Descent direction method with line search for unconstrained optimization in noisy environment

Nataša Krejić* Zorana Lužanin* Zoran Ovcin †
Irena Stojkovska‡

December 10, 2014

Abstract

A two-phase descent direction method for unconstrained stochastic optimization problem is proposed. A line search method with an arbitrary descent direction is used to determine the step sizes during the initial phase, and the second phase performs the stochastic approximation (SA) step sizes. The almost sure convergence of the proposed method is established, under standard assumption for descent direction and SA methods. The algorithm used for practical implementation combines a line search quasi-Newton method, in particular the BFGS and SR1 methods, with the SA iterations. Numerical results show good performance of the proposed method for different noise levels.

Key words. stochastic optimization, stochastic approximation, noisy function, descent direction method, line-search, quasi-Newton methods, unconstrained minimization

AMS subject classification. 90C15, 62L20, 60H40, 65K05, 90C53

1 Introduction

The main objective of this paper is to propose and discuss a new method based on combination of ideas from deterministic and stochastic optimization for the following unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

*Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia, e-mail: {natasak@uns.ac.rs, zorana@dmi.uns.ac.rs}

†Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, e-mail: zovcin@uns.ac.rs

‡Department of Mathematics, Faculty of Natural Sciences and Mathematics, Ss. Cyril and Methodius University, Arhimedova 3, 1000 Skopje, Macedonia, e-mail: irenatra@pmf.ukim.mk

where $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable, **possibly nonconvex** function bounded below on D. Throughout the paper we assume that only noisy measurements of the objective function $f(x)$ and gradient $\nabla f(x) = g(x)$ are available at $x \in D$. For $x \in D$, let $\xi(x)$ and $\varepsilon(x)$ be random variable and random vector respectively defined on a probability space $(\Omega, \mathcal{F}, P)$. Then, the noisy functional and gradient values at each $x \in D$ are

$$F(x) = f(x) + \xi(x) \quad \text{and} \quad G(x) = g(x) + \varepsilon(x), \tag{2}$$

where ξ and ε represent the random noise terms. Note that the noise terms show dependence on x as this property is relevant for many applications **[24], [27],[28].** We assume that there is a unique x^* that solves (1) and then the first order condition states that x^* satisfies

$$g(x^*) = 0. \tag{3}$$

Typical examples of such problems arise in many application areas, as noise is present whenever physical system measurements or computer simulations are used for approximation. The presence of noise might mislead an optimization algorithm throughout the entire process and result in false optimal solutions. Some of the results regarding optimization problems in noisy environment are given in [13, 22].

One of the well known methods for solving problem (1) in presence of noise is stochastic approximation (SA) **by** Robbins and Monro (1951) [20]. The first results for this method dealt with the asymptotic analysis and the conditions have been obtained under which the convergence in mean square is guaranteed. Later, an almost sure convergence is established. However, SA is rather slow in practice and that motivated further research in order to accelerate the convergence in practical implementations, such as accelerated SA procedures presented in [9, 14, 25]. In the absence of noise the SA reduces to the well know descent direction method from deterministic optimization. Therefore, one natural idea is to mimic the line-search procedure that leads to linear convergence in deterministic case. This idea is developed in [29, 30]. But the convergence conditions require increase in the sample size used for approximation of the objective function and the gradient throughout the iterative procedure, thus resulting in an expensive method. Other ideas coming from deterministic optimization are considered in the literature. For example, Spall developed a method which uses a second order approximation of the Hessian matrix obtained by finite differences and thus generates a quasi-Newton sequence, [24]. Other modifications of the SA regarding step size and/or search direction selection are given in [2], [32], [33], [34]. **A direct line search method based on probabilistic descent is analyzed in [1]. An interesting attempt of the opposite kind, an extension of stochastic based methods to deterministic problems, is presented in [12] where the probabilistic methods are considered in the trust region framework.**

A successful combination of the SA and line-search gradient method is proposed and analyzed in [15]. A two-phase method consisting of the negative

gradient line search procedure during the initial phase of the iterative procedure and the SA steps afterwards, is proven to be almost surely convergent and rather efficient. The key point is the choice of switching point from the initial phase to the SA phase. The objective of the work presented in this paper is to extend this approach to general descent direction methods and thus allow the application of faster, second order methods while keeping the almost sure convergence. The application of second order methods appears to be particularly important within the framework of large scale problems arising in machine learning where an SA quasi Newton approach is successfully applied, [3, 4, 5]. In the algorithm presented in this paper a line search is applied while we are far away from a solution and hopefully the obtained reduction in gradient value is a consequence of reduction in the exact gradient. Therefore it is reasonable to expect that the method will generate large steps and make fast progress. Once we approach some small neighborhood of a solution line-search step is not good anymore since the noise is influencing functional and gradient values much stronger. Therefore we switch to the safe but slow SA steps. Again, the main issue is the switching point from line-search to SA. The almost sure convergence of the proposed method will be established. The convergence conditions do not require increase in the sample size, which is an important property in practical implementation.

The paper is organized as follows. A brief overview of the SA and line search descent direction methods is given in Section 2. The algorithