha_k c_3 \frac{2}{L} (\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) \geq c_3 \frac{2}{L}\, \varepsilon \sum_k \alpha_k = \infty \text{ a.s.,}$$

which is a contraction. Thus we conclude that $\mathbf{x}_k \to \mathbf{x}_*$ a.s.. ◻

Now we extend the scope of search directions towards second-order approximations. Since Assumption 2.1 holds, we also assume that the approximate Hessians are positive definite and bounded.

ASSUMPTION 3.4. *For every approximate Hessian $B(\mathbf{x})$,*

$$\mu I \preceq B(\mathbf{x}) \preceq LI.$$

This assumption is fulfilled in many significant cases. For example, in binary classification, mini-batch subsampled Hessians are taken as positive definite and bounded matrices, either with a proper choice of the subsample [32], or with regularization [8]. The same is true for least squares problems.

Assumption 3.4 implies

$$\frac{1}{L} I \preceq B^{-1}(\mathbf{x}) \preceq \frac{1}{\mu} I,$$

and hence $\|B^{-1}(\mathbf{x})\| \leq \mu^{-1}$.

We also assume that the noise terms $\varepsilon_f(\mathbf{x}), \boldsymbol{\varepsilon}_g(\mathbf{x})$ and $\varepsilon_B(\mathbf{x})$ are mutually independent, which implies that the same is true for $f$, $\mathbf{g}$ and $B$. This independence assumption will be relaxed in section 5 in order to cope with finite-sum problems, where the gradient and Hessian approximations may be taken from the same sample. By defining

$$\mathbf{d}_k = -D_k \mathbf{g}(\mathbf{x}_k), \quad D_k = B^{-1}(\mathbf{x}_k), \tag{3.5}$$

we have

$$\|\mathbf{d}_k\| \leq \frac{1}{\mu} \|\mathbf{g}(\mathbf{x}_k)\|,$$

thus Assumption 3.2 holds. Furthermore, since $D_k$ is independent of $\mathbf{g}(\mathbf{x}_k)$, we obtain

$$\begin{aligned}
\mathbb{E}(\nabla\phi(\mathbf{x}_k)^\top \mathbf{d}_k|\mathcal{F}_k) &= \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(-D_k\mathbf{g}(\mathbf{x}_k)|\mathcal{F}_k)\\
&= \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(-D_k|\mathcal{F}_k)\mathbb{E}(\mathbf{g}(\mathbf{x}_k)|\mathcal{F}_k)\\
&= \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(-D_k|\mathcal{F}_k)\nabla\phi(\mathbf{x}_k)\\
&= \mathbb{E}(-\nabla\phi(\mathbf{x}_k)^\top D_k\nabla\phi(\mathbf{x}_k)|\mathcal{F}_k)\\
&\le \mathbb{E}(-\frac{1}{L}\|\nabla\phi(\mathbf{x}_k)\|^2|\mathcal{F}_k) = -\frac{1}{L}\|\nabla\phi(\mathbf{x}_k)\|^2
\end{aligned}$$

and hence

$$\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \le -\frac{1}{L}\left\|\nabla\phi(\mathbf{x}_k)\right\|^2. \tag{3.6}$$

Then Assumption 3.1 holds with $c_2 = 0$ and $c_3 = \frac{1}{L}$.

COROLLARY 3.5. *Let Assumptions 2.1 to 2.3 and Assumption 3.4 hold, and let (3.3) hold. If $\{\mathbf{x}_k\}$ is a sequence generated by Algorithm 2.1 with $\mathbf{d}_k$ defined in (3.5), then $\mathbf{x}_k \to \mathbf{x}_*$ a.s.*

*Proof.* The proof is an immediate consequence of Theorem 3.3 and the previous observations. ∎

Finally, let us consider the case of inexact Newton methods in the stochastic approximation framework, i.e., when the linear system

$$B(\mathbf{x}_k)\mathbf{d}_k = -\mathbf{g}(\mathbf{x}_k) \tag{3.7}$$

is solved only approximately, i.e.,

$$\|B(\mathbf{x}_k)\mathbf{d}_k + \mathbf{g}(\mathbf{x}_k)\| \le \delta_k\gamma_k, \tag{3.8}$$

where $\gamma_k$ is a random variable, and $\delta_k$ satisfies (3.3).

For deterministic inexact Newton methods, global convergence has been proved when $\{\mathbf{x}_k\}$ is bounded and the forcing terms are small enough – see the alternative statement of Theorem 3.4 in [15, page 400]. Thus, we will assume $\{\mathbf{x}_k\}$ bounded in the stochastic case as well. For $\gamma_k$ we assume bounded variance as follows.

ASSUMPTION 3.6. *The sequence of random variables $\{\gamma_k\}$ is such that*

$$\mathbb{E}(\gamma_k^2|\mathcal{F}_k) \le M_3.$$

Note that Assumption 3.6 implies

$$\mathbb{E}(\gamma_k|\mathcal{F}_k) \le \sqrt{\mathbb{E}(\gamma_k^2|\mathcal{F}_k)} \le \sqrt{M_3} := M_4.$$

The main property of the search direction that allows us to prove Theorem 3.3 is stated in Assumption 3.1. Now we prove that Assumption 3.1 holds if the sequence $\{\mathbf{x}_k\}$ is bounded and Assumption 3.6 holds.

LEMMA 3.7. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 2.1 such that (3.8) and Assumption 3.6 hold. If $\{\mathbf{x}_k\}$ is bounded, then Assumption 3.1 holds.*

*Proof.* If $\{\mathbf{x}_k\}$ is bounded then $\|\nabla\phi(\mathbf{x}_k)\| \le M_5$ as $\phi$ is continuously differentiable. Furthermore, Assumption 3.6 implies

$$\|\nabla\phi(\mathbf{x}_k)\|\,\mathbb{E}(\gamma_k|\mathcal{F}_k) \le M_5M_4 := M_6. \tag{3.9}$$

Let us denote $\mathbf{r}_k = B(\mathbf{x}_k)\mathbf{d}_k + \mathbf{g}(\mathbf{x}_k)$. Then, by (3.8), $\|\mathbf{r}_k\| \le \delta_k \gamma_k$. Furthermore,

$$\mathbf{d}_k = B(\mathbf{x}_k)^{-1}\mathbf{r}_k - B(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k).$$

Setting $\mathbf{d}_k^N = -B(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k)$, we have

$$\mathbf{d}_k - \mathbf{d}_k^N = B(\mathbf{x}_k)^{-1}\mathbf{r}_k$$

and

$$\nabla\phi(\mathbf{x}_k)^\top \mathbf{d}_k = \nabla\phi(\mathbf{x}_k)^\top \left(\mathbf{d}_k - \mathbf{d}_k^N + \mathbf{d}_k^N\right) = \nabla\phi(\mathbf{x}_k)^\top \mathbf{d}_k^N + \nabla\phi(\mathbf{x}_k)^\top \left(\mathbf{d}_k - \mathbf{d}_k^N\right).$$

Taking the conditional expectation, we get

$$\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) = \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k^N|\mathcal{F}_k) + \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k - \mathbf{d}_k^N|\mathcal{F}_k).$$

It has been shown, see (3.6), that

$$\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k^N|\mathcal{F}_k) \le -\frac{1}{L}\|\nabla\phi(\mathbf{x}_k)\|^2,$$

thus

$$\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \le -\frac{1}{L}\|\nabla\phi(\mathbf{x}_k)\|^2 + \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(B(\mathbf{x}_k)^{-1}\mathbf{r}_k|\mathcal{F}_k). \tag{3.10}$$

Furthermore,

$$\begin{aligned}\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(B(\mathbf{x}_k)^{-1}\mathbf{r}_k|\mathcal{F}_k) &\le \|\nabla\phi(\mathbf{x}_k)\|\ \mathbb{E}\left(\|B(\mathbf{x}_k)^{-1}\|\ \|\mathbf{r}_k\|\ \middle|\mathcal{F}_k\right)\\ &\le \frac{1}{\mu}\|\nabla\phi(\mathbf{x}_k)\|\delta_k\mathbb{E}(\gamma_k|\mathcal{F}_k) \le \frac{1}{\mu}\delta_k M_6 \qquad (3.11)\end{aligned}$$

because of (3.8) and (3.9). Putting together (3.10) and (3.11), we get

$$\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \le \delta_k c_2 - c_3\|\nabla\phi(\mathbf{x}_k)\|^2$$

with $c_2 = M_6/\mu$ and $c_3 = 1/L$. Therefore, Assumption 3.1 holds. ∎

Notice that Assumption 3.2 is not necessarily satisfied by the direction $\mathbf{d}_k$ in (3.8). Therefore, we cannot apply Theorem 3.3. Nevertheless, we can prove the following.

THEOREM 3.8. *Let Assumptions 2.1 to 2.3 and Assumptions 3.4 and 3.6 hold. Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 2.1 with search direction $\mathbf{d}_k$ satisfying (3.8) with $\delta_k$ such that (3.3) holds. If $\{\mathbf{x}_k\}$ is bounded, then $\mathbf{x}_k \to \mathbf{x}_*$ a.s..*

*Proof.* The direction $\mathbf{d}_k$ satisfies

$$\|\mathbf{d}_k\| = \|B(\mathbf{x}_k)^{-1}(\mathbf{r}_k - \mathbf{g}(\mathbf{x}_k))\| \le \frac{1}{\mu}(\|\mathbf{r}_k\| + \|\mathbf{g}(\mathbf{x}_k)\|) \le \frac{1}{\mu}(\delta_k\gamma_k + \|\mathbf{g}(\mathbf{x}_k)\|),$$

thanks to (3.8). Therefore,

$$\|\mathbf{d}_k\|^2 \le \frac{2}{\mu^2}\left(\delta_k^2\gamma_k^2 + \|\mathbf{g}(\mathbf{x}_k)\|^2\right)$$

and

$$\begin{aligned}\mathbb{E}(\|\mathbf{d}_k\|^2|\mathcal{F}_k) &\le \frac{2}{\mu^2}\left(\delta_k^2\mathbb{E}(\gamma_k^2|\mathcal{F}_k) + \mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k)\right)\\ &\le \frac{2}{\mu^2}\left(\delta_k^2 M_3 + c_1(1 + \|\mathbf{x}_k - \mathbf{x}_*\|^2)\right),\end{aligned}$$

because of (2.5) and Assumption 3.6. Therefore,

$$\mathbb{E}(\|\mathbf{d}_k\|^2|\mathcal{F}_k) \leq c_7 + c_8\|\mathbf{x}_k - \mathbf{x}_*\|^2, \tag{3.12}$$

for $c_7 = (2/\mu^2)(\delta_{\max}^2 M_3 + c_1)$ and $c_8 = 2c_1/\mu^2$, where $\delta_{\max} = \max_k \delta_k$. Using the descent lemma and Assumption 2.1 as in Theorem 3.3, we get

$$\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*) \leq \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) + \alpha_k \nabla\phi(\mathbf{x}_k)^\top \mathbf{d}_k + \frac{L}{2}\alpha_k^2\|\mathbf{d}_k\|^2$$

and, by (3.12) and Lemma 3.7,

$$\begin{aligned}
\mathbb{E}(\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*)|\mathcal{F}_k) \leq{}& \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) + \alpha_k \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \\
&+\frac{L}{2}\alpha_k^2\,\mathbb{E}(\|\mathbf{d}_k\|^2|\mathcal{F}_k) \\
\leq{}& \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) + \alpha_k\left(\delta_k\frac{M_6}{\mu} - \frac{1}{L}\|\nabla\phi(\mathbf{x}_k)\|^2\right) \\
&+\alpha_k^2\frac{L}{2}(c_7 + c_8\|\mathbf{x}_k - \mathbf{x}_*\|^2).
\end{aligned}$$

Using (2.2), we get

$$\begin{aligned}
\mathbb{E}(\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*)|\mathcal{F}_k) \leq{}& (\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*))\left(1 + \alpha_k^2\frac{L}{\mu}c_8\right) + \alpha_k^2\frac{L}{2}c_7 \\
&+\alpha_k\delta_k\frac{M_6}{\mu} - \alpha_k\frac{2}{L^2}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)).
\end{aligned}$$

Now we define

$$\beta_k = \alpha_k^2\frac{L}{\mu}c_8, \quad \xi_k = \alpha_k^2\frac{L}{2}c_7 + \alpha_k\delta_k\frac{M_6}{\mu}, \quad \rho_k = \alpha_k\frac{2}{L^2}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*))$$

and

$$U_k = \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*).$$

As the hypotheses of Theorem 2.4 are fulfilled due to (3.3) and Assumption 2.2, we have that $\mathbf{x}_k \to \mathbf{x}_*$ a.s.. ∎

So far we have required only that $\gamma_k$ is a random variable with bounded variance. Following inexact Newton methods in the deterministic case [15], the norm of the left-hand side in (3.7) can be used to define the inexactness in the solution of the linear system. By setting

$$\gamma_k = c_9\,\|\mathbf{g}(\mathbf{x}_k)\|,$$

we have that (3.8) implies

$$\begin{aligned}
\|\mathbf{d}_k\| = \|B(\mathbf{x}_k)^{-1}(\mathbf{r}_k - \mathbf{g}(\mathbf{x}_k))\| &\leq \frac{1}{\mu}(\|\mathbf{r}_k\| + \|\mathbf{g}(\mathbf{x}_k)\|) \\
&\leq \frac{1}{\mu}(\delta_k c_9\,\|\mathbf{g}(\mathbf{x}_k)\| + \|\mathbf{g}(\mathbf{x}_k)\|) = \frac{1}{\mu}(\delta_{\max}c_9 + 1)\|\mathbf{g}(\mathbf{x}_k)\|.
\end{aligned}$$

Therefore, for this choice of $\gamma_k$ we have that Assumption 3.2 holds as well. Furthermore, by (2.5) we get

$$\mathbb{E}(\gamma_k^2|\mathcal{F}_k) = c_9^2\mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k) \leq c_9^2 c_1(1 + \|\mathbf{x}_k - \mathbf{x}_*\|^2).$$

Assuming that $\{\mathbf{x}_k\}$ is bounded, we get

$$\mathbb{E}(\gamma_k^2|\mathcal{F}_k) \le c_9^2 c_1(1+\|\mathbf{x}_k-\mathbf{x}_*\|^2) \le c_9^2 c_1(1+M_7) := M_8$$

and hence Assumption 3.6 holds as well. The previous observations imply the following convergence statement, whose proof is straightforward.

COROLLARY 3.9. *Let Assumptions 2.1 to 2.3 and Assumption 3.4 hold. Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 2.1 where $\mathbf{d}_k$ satisfies*

$$\|B(\mathbf{x}_k)\mathbf{d}_k + \mathbf{g}(\mathbf{x}_k)\| \le \delta_k \|\mathbf{g}(\mathbf{x}_k)\|,$$

*and $\delta_k$ satisfies (3.3) holds. If $\{\mathbf{x}_k\}$ is bounded then $\mathbf{x}_k \to \mathbf{x}_*$ a.s..*

The next theorem considers the most general case, extending the tolerance for inexact solutions of the Newton linear system (3.7) even further. Let us define

$$\gamma_k = \omega_1 \eta_k + \omega_2 \|\mathbf{g}(\mathbf{x}_k)\| \tag{3.13}$$

for some $\omega_1, \omega_2 \ge 0$ and a random variable $\eta_k$ such that

$$\mathbb{E}(\eta_k^2|\mathcal{F}_k) \le M_9, \tag{3.14}$$

i.e., with bounded variance.

THEOREM 3.10. *Let Assumptions 2.1 to 2.3 and Assumption 3.4 hold. Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 2.1 where $\mathbf{d}_k$ satisfies*

$$\|B(\mathbf{x}_k)\mathbf{d}_k + \mathbf{g}(\mathbf{x}_k)\| \le \delta_k \gamma_k,$$

*with $\gamma_k$ defined by (3.13) and $\delta_k$ such that (3.3) holds. If $\{\mathbf{x}_k\}$ is bounded, then $\mathbf{x}_k \to \mathbf{x}_*$ a.s..*

*Proof.* First, note that the search direction $\mathbf{d}_k$ satisfies

$$\begin{aligned}
\|\mathbf{d}_k\| &= \|B(\mathbf{x}_k)^{-1}(\mathbf{r}_k - \mathbf{g}(\mathbf{x}_k))\| \le \frac{1}{\mu}(\delta_k\gamma_k + \|\mathbf{g}(\mathbf{x}_k)\|) \\
&= \frac{1}{\mu}\left(\omega_1\delta_k\eta_k + (1+\omega_2\delta_k)\|\mathbf{g}(\mathbf{x}_k)\|\right),
\end{aligned}$$

and then, by (3.14), (2.5), and Assumption 2.1,

$$\begin{aligned}
\mathbb{E}(\|\mathbf{d}_k\|^2|\mathcal{F}_k) &\le \frac{2}{\mu^2}\left(\omega_1^2\delta_k^2\,\mathbb{E}(\eta_k^2|\mathcal{F}_k) + (1+\omega_2\delta_k)^2\,\mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k)\right) \\
&\le \frac{2}{\mu^2}\omega_1^2\delta_k^2 M_9 + \frac{2}{\mu^2}(1+\omega_2\delta_k)^2 c_1(1+\|\mathbf{x}_k-\mathbf{x}_*\|^2) \\
&= \frac{2}{\mu^2}\left(\omega_1^2\delta_k^2 M_9 + c_1(1+\omega_2\delta_k)^2\right) + \frac{2}{\mu^2}c_1(1+\omega_2\delta_k)^2\frac{2}{\mu}(\phi(\mathbf{x}_k)-\phi(\mathbf{x}_*)) \\
&\le M_{10} + M_{11}(\phi(\mathbf{x}_k)-\phi(\mathbf{x}_*)),
\end{aligned}$$

where $M_{10} = 2/\mu^2\,(\omega_1^2\delta_{\max}^2 M_9 + c_1(1+\omega_2\delta_{\max})^2)$ and $M_{11} = 4c_1/\mu^3\,(1+\omega_2\delta_{\max})^2$.

Since $\mathbf{d}_k = B(\mathbf{x}_k)^{-1}(\mathbf{r}_k - \mathbf{g}(\mathbf{x}_k))$, using the same arguments as for (3.10) we obtain

$$\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) \le -\frac{1}{L}\|\nabla\phi(\mathbf{x}_k)\|^2 + \nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(B(\mathbf{x}_k)^{-1}\mathbf{r}_k|\mathcal{F}_k). \tag{3.15}$$

Furthermore,

$$
\begin{aligned}
\nabla\phi(\mathbf{x}_k)^\top \mathbb{E}(B(\mathbf{x}_k)^{-1}\mathbf{r}_k|\mathcal{F}_k) &\leq \|\nabla\phi(\mathbf{x}_k)\|\,\mathbb{E}(\|B(\mathbf{x}_k)^{-1}\|\|\mathbf{r}_k\||\mathcal{F}_k)\\
&\leq \frac{1}{\mu}\|\nabla\phi(\mathbf{x}_k)\|\,\delta_k\,\mathbb{E}(\gamma_k|\mathcal{F}_k)
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbb{E}(\gamma_k|\mathcal{F}_k) &= \mathbb{E}(\omega_1\eta_k + \omega_2\|\mathbf{g}(\mathbf{x}_k)\||\mathcal{F}_k)\\
&= \omega_1\mathbb{E}(\eta_k|\mathcal{F}_k) + \omega_2\mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\||\mathcal{F}_k)\\
&\leq \omega_1\sqrt{\mathbb{E}(\eta_k^2|\mathcal{F}_k)} + \omega_2\sqrt{\mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k)}\\
&\leq \omega_1\sqrt{M_9} + \omega_2\sqrt{c_1(1+\|\mathbf{x}_k-\mathbf{x}_*\|^2)}\\
&\leq \omega_1\sqrt{M_9} + \omega_2\sqrt{c_1(1+M_7)} := M_{12}.
\end{aligned}
$$

Putting together the above estimates and using the descent lemma as in the previous proofs, we get

$$
\begin{aligned}
\mathbb{E}(\phi(\mathbf{x}_{k+1})-\phi(\mathbf{x}_*)|\mathcal{F}_k) &\leq \phi(\mathbf{x}_k)-\phi(\mathbf{x}_*) + \alpha_k\nabla\phi(\mathbf{x}_k)^\top\mathbb{E}(\mathbf{d}_k|\mathcal{F}_k) + \alpha_k^2\frac{L}{2}\,\mathbb{E}(\|\mathbf{d}_k\|^2|\mathcal{F}_k)\\
&\leq \phi(\mathbf{x}_k)-\phi(\mathbf{x}_*) + \alpha_k^2\frac{L}{2}\,(M_{10}+M_{11}(\phi(\mathbf{x}_k)-\phi(\mathbf{x}_*))\\
&\quad + \alpha_k\left(-\frac{1}{L}\|\nabla\phi(\mathbf{x}_k)\|^2 + \|\nabla\phi(\mathbf{x}_k)\|\frac{1}{\mu}\delta_k M_{12}\right)\\
&\leq (\phi(\mathbf{x}_k)-\phi(\mathbf{x}_*))\left(1+\alpha_k^2\frac{L}{2}M_{11}\right) + \alpha_k^2\frac{L}{2}M_{10}\\
&\quad + \alpha_k\delta_k\frac{1}{\mu}M_{13}M_{12} - \alpha_k\frac{2}{L^2}(\phi(\mathbf{x}_k)-\phi(\mathbf{x}_*)),
\end{aligned}
$$

where $\|\nabla\phi(\mathbf{x}_k)\| \leq M_{13}$ because of the boundedness of $\{\mathbf{x}_k\}$ and the continuity of $\nabla\phi$. By defining

$$
\beta_k = \alpha_k^2\frac{L}{2}M_{11},\quad \xi_k = \alpha_k^2\frac{L}{2}M_{10} + \alpha_k\delta_k\frac{1}{\mu}M_{13}M_{12},\quad \rho_k = \alpha_k\frac{2}{L^2}(\phi(\mathbf{x}_k)-\phi(\mathbf{x}_*)),
$$

and observing that (3.3) and Assumption 2.2 hold, we can apply Theorem 2.4 to get the thesis. ∎

## 4. SOS method with line search.

**4. SOS method with line search.** It is well known that in practice a gain sequence that satisfies Assumption 2.2 is usually too conservative and makes the algorithm slow because the step length becomes too small soon. In order to avoid this drawback, we propose a practical version of Algorithm SOS that uses a line search in the initial phase and then reduces to SOS if the step length obtained with the line search becomes too small, e.g., smaller than some predetermined threshold $t_{\min} > 0$.

Since the search directions considered in the previous sections do not have to be descent directions (not even for the current objective function approximation), and the line search can be performed considering only the approximate objective function, we choose a nonmonotone line-search strategy.

We state the new algorithmic framework in Algorithm 4.1. Note that this algorithm remains well defined even with the monotone (classical Armijo) line search – if the search direction is not a descent one, we shift to the predefined gain sequence.

We prove the a.s. convergence of Algorithm LSOS under a mild additional assumption.

**Algorithm 4.1** Line-search Second-Order Stochastic (LSOS) method

given $\mathbf{x}^0 \in \mathbb{R}^n$, $\eta \in (0,1)$, $t_{\min} > 0$ and $\{\alpha_k\}, \{\delta_k\}, \{\zeta_k\} \subset \mathbb{R}_+$
set LSphase = *active*
**for** $k = 0, 1, 2, \dots$ **do**
  compute a search direction $\mathbf{d}_k$ such that

$$\|B(\mathbf{x}_k)\mathbf{d}_k + \mathbf{g}(\mathbf{x}_k)\| \le \delta_k \|\mathbf{g}(\mathbf{x}_k)\|. \tag{4.1}$$

  find a step length $t_k$ as follows:
    **if** LSphase = *active* **then** find $t_k$ that satisfies

$$f(\mathbf{x}_k + t_k \mathbf{d}_k) \le f(\mathbf{x}_k) + \eta t_k \mathbf{g}(\mathbf{x}_k)^\top \mathbf{d}_k + \zeta_k \tag{4.2}$$

    **if** $t_k < t_{\min}$ **then** set LSphase = *inactive*
    **if** LSphase = *inactive* **then** set $t_k = \alpha_k$
  set $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k \mathbf{d}_k$
**end for**

ASSUMPTION 4.1. *The objective function estimator $f$ is unbiased, i.e.,*

$$\mathbb{E}(\varepsilon_f(\mathbf{x})|\mathcal{F}_k) = 0.$$

THEOREM 4.2. *Let Assumptions 2.1 to 2.3, Assumption 3.4 and Assumption 4.1 hold. Assume also that the sequence $\{\zeta_k\}$ is summable and the forcing term sequence $\{\delta_k\}$ satisfies (3.3). If the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 4.1 is bounded, then $\mathbf{x}_k \to \mathbf{x}_*$ a.s..*

*Proof.* If there exists an iteration $k$ such that $t_k < t_{\min}$, then Algorithm LSOS reduces to SOS and the thesis follows from Corollary 3.9. Let us consider the case $t_k \ge t_{\min}$ for all $k$. Using $\mathbf{r}_k = B(\mathbf{x}_k)\mathbf{d}_k + \mathbf{g}(\mathbf{x}_k)$ we obtain

$$\mathbf{d}_k = B(\mathbf{x}_k)^{-1}\mathbf{r}_k - B(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k).$$

Furthermore, Assumption 3.4 together with (4.1) implies

$$\begin{aligned}
\mathbf{g}(\mathbf{x}_k)^\top \mathbf{d}_k &= \mathbf{g}(\mathbf{x}_k)^\top B(\mathbf{x}_k)^{-1}\mathbf{r}_k - \mathbf{g}(\mathbf{x}_k)^\top B(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k) \\
&\le \|\mathbf{g}(\mathbf{x}_k)\| \left\|B(\mathbf{x}_k)^{-1}\right\| \|\mathbf{r}_k\| - \frac{1}{L}\|\mathbf{g}(\mathbf{x}_k)\|^2 \\
&\le \frac{1}{\mu}\delta_k \|\mathbf{g}(\mathbf{x}_k)\|^2 - \frac{1}{L}\|\mathbf{g}(\mathbf{x}_k)\|^2 \\
&= \left(\frac{\delta_k}{\mu} - \frac{1}{L}\right)\|\mathbf{g}(\mathbf{x}_k)\|^2.
\end{aligned}$$

Assumption 2.2 together with (3.3) implies $\delta_k \to 0$. Therefore, there exists $\overline{k}$ such that

$$\delta_k \le \frac{\mu}{2L} \quad \text{for all } k \ge \overline{k}$$

and hence

$$\mathbf{g}(\mathbf{x}_k)^\top \mathbf{d}_k \le -\frac{1}{2L}\|\mathbf{g}(\mathbf{x}_k)\|^2. \tag{4.3}$$

Furthermore, LSphase is active at each iteration and for $k \geq \bar{k}$ we get

$$\begin{aligned} f(\mathbf{x}_k + t_k \mathbf{d}_k) &\leq f(\mathbf{x}_k) + \eta\, t_k\, \mathbf{g}(\mathbf{x}_k)^\top \mathbf{d}_k + \zeta_k \\ &\leq f(\mathbf{x}_k) - \eta\, t_k\, \frac{1}{2L} \|\mathbf{g}(\mathbf{x}_k)\|^2 + \zeta_k \\ &\leq f(\mathbf{x}_k) - \eta\, t_{\min}\, \frac{1}{2L} \|\mathbf{g}(\mathbf{x}_k)\|^2 + \zeta_k. \end{aligned}$$

Setting $c_{10} = \eta\, t_{\min}/(2L)$, taking the conditional expectation and using Assumption 4.1, we get

$$\phi(\mathbf{x}_{k+1}) \leq \phi(\mathbf{x}_k) - c_{10}\, \mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2 | \mathcal{F}_k) + \zeta_k. \tag{4.4}$$

Assumption 2.3 implies

$$\mathbb{E}(\mathbf{g}(\mathbf{x}_k)|\mathcal{F}_k) = \nabla \phi(\mathbf{x}_k), \tag{4.5}$$

and thus we get

$$\|\nabla \phi(\mathbf{x}_k)\|^2 = \|\mathbb{E}(\mathbf{g}(\mathbf{x}_k)|\mathcal{F}_k)\|^2 \leq \mathbb{E}^2(\|\mathbf{g}(\mathbf{x}_k)\||\mathcal{F}_k) \leq \mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k), \tag{4.6}$$

which, together with Assumption 2.1, implies

$$\frac{\mu}{L}\|\mathbf{x}_k - \mathbf{x}_*\|^2 \leq \|\nabla \phi(\mathbf{x}_k)\|^2 \leq \mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2|\mathcal{F}_k). \tag{4.7}$$

Combining (4.7) with (4.4) we have

$$\phi(\mathbf{x}_{k+1}) \leq \phi(\mathbf{x}_k) - c_{11}\, \|\mathbf{x}_k - \mathbf{x}_*\|^2 + \zeta_k \quad \text{for all } k \geq \bar{k} \tag{4.8}$$

for a suitable $\bar{k}$, where $c_{11} = c_{10}\mu/L$. The boundedness of the iterates and the continuity of $\phi$ imply the existence of a constant $Q$ such that $\phi(\mathbf{x}_k) \geq Q$ for all $k$. Furthermore, (4.8) implies that, for all $p \in \mathbb{N}$,

$$Q \leq \phi(\mathbf{x}_{\bar{k}+p}) \leq \phi(\mathbf{x}_{\bar{k}}) - c_{11} \sum_{j=0}^{p-1} \|\mathbf{x}_{\bar{k}+j} - \mathbf{x}_*\|^2 + \sum_{j=0}^{p-1} \zeta_{\bar{k}+j}.$$

Taking the expectation, letting $p$ tend to infinity and using the summability of $\zeta_k$, we conclude that

$$\sum_{k=0}^{\infty} \mathbb{E}(\|\mathbf{x}_k - \mathbf{x}_*\|^2) < \infty.$$

Finally, using Markov's inequality we have that for any $\epsilon > 0$

$$P(\|\mathbf{x}_k - \mathbf{x}_*\| \geq \epsilon) \leq \frac{\mathbb{E}(\|\mathbf{x}_k - \mathbf{x}_*\|^2)}{\epsilon^2}$$

and therefore

$$\sum_{k=0}^{\infty} P(\|\mathbf{x}_k - \mathbf{x}_*\| \geq \epsilon) < \infty.$$

The almost sure convergence follows from Borel-Cantelli Lemma [22, Theorem 2.7], which completes the proof. ∎

## 5. Specializing LSOS for finite sums.

Now we consider finite-sum problems, where the objective function is, e.g., the sample mean of a finite family of convex functions. This is the case, for example, of machine learning problems in which the logistic loss, the quadratic loss or other loss functions are used, usually coupled with $\ell_2$-regularization terms. Recently, much attention has been devoted to the development of methods for the solution of problems of this type. Therefore, we analyze extensions to this setting of the LSOS algorithmic framework.

Specifically, we focus on objective functions of the form

$$\phi(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\phi_i(\mathbf{x}), \tag{5.1}$$

where each $\phi_i(\mathbf{x})$ is twice continuously differentiable and $\overline{\mu}$-strongly convex, and has Lipschitz-continuous gradient with Lipschitz constant $\overline{L}$. It is straightforward to show that these assumptions imply that $\phi$ satisfies Assumption 2.1.

We assume that at each iteration $k$ a sample $\mathcal{N}_k$ of size $N_k \ll N$ is chosen randomly and uniformly from $\mathcal{N} = \{1, ..., N\}$. Then, we consider

$$f_{\mathcal{N}_k}(\mathbf{x}) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\phi_i(\mathbf{x}),$$

which is an unbiased estimator of $\phi(\mathbf{x})$, i.e., Assumption 4.1 holds.

By considering the first and second derivatives of $f_{\mathcal{N}_k}$, we obtain the following subsampled gradient and Hessian of $\phi$:

$$\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\nabla\phi_i(\mathbf{x}), \quad B_{\mathcal{N}_k}(\mathbf{x}) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\nabla^2\phi_i(\mathbf{x}), \tag{5.2}$$

which are unbiased estimators of the gradient and the Hessian of $\phi$ as well. More precisely, the first equality in Assumption 2.3 holds (i.e., $\mathbb{E}(\boldsymbol{\varepsilon}_g(\mathbf{x})|\mathcal{F}_k) = 0$) together with Assumption 3.4.

The derivative estimates in (5.2) can be replaced by more sophisticated ones, with the aim of improving the performance of second-order stochastic optimization methods. The Hessian approximation $B_{\mathcal{N}_k}(\mathbf{x})$ only needs to satisfy Assumption 3.4 in order to prove the results contained in this section. Therefore, the theory we develop still holds if one replaces the subsampled Hessian approximation with a quasi-Newton approximation. For example, in [10] Byrd et al. propose to use subsampled gradients and an approximation of the inverse of the Hessian $\nabla^2\phi(\mathbf{x})$, say $H_k$, built by means of a stochastic variant of limited memory BFGS (L-BFGS). Given a memory parameter $m$, $H_k$ is defined by applying $m$ BFGS updates to an initial matrix, using the $m$ most recent correction pairs $(\mathbf{s}_j, \mathbf{y}_j) \in \mathbb{R}^n \times \mathbb{R}^n$ like in the deterministic version of the L-BFGS method. The pairs are obtained by averaging iterates, i.e., every $l$ steps the following vectors are computed

$$\mathbf{w}_j = \frac{1}{l}\sum_{i=k-l+1}^{k}\mathbf{x}_i, \quad \mathbf{w}_{j-1} = \frac{1}{l}\sum_{i=k-2l+1}^{k-L}\mathbf{x}_i, \tag{5.3}$$

where $j = \frac{k}{l}$, and they are used to build $\mathbf{s}_j$ and $\mathbf{y}_j$ as specified next:

$$\mathbf{s}_j = \mathbf{w}_j - \mathbf{w}_{j-1}, \quad \mathbf{y}_j = B_{\mathcal{T}_j}(\mathbf{w}_j)\,\mathbf{s}_j, \tag{5.4}$$

where $\mathcal{T}_j \subset \{1,\ldots,N\}$. By defining the set of the $m$ most recent correction pairs as

$$\{(\mathbf{s}_j,\mathbf{y}_j),\ j=1,\ldots,m\},$$

the inverse Hessian approximation is computed as

$$H_k = H_k^{(m)}, \tag{5.5}$$

where for $j=1,\ldots,m$

$$H_k^{(j)} = \left(I - \frac{\mathbf{s}_j\,\mathbf{y}_j^\top}{\mathbf{s}_j^\top\mathbf{y}_j}\right)^\top H_k^{(j-1)} \left(I - \frac{\mathbf{y}_j\,\mathbf{s}_j^\top}{\mathbf{s}_j^\top\mathbf{y}_j}\right) + \frac{\mathbf{s}_j\,\mathbf{s}_j^\top}{\mathbf{s}_j^\top\mathbf{y}_j}, \tag{5.6}$$

and $H_k^{(0)} = (\mathbf{s}_m^\top\mathbf{y}_m/\|\mathbf{y}_m\|^2)\,I$. It can be proved (see [10, Lemma 3.1] and [30, Lemma 4]) that for approximate inverse Hessians of the form (5.5) there exist constants $\lambda_2 \ge \lambda_1 > 0$ such that

$$\lambda_1 I \preceq H_k \preceq \lambda_2 I,$$

i.e., Assumption 3.4 holds with $\mu = \min\{\overline{\mu},\, 1/\lambda_2\}$ and $L = \max\{\overline{L},\, 1/\lambda_1\}$. The authors of [10] propose a version of Algorithm 2.1 in which the direction is computed as

$$\mathbf{d}_k = -H_k\,\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k),$$

and prove R-linear decrease of the expected value of the error in the function value.

As regards the gradient estimate, we observe that the second part of Assumption 2.3 is not required by the method presented in this section. Notice that we can replace the subsampled gradient estimate in (5.2) with alternative estimates coming, e.g., from variance reduction techniques, which have gained much attention in the literature. This is the case of the stochastic L-BFGS algorithm by Moritz et al. [30] and the stochastic block L-BFGS by Gower et al. [17], where SVRG gradient approximations are used. The former method computes the same inverse Hessian approximation as in [10], while the latter uses an adaptive sketching technique exploiting the action of a sub-sampled Hessian on a set of random vectors rather than just on a single vector. Both stochastic BFGS algorithms use constant step lengths and have Q-linear rate of convergence of the expected value of the error in the objective function, but the block L-BFGS one appears more efficient than the other in most of the numerical experiments reported in [17].

Instead of choosing the SVRG approximation, we apply a mini-batch variant of the SAGA algorithm [12], used in [18]. Starting from the matrix $J^0 \in \mathbb{R}^{n\times N}$ whose columns are defined as $J_0^{(i)} = \nabla\phi_i(\mathbf{x}^0)$, at each iteration we compute the gradient approximation as

$$\mathbf{g}_{\mathcal{N}_k}^{\mathrm{SAGA}}(\mathbf{x}_k) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\left(\nabla\phi_i(\mathbf{x}_k) - J_k^{(i)}\right) + \frac{1}{N}\sum_{l=1}^{N} J_k^{(l)}, \tag{5.7}$$

and, after updating the iterate, we set

$$J_{k+1}^{(i)} = \begin{cases} J_k^{(i)} & \text{if } i \notin \mathcal{N}_k, \\ \nabla\phi_i(\mathbf{x}_{k+1}) & \text{if } i \in \mathcal{N}_k. \end{cases} \tag{5.8}$$

As in SVRG, the set $\{1,\ldots,N\}$ is partitioned into a fixed-number $n_b$ of random mini-batches which are used in order. One advantage of SAGA over SVRG is that it only requires a full gradient computation at the beginning of the algorithm, while SVRG requires a full gradient evaluation each $n_b$ iterations.

*Remark* 5.1. By assuming that all the $\phi_i$'s have Lipschitz-continuous gradients with Lipschitz constant $\overline{L}$, we have that the gradient estimates $\mathbf{g}_{\mathcal{N}_k}(\mathbf{x})$ and $\mathbf{g}_{\mathcal{N}_k}^{\mathrm{SAGA}}(\mathbf{x})$ are Lipschitz continuous with the same Lipschitz constant.

Our algorithmic framework for objective functions of the form (5.1) is called LSOS-FS (where FS stands for Finite Sums) and is outlined in Algorithm 5.1. For the sake of generality, we refer to generic gradient and Hessian approximations, denoted $\mathbf{g}(\mathbf{x}_k)$ and $B(\mathbf{x}_k)$, respectively. We consider the possibility of introducing inexactness in the computation of the direction

$$\mathbf{d}_k = -B(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k),$$

even if for the L-BFGS strategy mentioned above, where $H_k = B(\mathbf{x}_k)^{-1}$, the direction can be computed exactly by a matrix-vector product with $H_k$.

**Algorithm 5.1** LSOS for Finite Sums (LSOS-FS)

1: given $\mathbf{x}^0 \in \mathbb{R}^n$, $\eta, \beta \in (0,1)$, $\{\delta_k\} \subset \mathbb{R}_+$ and $\{\zeta_k\} \subset \mathbb{R}_{++}$
2: **for** $k = 0, 1, 2, \dots$ **do**
3: compute $f_{\mathcal{N}_k}(\mathbf{x}_k)$, $\mathbf{g}(\mathbf{x}_k)$ and $B(\mathbf{x}_k)$
4: find a search direction $\mathbf{d}_k$ such that

$$\|B(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k)\| \le \delta_k \|\mathbf{g}(\mathbf{x}_k)\| \tag{5.9}$$

5: find the smallest integer $j \ge 0$ such that the step length $t_k = \beta^j$ satisfies

$$f_{\mathcal{N}_k}(\mathbf{x}_k + t_k\mathbf{d}_k) \le f_{\mathcal{N}_k}(\mathbf{x}_k) + \eta\, t_k\, \mathbf{g}(\mathbf{x}_k)^\top \mathbf{d}_k + \zeta_k \tag{5.10}$$

6: set $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k\mathbf{d}_k$
7: **end for**

## 6. Convergence theory of Algorithm LSOS-FS.

We assume

$$\sum_k \zeta_k < \infty. \tag{6.1}$$

In the initial phase of the computation, nondescent directions are likely to occur; however, by requiring $\zeta_k > 0$ we ensure that the line search remains well defined. Furthermore, by (6.1) it is $\zeta_k \to 0$, which, by reasoning as in the proof of Theorem 4.2, implies that Algorithm 5.1 will eventually determine a descent direction for the current approximation of the objective function.

Algorithm LSOS-FS computes the step length $t_k$ by applying a backtracking line-search to the approximate function $f_{\mathcal{N}_k}(\mathbf{x})$. In the next lemma we prove that the sequence $\{t_k\}$ is bounded away from zero for all $k$ large enough, if the gradient approximation is the subsampled gradient $\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)$. Throughout this section we use $\delta_{\max}$ defined at the beginning of page 10.

LEMMA 6.1. *Let Algorithm 5.1 be applied to problem (5.1) with $\mathbf{g}(\mathbf{x}_k) = \mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)$, and let $\delta_k \to 0$. Then the step-length sequence $\{t_k\}$ is such that*

$$t_k \ge \frac{\beta(1-\eta)\mu^2}{L^2(1+\delta_{\max})^2} := t_{\min} \in (0,1), \tag{6.2}$$

*for all $k$ large enough.*

*Proof.* If $t_k = 1$, then (6.2) holds. If $t_k < 1$, then there exists $t'_k = t_k/\beta$ such that

$$f_{\mathcal{N}_k}(\mathbf{x}_k + t'_k \mathbf{d}_k) > f_{\mathcal{N}_k}(\mathbf{x}_k) + \eta\, t'_k\, \mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)^\top \mathbf{d}_k. \tag{6.3}$$

Furthermore, by the descent lemma applied to $f_{\mathcal{N}_k}$ and the Lipschitz continuity of $\mathbf{g}_{\mathcal{N}_k}$ we have

$$f_{\mathcal{N}_k}(\mathbf{x}_k + t'_k \mathbf{d}_k) \le f_{\mathcal{N}_k}(\mathbf{x}_k) + t'_k\, \mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)^\top \mathbf{d}_k + \frac{L}{2}(t'_k)^2 \|\mathbf{d}_k\|^2. \tag{6.4}$$

Combining (6.3) and (6.4) we obtain

$$t_k = \beta t'_k > \frac{-2\beta(1-\eta)\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)^\top \mathbf{d}_k}{L\|\mathbf{d}_k\|^2}. \tag{6.5}$$

Following the proof of Theorem 4.2, we can show that (4.3) holds for all $k \ge \overline{k}$ with $\mathbf{g} = \mathbf{g}_{\mathcal{N}_k}$ and thus

$$-\,\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)^\top \mathbf{d}_k \ge \frac{\|\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)\|^2}{2L}. \tag{6.6}$$

On the other hand,

$$\begin{aligned}
\|\mathbf{d}_k\| = \left\|(B_{\mathcal{N}_k}(\mathbf{x}_k))^{-1}(\mathbf{r}_k - \mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k))\right\| &\le \frac{1}{\mu}(\|\mathbf{r}_k\| + \|\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)\|)\\
&\le \frac{\delta_k + 1}{\mu}\|\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)\|,
\end{aligned}$$

where the last inequality comes from (5.9). Therefore, since $\delta_k \le \delta_{\max}$, we obtain

$$\|\mathbf{d}_k\|^2 \le \frac{(\delta_{\max}+1)^2}{\mu^2}\|\mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)\|^2.$$

This, together with (6.5) and (6.6), gives the thesis. ∎

In the following theorem we state the convergence of the LSOS-FS method. The proof is omitted since it follows the steps of the proof of Theorem 4.2. The Lemma above exploits $\mathbf{g}(\mathbf{x}_k) = \mathbf{g}_{\mathcal{N}_k}(\mathbf{x}_k)$; for general $\mathbf{g}(\mathbf{x}_k)$ we have to assume that the step lengths are bounded away from zero. Notice that we do not need the assumption of bounded iterates since the line search is performed at each iteration and the function is strongly convex and thus bounded from below.

THEOREM 6.2. *Let $\{\mathbf{x}_k\}$ be the sequence generated by Algorithm 5.1 applied to problem (5.1). Assume that (6.1) and Assumption 3.4 hold, and $\mathbf{g}(\mathbf{x})$ is a Lipschitz-continuous unbiased gradient estimate. Moreover, assume $\{t_k\}$ is bounded away from zero. Then $\{\mathbf{x}_k\}$ converges a.s. to the unique minimizer of $\phi$.*

Finally, we provide the convergence rate analysis of LSOS-FS. We prove that the expected function error converges $R$-linearly provided that $\zeta_k$ vanishes $R$-linearly. We also prove that a $Q$-linear rate of convergence can be achieved if the monotone (Armijo) line search is employed and the descent direction is ensured. The latter condition can be provided by putting an upper bound on the forcing term, which is in line with the classical (deterministic) analysis. The results are stated in the following three theorems, whose proofs rely on the steps of the proof of Theorem 4.2. Since $L$ is an upper bound of the spectrum of the Hessian estimates, without loss of generality we can assume $L \ge 1$.

THEOREM 6.3. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 5.1 applied to problem (5.1). Let $\delta_k \to 0$ and let $\zeta_k \to 0$ R-linearly. Let Assumption 3.4 hold, $\mathbf{g}(\mathbf{x})$ be a Lipschitz-continuous unbiased gradient estimate and the sequence $\{t_k\}$ be bounded away from zero. Then there exist constants $\rho_1 \in (0,1)$ and $C > 0$ such that*

$$\mathbb{E}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) \leq \rho_1^k C. \tag{6.7}$$

*Proof.* Let $t_{\min}$ be a lower bound for the sequence $\{t_k\}$. Following the steps of the proof of Theorem 4.2 we obtain (4.4) with $c_{10} = \eta\, t_{\min}/(2L)$, or equivalently,

$$\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*) \leq \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) - c_{10}\, \mathbb{E}(\|\mathbf{g}(\mathbf{x}_k)\|^2 | \mathcal{F}_k) + \zeta_k.$$

Moreover, using (4.5), (4.6) and the right-hand inequality in (2.2), we have

$$\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*) \leq \phi(\mathbf{x}_k) - \phi(\mathbf{x}_*) - \frac{2c_{10}}{L}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) + \zeta_k.$$

Applying the expectation we get

$$\mathbb{E}(\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*)) \leq \rho\, \mathbb{E}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) + \zeta_k, \tag{6.8}$$

where $\rho = 1 - 2c_{10}/L = 1 - \eta t_{\min}/L^2 \in (0,1)$. Applying the induction argument we obtain

$$\mathbb{E}(\phi(\mathbf{x}_j) - \phi(\mathbf{x}_*)) \leq \rho^j \mathbb{E}(\phi(\mathbf{x}_0) - \phi(\mathbf{x}_*)) + \upsilon_j,$$

where $\upsilon_j = \sum_{i=1}^{j-1} \rho^{i-1} \zeta_{j-i}$. The thesis follows by recalling that Lemma 4.2 from [23] implies $\upsilon_j \to 0$ R-linearly, with a factor $\rho_v = \frac{1}{2}(1 + \max\{\rho, \rho_\zeta\}) \in (0,1)$, where $\rho_\zeta \in (0,1)$ is an R-linear convergence factor of the sequence $\zeta_k$. Finally, the statement holds with $\rho_1 = \max\{\rho, \rho_v\}$. ∎

Notice that the condition $\delta_k \to 0$ can be relaxed with $0 < \delta_k \to \delta_{\min}$ where $\delta_{\min} < \mu/(2L)$. The reason is that, eventually, the inexact second-order direction becomes a descent direction if (6.6) holds for all $k$ large enough. Under the same argument we can prove Lemma 6.1 and the proof is essentially the same as for Theorem 6.3. Thus, the R-linear convergence is attainable under the persistent inexactness in solving the Newton equation.

THEOREM 6.4. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 5.1 applied to problem (5.1). Assume that $\zeta_k \to 0$ R-linearly and $\delta_k \to \delta_{\min}$, where $\delta_{\min} < \mu/(2L)$. Moreover, let Assumption 3.4 be satisfied, $\mathbf{g}(\mathbf{x})$ be a Lipschitz-continuous unbiased gradient estimate and $\{t_k\}$ be bounded away from zero. Then there exist $\rho_1 \in (0,1)$ and $C > 0$ such that (6.7) holds.*

An immediate consequence of the previous theorem is the following worst-case complexity result.

COROLLARY 6.5. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 5.1 applied to problem (5.1). Assume that $\zeta_k \to 0$ R-linearly and $\delta_k \to \delta_{\min}$, where $\delta_{\min} < \mu/(2L)$. Moreover, Let Assumption 3.4 be satisfied, $\mathbf{g}(\mathbf{x})$ be a Lipschitz-continuous unbiased gradient estimate and $\{t_k\}$ be bounded away from zero. Then, to achieve $\mathbb{E}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) \leq \varepsilon$ for some $\varepsilon \in (0, e^{-1})$, Algorithm 5.1 takes at most*

$$k_{\max} = \left\lceil \frac{|log(C)| + 1}{|log(\rho_1)|} log(\varepsilon^{-1}) \right\rceil,$$

*where $\rho_1 \in (0,1)$ and $C > 0$ satisfy (6.7).*

*Proof.* Theorem 6.4 implies (6.7). Thus, $\mathbb{E}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) \leq \varepsilon$ for all

$$k \geq \frac{log(C) - log(\varepsilon)}{-log(\rho_1)}.$$

Now, using the fact that $log(\varepsilon) < -1$ and $log(\rho_1) < 0$ we can provide an upper bound to the right-hand side of the previous inequality as follows

$$\frac{log(C) - log(\varepsilon)}{-log(\rho_1)} \leq \frac{|log(C)| log(\varepsilon^{-1}) + log(\varepsilon^{-1})}{|log(\rho_1)|} = \frac{|log(C)| + 1}{|log(\rho_1)|} log(\varepsilon^{-1})$$

and the thesis holds. ◻

In order to achieve a Q-linear rate of convergence, the standard Armijo line search has to be used, i.e., $\zeta_k = 0$ has to be set in (5.10). Again, the forcing terms $\delta_k$ need not vanish in order to achieve the desired rate (i.e., Newton's equation can be solved inexactly), but it must be bounded above away from one. More in detail, it must be $\delta_{\max} \leq \mu/(2L)$, as stated in the following theorem. A sequence $\{\delta_k\}$ satisfying the requirement of the theorem can be defined as $\delta_k = \mu/(2L)$ for all $k$.

THEOREM 6.6. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 5.1 applied to problem (5.1). Assume that $\delta_{\max} \leq \mu/(2L)$ and $\zeta_k = 0$ for all $k$. Moreover, suppose that Assumption 3.4 is satisfied, $\mathbf{g}(\mathbf{x})$ is a Lipschitz-continuous unbiased gradient estimate and the sequence $\{t_k\}$ is bounded away from zero. Then there exists $\rho_2 \in (0, 1)$ such that for all $k$*

$$\mathbb{E}(\phi(\mathbf{x}_{k+1}) - \phi(\mathbf{x}_*)) \leq \rho_2 \mathbb{E}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)). \tag{6.9}$$

*Proof.* Notice that the Lipschitz continuity of the gradient estimate implies that (6.6) holds for every $k$ since $\delta_k \leq \mu/(2L)$. Let $t_{\min}$ be a lower bound for the sequence $\{t_k\}$. By following the steps of the proof of Theorem 6.3, we have that (6.8) holds with $\zeta_k = 0$. Therefore, by setting

$$\rho_2 = \rho = 1 - \frac{\eta t_{\min}}{L^2} \leq 1 - \frac{\eta(1-\eta)\beta\mu^2}{L^2(2L+\mu)}$$

the thesis holds. ◻

Since Theorem 6.6 implies $\mathbb{E}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) \leq \rho_2^k(\phi(\mathbf{x}_0) - \phi(\mathbf{x}_*))$, following the same reasoning as in Corollary 6.5, we obtain the following complexity result.

COROLLARY 6.7. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 5.1 applied to problem (5.1). Assume that $\delta_{\max} \leq \mu/(2L)$ and $\zeta_k = 0$ for all $k$. Moreover, suppose that Assumption 3.4 is satisfied, $\mathbf{g}(\mathbf{x})$ is a Lipschitz-continuous unbiased gradient estimate and the sequence $\{t_k\}$ is bounded away from zero. Then, in order to achieve $\mathbb{E}(\phi(\mathbf{x}_k) - \phi(\mathbf{x}_*)) \leq \varepsilon$ for some $\varepsilon \in (0, e^{-1})$, LSOS-FS takes at most $k_{\max} = \mathcal{O}(log(\varepsilon^{-1}))$ iterations. More precisely,*

$$k_{\max} = \left\lceil \frac{|log(\phi(\mathbf{x}_0) - \phi(\mathbf{x}_*))| + 1}{|log(\rho_2)|} log(\varepsilon^{-1}) \right\rceil,$$

*where $\rho_2$ satisfies (6.9).*

**7. Numerical experiments.** We developed MATLAB implementations of the algorithms discussed in the previous sections and tested them on two sets of stochastic optimization problems. The first set consists of general convex problems with the addition of random noise in the evaluation of the objective function and its derivatives. On these problems we tested Algorithms SOS and LSOS discussed in sections 2 to 4. The second set consists of finite-sum problems arising in training linear classifiers with regularized logistic regression models. On these problems we tested a specialized version of LSOS-FS. All the tests were run with MATLAB R2019b on a server available at the University of Campania "L. Vanvitelli", equipped with 8 Intel Xeon Platinum 8168 CPUs, 1536 GB of RAM and Linux CentOS 7.5 operating system.

**7.1. Convex random problems.** The first set of test problems was defined by setting

$$\phi(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \left(e^{x_i} - x_i\right) + (\mathbf{x} - \mathbf{e})^\top A(\mathbf{x} - \mathbf{e}), \tag{7.1}$$

where, given a scalar $\kappa \gg 1$, the coefficients $\lambda_i$ are logarithmically spaced between 1 and $\kappa$, $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite with eigenvalues $\lambda_i$, and $\mathbf{e} \in \mathbb{R}^n$ has all entries equal to 1. Changing the values of $n$ and $\kappa$ allows us to have strongly convex problems with variable size and conditioning. In order to obtain unbiased estimates of $\phi$ and its gradient, we considered $\varepsilon_f(\mathbf{x}) \sim \mathcal{N}(0,\sigma)$ and $(\boldsymbol{\varepsilon}_g(\mathbf{x}))_i \sim \mathcal{N}(0,\sigma)$ for all $i$, where $\mathcal{N}(0,\sigma)$ is the normal distribution with mean 0 and standard deviation $\sigma$. We considered $\sigma \in (0,1]$. Since the Hessian estimate can be biased, we set it equal to the diagonal matrix $\varepsilon_B(\mathbf{x}) = \mathrm{diag}\,(\mu_1,\ldots,\mu_n)$, where $\mu_j \sim \mathcal{N}(0,\sigma)$ for all $j$.

In applying Algorithm 4.1 to this set of problems, we introduced a small modification in the switching criterion at line 7 of the algorithm, by deactivating the line search whenever $t_k\|\mathbf{d}_k\| < t_{\min}$ instead of deactivating it when $t_k < t_{\min}$.

We first ran Algorithm LSOS with exact solution of the noisy Newton systems, i.e., $\delta_k = 0$ in (4.1). The parameters were set as $n = 10^3$, $\kappa = 10^2, 10^3, 10^4$, $\sigma = 0.1\%\,\kappa, 0.5\%\,\kappa, 1\%\,\kappa$, and $A$ was generated by using the MATLAB `sprandsym` function with density 0.5 and eigenvalues $\lambda_1,\ldots,\lambda_n$. It was verified experimentally that the condition number of the Hessian of $\phi$ is close to $\kappa$ at the solution. This solution was computed with high accuracy by using the (deterministic) L-BFGS implementation by Mark Schmidt, available from https://www.cs.ubc.ca/~schmidtm/Software/minFunc.html. The starting point was set as a random vector with distribution of entries $\mathcal{N}(0,5)$. The noisy Newton systems were solved by the MATLAB `backslash` operator. The parameter used to switch between the line search and the pre-defined gain sequence was set as $t_{\min} = 10^{-3}$. The gain sequence $\{\alpha_k\}$ used after the deactivation of the line search was defined as

$$\alpha_k = \alpha_{k_\tau} \frac{T}{T + k - k_\tau} \quad \text{for all } k > k_\tau,$$

where $k_\tau$ is the first iteration such that $t_{k_\tau}\|\mathbf{d}_{k_\tau}\| < t_{\min}$, $\alpha_{k_\tau} = t_{\min}/\|\mathbf{d}_{k_\tau}\|$ and $T = 10^6$. In the nonmonotone line search we set $\eta = 10^{-4}$ and $\zeta_k = \vartheta^k$ for all $k$, where $\vartheta = 0.9$.

LSOS was compared with the following algorithms:

- SOS (Algorithm 2.1) with exact solution of the noisy Newton systems and gain sequence defined as

$$\alpha_k = \frac{1}{\|\mathbf{d}^0\|} \frac{T}{T + k} \tag{7.2}$$

- Stochastic Gradient Descent with step length (7.2), referred to as SGD.

For both SOS and SGD the choice of the starting point was the same as for LSOS.

The comparison was performed in terms of the absolute error of the objective function value (with respect to the optimal value computed by the deterministic L-BFGS algorithm) versus the execution time. We ran each algorithm 20 times on each problem and computed the average error and the average execution time spent until each iteration $k$. The results are shown in Figure 1, where each error line is plotted with its 95% confidence interval (which does not appear in the pictures because its size is negligible). The time interval on the $x$ axis is the average time required by LSOS to perform 50 iterations.

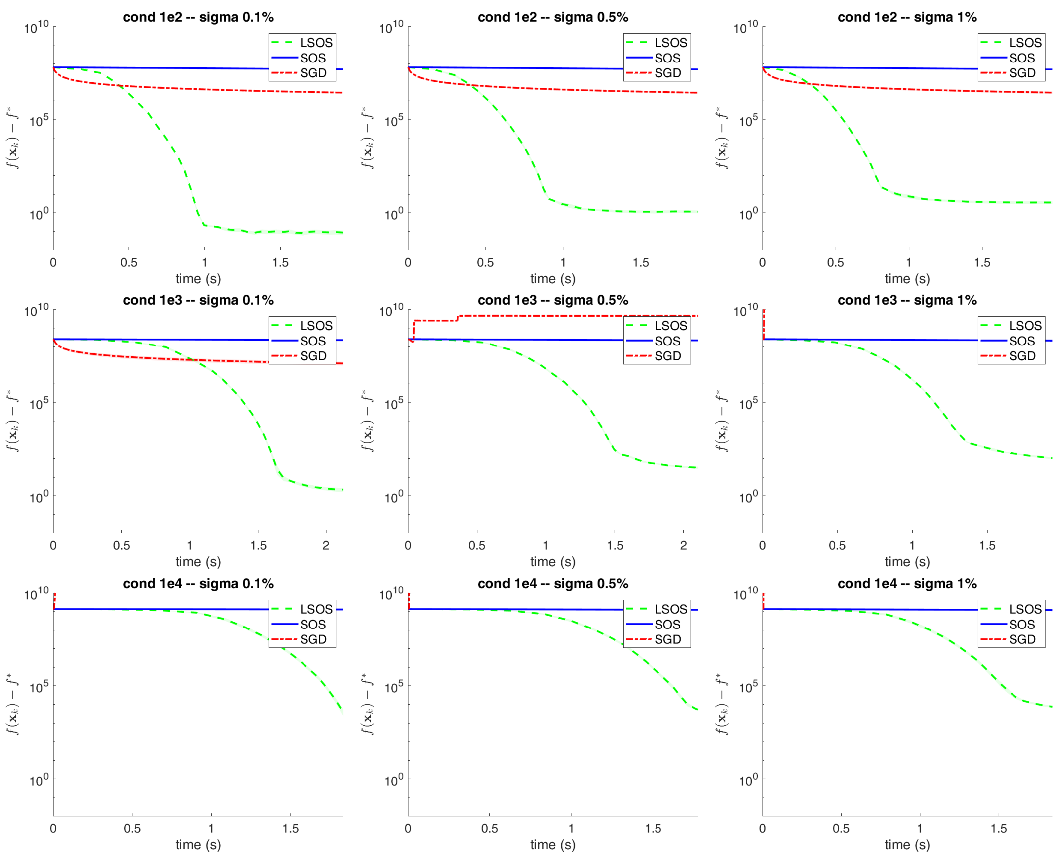


FIG. 1. *Test set 1: comparison of LSOS, SOS and SGD. The condition number increases from top to bottom, the noise increases from left to right.*

The figure shows that the introduction of the line search yields much better exploitation of the second-order directions, thus enabling the method to approach the solution faster. The line search also allows us to overcome typical problems associated with the choice of a pre-defined gain sequence, which may strongly affect the speed of the algorithm and possibly lead to divergence in practice.

We also investigated the effect of the inexactness in the solution of the noisy Newton systems. To this aim, we considered problems of the form (7.1) with size $n = 2 \cdot 10^4$, where, following [14], the symmetric positive definite matrix $A$ was defined as

$$A = V\, D\, V^T.$$

Here $D$ is a diagonal matrix with diagonal entries $\lambda_1, \ldots, \lambda_n$ and

$$V = (I - 2\,\mathbf{v}_3\mathbf{v}_3^T)(I - 2\,\mathbf{v}_2\mathbf{v}_2^T)(I - 2\,\mathbf{v}_1\mathbf{v}_1^T),$$

with $\mathbf{v}_j$ random vectors of unit norm. Since for these problems the Hessian is available in factorized form, we solved the noisy Newton systems with the Conjugate Gradient (CG) method implemented in the MATLAB `pcg` function, exploting the factorization to compute matrix-vector products. In this case, we compared three versions of Algorithm LSOS:

- LSOS with with $\delta_k = 0$ in (4.1);
- LSOS with $\delta_k = \varrho^k$ and $\varrho = 0.95$, referred to as LSOS-I (where I denotes the inexact solution of the Newton systems according to (4.1));
- a line-search version of the SGD algorithm (corresponding to LSOS with $\mathbf{d}_k = -\mathbf{g}(\mathbf{x}_k)$), referred to as SGD-LS.

The CG method in LSOS and LSOS-I was run until the residual norm of the Newton system had been reduced by $\max(\delta_k, 10^{-6})$ with respect to the initial residual norm.

In Figure 2 we report the results obtained with the three algorithms, in terms of average error on the objective function versus average execution time over 20 runs, with 95% confidence intervals (not visible, as in the previous tests). In this case the time interval on the $x$ axis is the average time required by LSOS-I to perform 250 iterations. The plots clearly show that LSOS-I outperforms the other methods.

**7.2. Binary classification problems.** The second set of test problems models the training a linear classifier by minimization of the $\ell_2$-regularized logistic regression. Given $N$ pairs $(\mathbf{a}_i, b_i)$, where $\mathbf{a}_i \in \mathbb{R}^n$ is a training point and $b_i \in \{-1, 1\}$ the corresponding class label, an unbiased hyperplane approximately separating the two classes can be found by minimizing the function

$$\phi(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \phi_i(\mathbf{x}), \tag{7.3}$$

where

$$\phi_i(\mathbf{x}) = \log\left(1 + e^{-b_i\, \mathbf{a}_i^\top \mathbf{x}}\right) + \frac{\mu}{2}\|\mathbf{x}\|^2$$

and $\mu > 0$. By setting $z_i(\mathbf{x}) = 1 + e^{-b_i\, \mathbf{a}_i^\top \mathbf{x}}$, the gradient and the Hessian of $\phi_i$ are

$$\nabla\phi_i(\mathbf{x}) = \frac{1 - z_i(\mathbf{x})}{z_i(\mathbf{x})} b_i\, \mathbf{a}_i + \mu\mathbf{x} \quad \text{and} \quad \nabla^2\phi_i(\mathbf{x}) = \frac{z_i(\mathbf{x}) - 1}{z_i^2(\mathbf{x})}\mathbf{a}_i\mathbf{a}_i^\top + \mu I.$$

From $\frac{z_i(\mathbf{x})-1}{z_i^2(\mathbf{x})} \in (0,1)$ it follows that $\phi_i$ is $\mu$-strongly convex and

$$\mu I \preceq \nabla^2\phi_i(\mathbf{x}) \preceq LI, \quad L = \mu + \max_{i=1,\ldots,N} \|a_i\|^2.$$

We applied the L-BFGS version of Algorithm LSOS-FS described in section 5, which is sketched in Algorithm 7.1.

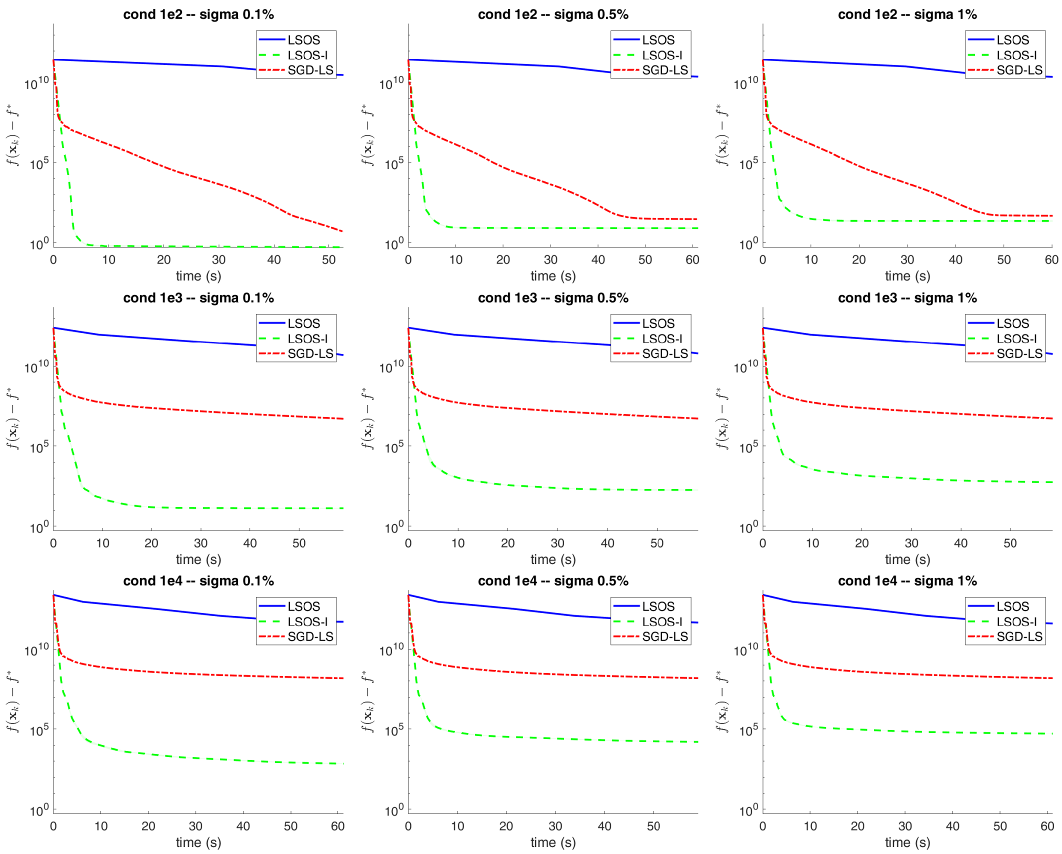


FIG. 2. *Test set 2: comparison of LSOS, LSOS-I and SGD-LS. The condition number increases from top to bottom, the noise increases from left to right.*

**Algorithm 7.1** LSOS-BFGS

given $\mathbf{x}^0 \in \mathbb{R}^n$, $m, l \in \mathbb{N}$, $\eta, \vartheta \in (0,1)$
**for** $k = 0, 1, 2, \ldots$ **do**
  compute a partition $\{\mathcal{K}_0, \mathcal{K}_1, \ldots, \mathcal{K}_{n_b-1}\}$ of $\{1, \ldots, N\}$
  **for** $r = 0, \ldots, n_b - 1$ **do**
    choose $\mathcal{N}_k = \mathcal{K}_r$ and compute $\mathbf{g}(\mathbf{x}_k) = \mathbf{g}^{\mathrm{SAGA}}_{\mathcal{N}_k}(\mathbf{x}_k)$ as in (5.7)-(5.8)
    compute $\mathbf{d}_k = -H_k\, \mathbf{g}(\mathbf{x}_k)$ with $H_k$ defined in (5.5)-(5.6)
    find a step length $t_k$ satisfying
$$f_{\mathcal{N}_k}(\mathbf{x}_k + t_k \mathbf{d}_k) \leq f_{\mathcal{N}_k}(\mathbf{x}_k) + \eta t_k\, \mathbf{g}(x_k)^\top \mathbf{d}_k + \vartheta^k$$
    set $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k \mathbf{d}_k$;
    **if** $\mod(k, l) = 0$ and $k \geq 2\,l$ **then**
      update the L-BFGS correction pairs by using (5.3)-(5.4)
    **end if**
  **end for**
**end for**

To test the effectiveness of LSOS-BFGS we considered six binary classification datasets from the LIBSVM collection available from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, which we list in Table 1.

TABLE 1
*Datasets from LIBSVM. For each dataset the number of training points and the number of features (space dimension) are reported; the datasets are sorted by the increasing number of features. Whenever a training set was not specified in LIBSVM, we selected it by using the MATLAB* `crossvalind` *function so that it contained 70% of the available data.*

| name | $N$ | $n$ |
|---|---|---|
| covtype | 406709 | 54 |
| w8a | 49749 | 300 |
| epsilon | 400000 | 2000 |
| gisette | 6000 | 5000 |
| real-sim | 50617 | 20958 |
| rcv1 | 20242 | 47236 |

We compared Algorithm 7.1 with the stochastic L-BFGS algorithms proposed in [17] and [30] (referred to as GGR and MNJ, respectively), both using a constant step length selected by means of a grid search over the set $\{1, 5\cdot 10^{-1}, 10^{-1}, 5\cdot 10^{-2}, 10^{-2}, \ldots, 5\cdot 10^{-5}, 10^{-5}\}$, and with a mini-batch variant of the SAGA algorithm equipped with the same line search used in LSOS-BFGS. The implementations of GGR and MNJ were taken from the MATLAB StochBFGS code available from https://perso.telecom-paristech.fr/rgower/software/StochBFGS_dist-0.0.zip. In Algorithm 7.1 we set $\vartheta = 0.999$ and started the line searches from a value $t_{\rm ini}$ selected by means of a grid search over $\{1, 5\cdot 10^{-1}, 10^{-1}, 5\cdot 10^{-2}, 10^{-2}, \ldots, 5\cdot 10^{-5}, 10^{-5}\}$. In particular, we set $t_{\rm ini} = 5\cdot 10^{-3}$ for epsilon, $t_{\rm ini} = 5\cdot 10^{-2}$ for covtype and w8a, and $t_{\rm ini} = 1\cdot 10^{-2}$ for gisette, rcv1 and real-sim. We adopted the same strategy as the line-search version of SAGA used for the comparison, setting $t_{\rm ini} = 5\cdot 10^{-1}$ for epsilon and $t_{\rm ini} = 1$ for the other datasets. Furthermore, we set $m = 10$ and $l = 5$. Since the first L-BFGS update pair is available after the first $2l = 10$ iterations, following [10] we take $\mathbf{d}_k = -\mathbf{g}(\mathbf{x}_k)$ for the first 10 iterations. The same values of $m$ and $l$ were used in the MNJ algorithm proposed in [30]. For GGR, following the indications coming from the results in [17], we set $m = 5$ and used the sketching based on the previous directions (indicated as `prev` in [17]), with sketch size $l = \lceil \sqrt[3]{n} \rceil$. We chose the sample size equal to $\left\lceil \sqrt{N} \right\rceil$ and the regularization parameter $\mu = 1/N$, as in the experiments reported in [17]. We decided to stop the algorithms when a maximum execution time was reached, i.e., 60 seconds for covtype, w8a and gisette, and 300 seconds for epsilon, real-sim and rcv1.

Figure 3 shows a comparison among the four algorithms in terms of the average absolute error of the objective function (with respect to the optimal value computed with the L-BFGS code by Mark Schmidt) versus the average execution time. As in the previous experiments, the error and the times were averaged over 20 runs and the plots show their 95% confidence interval (shaded lines, when visible). For all the algorithms, the grid search for defining or initializing the step lengths was performed on the first of the 20 runs and then fixed for the remaining 19 runs.

The results show that LSOS-BFGS algorithm outperforms the other stochastic L-BFGS algorithms on w8a and gisette, and outperforms GGR on real-sim and rcv1. It is worth noting that for covtype and rcv1 the error for GGR tends to increase after a certain iteration, while the other algorithms seem to keep a much less "swinging" decrease. Furthermore, LSOS-BFGS seems to have a less oscillatory behavior with respect to GGR and MNJ. We conjecture that this behavior is due to the use of the line-search strategy. Since, in general, stopping criteria on this type of problems rely on the number of iterations, the number of epochs or the computational time, we believe that a smoother behaviour could be associated with more consistent results if one decides to stop the execution in advance (see, e.g., the behavior of MNJ on epsilon). Finally, we observe that LSOS-BFGS is more efficient than the line-search-based mini-batch SAGA on all the problems, showing that the introduction of stochastic second-order information is crucial for the performance of the algorithm.

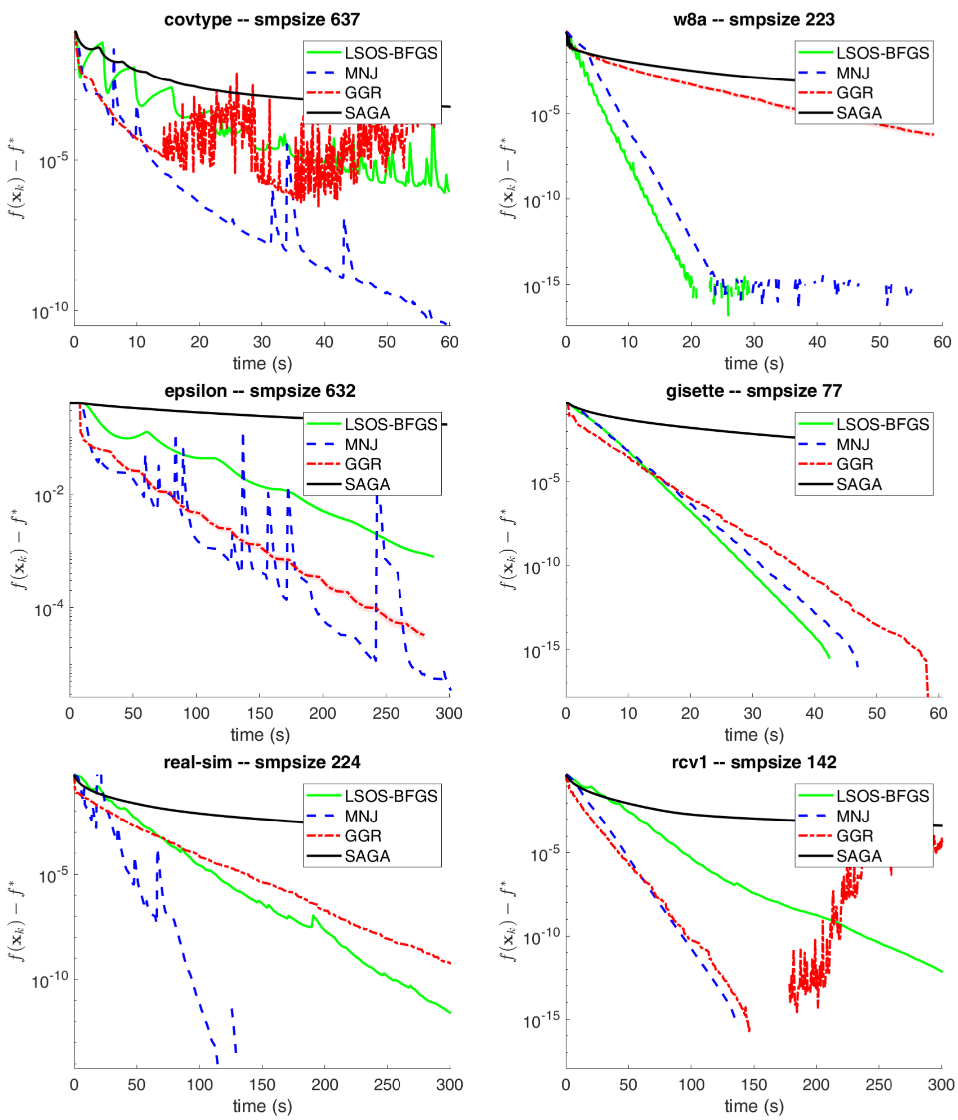


FIG. 3. *Binary classification problems: comparison of LSOS-BFGS, MNJ, GGR and SAGA.*

**8. Conclusions.** The proposed LSOS framework includes a variety of second-order stochastic optimization algorithms, using Newton, inexact Newton and, for finite-sum problems, limited-memory quasi-Newton directions. Almost sure convergence of the sequences generated by all the LSOS variants has been proved. For finite-sum problems, R-linear and Q-linear convergence rates of the expected objective function error have been proved for stochastic L-BFGS Hessian approximations and any Lipschitz-continuous unbiased gradient estimates. In this case, an $\mathcal{O}(\log(\varepsilon^{-1}))$ complexity bound has been also provided.

Numerical experiments have confirmed that line-search techniques in second-order stochastic methods yield a significant improvement over predefined step-length sequences. Furthermore, in the case of finite-sum problems, the experiments have shown that combining stochastic L-BFGS Hessian approximations with the SAGA variance reduction technique and with line searches produces methods that are highly competitive with state-of-the art second-order stochastic optimization methods.

A challenging future research agenda includes the extension of (some) of these results to problems that do not satisfy the strong convexity assumption, as well as extensions to constrained stochastic problems.

## REFERENCES


[1] S. Bellavia, G. Gurioli, and B. Morini, *Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization*, IMA Journal of Numerical Analysis, (2020), https://doi.org/10.1093/imanum/drz076.

[2] S. Bellavia, N. Krejić, and N. Krklec Jerinkić, *Subsampled inexact Newton methods for minimizing large sums of convex functions*, IMA Journal of Numerical Analysis, (2019), https://doi.org/10.1093/imanum/drz027.

[3] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive algorithms and stochastic approximations*, vol. 22 of Applications of Mathematics (New York), Springer-Verlag, Berlin, 1990, https://doi.org/10.1007/978-3-642-75894-2. Translated from the French by Stephen S. Wilson.

[4] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific Optimization and Computation Series, Athena Scientific, Belmont, MA, second ed., 1999.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Gradient convergence in gradient methods with errors*, SIAM J. Optim., 10 (2000), pp. 627–642, https://doi.org/10.1137/S1052623497331063.

[6] R. Bollapragada, R. H. Byrd, and J. Nocedal, *Exact and inexact subsampled Newton methods for optimization*, IMA J. Numer. Anal., 39 (2019), pp. 545–578, https://doi.org/10.1093/imanum/dry009.

[7] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization methods for large-scale machine learning*, SIAM Rev., 60 (2018), pp. 223–311, https://doi.org/10.1137/16M1080173.

[8] R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal, *On the use of stochastic Hessian information in optimization methods for machine learning*, SIAM J. Optim., 21 (2011), pp. 977–995, https://doi.org/10.1137/10079923X.

[9] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, *Sample size selection in optimization methods for machine learning*, Math. Program., 134 (2012), pp. 127–155, https://doi.org/10.1007/s10107-012-0572-5.

[10] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, *A stochastic Quasi-Newton method for large-scale optimization*, SIAM J. Optim., 26 (2016), pp. 1008–1031, https://doi.org/10.1137/140954362.

[11] P. J. Carrington, J. Scott, and S. Wasserman, eds., *Models and Methods in Social Network Analysis*, Structural Analysis in the Social Sciences, Cambridge University Press, 2005, https://doi.org/10.1017/CBO9780511811395.

[12] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Inc., 2014, pp. 1646–1654, https://dl.acm.org/doi/10.5555/2968826.2969010.

[13] B. Delyon and A. Juditsky, *Accelerated stochastic approximation*, SIAM J. Optim., 3 (1993), pp. 868–881, https://doi.org/10.1137/0803045.

[14] D. di Serafino, G. Toraldo, M. Viola, and J. Barlow, *A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables*, SIAM J. Optim., 28 (2018), pp. 2809–2838, https://doi.org/10.1137/17M1128538.

[15] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422, https://doi.org/10.1137/0804022.

[16] M. C. Fu, *Chapter 19 Gradient Estimation*, in Simulation, S. G. Henderson and B. L. Nelson, eds., vol. 13 of Handbooks in Operations Research and Management Science, Elsevier, 2006, pp. 575–616, https://doi.org/10.1016/S0927-0507(06)13019-4.

[17] R. M. Gower, D. Goldfarb, and P. Richtárik, *Stochastic block BFGS: Squeezing more curvature out of data*, in Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, JMLR.org, 2016, pp. 1869–1878, https://dl.acm.org/doi/10.5555/3045390.3045588.

[18] R. M. Gower, P. Richtárik, and F. Bach, *Stochastic quasi-gradient methods: variance reduction via Jacobian sketching*, Mathematical Programming, (2020), https://doi.org/10.1007/s10107-020-01506-0.

[19] A. N. Iusem, A. Jofré, R. I. Oliveira, and P. Thompson, *Variance-based extragradient methods with line search for stochastic variational inequalities*, SIAM J. Optim., 29 (2019), pp. 175–206, https://doi.org/10.1137/17M1144799, https://doi.org/10.1137/17M1144799.

[20] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds., Curran Associates, Inc., 2013, pp. 315–323, https://dl.acm.org/doi/10.5555/2999611.2999647.

[21] H. Kesten, *Accelerated stochastic approximation*, Ann. Math. Statist., 29 (1958), pp. 41–59, https://doi.org/10.1214/aoms/1177706705.

[22] A. Klenke, *Probability theory*, Universitext, Springer, London, second ed., 2014, https://doi.org/10.1007/978-1-4471-5361-0. A comprehensive course.

[23] N. Krejić and N. Krklec Jerinkić, *Nonmonotone line search methods with variable sample size*, Numer. Algorithms, 68 (2015), pp. 711–739, https://doi.org/10.1007/s11075-014-9869-1.

[24] N. Krejić, Z. Lužanin, Z. Ovcin, and I. Stojkovska, *Descent direction method with line search for unconstrained optimization in noisy environment*, Optim. Methods Softw., 30 (2015), pp. 1164–1184, https://doi.org/10.1080/10556788.2015.1025403.

[25] N. Krejić, Z. Lužanin, and I. Stojkovska, *A gradient method for unconstrained optimization in noisy environment*, Appl. Numer. Math., 70 (2013), pp. 1–21, https://doi.org/10.1016/j.apnum.2013.02.006.

[26] K. Marti, *Stochastic optimization methods*, Springer, Heidelberg, third ed., 2015, https://doi.org/10.1007/978-3-662-46214-0. Applications in engineering and operations research.

[27] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez, *Packmol: A package for building initial configurations for molecular dynamics simulations*, Journal of Computational Chemistry, 30 (2009), pp. 2157–2164, https://doi.org/10.1002/jcc.21224.

[28] A. Mokhtari and A. Ribeiro, *RES: regularized stochastic BFGS algorithm*, IEEE Trans. Signal Process., 62 (2014), pp. 6089–6104, https://doi.org/10.1109/TSP.2014.2357775.

[29] A. Mokhtari and A. Ribeiro, *Global convergence of online limited memory BFGS*, J. Mach. Learn. Res., 16 (2015), pp. 3151–3181.

[30] P. Moritz, R. Nishihara, and M. Jordan, *A linearly-convergent stochastic L-BFGS algorithm*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, A. Gretton and C. C. Robert, eds., vol. 51 of Proceedings of Machine Learning Research, Cadiz, Spain, 09–11 May 2016, PMLR, pp. 249–258, http://proceedings.mlr.press/v51/moritz16.html.

[31] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statistics, 22 (1951), pp. 400–407, https://doi.org/10.1214/aoms/1177729586.

[32] F. Roosta-Khorasani and M. W. Mahoney, *Sub-sampled Newton methods*, Math. Program., 174 (2019), pp. 293–326, https://doi.org/10.1007/s10107-018-1346-5.

[33] D. Ruppert, *A Newton-Raphson version of the multivariate Robbins-Monro procedure*, Ann. Statist., 13 (1985), pp. 236–245, https://doi.org/10.1214/aos/1176346589.

[34] J. C. Spall, *A second order stochastic approximation algorithm using only function measurements*, in Proceedings of 1994 33rd IEEE Conference on Decision and Control, vol. 3, 1994, pp. 2472–2477.

[35] J. C. Spall, *Stochastic version of second-order (Newton-Raphson) optimization using only function measurements*, in Proceedings of the 27th Conference on Winter Simulation, WSC '95, USA, 1995, IEEE Computer Society, p. 347–352, https://doi.org/10.1145/224401.224633.

[36] J. C. Spall, *Accelerated second-order stochastic optimization using only function measurements*, in Proceedings of the 36th IEEE Conference on Decision and Control, vol. 2, 1997, pp. 1417–1424.

[37] J. C. Spall, *Introduction to stochastic search and optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience [John Wiley & Sons], Hobo-

ken, NJ, 2003, https://doi.org/10.1002/0471722138. Estimation, simulation, and control.

[38] D. Vicari, A. Okada, G. Ragozini, and C. Weihs, eds., *Analysis and Modeling of Complex Data in Behavioral and Social Sciences*, Springer, Cham, 2014, https://doi.org/10.1007/978-3-319-06692-9.

[39] Z. Xu and Y.-H. Dai, *New stochastic approximation algorithms with adaptive step sizes*, Optim. Lett., 6 (2012), pp. 1831–1846, https://doi.org/10.1007/s11590-011-0380-5.

[40] F. Yousefian, A. Nedić, and U. V. Shanbhag, *On stochastic gradient and subgradient methods with adaptive steplength sequences*, Automatica J. IFAC, 48 (2012), pp. 56–67, https://doi.org/10.1016/j.automatica.2011.09.043.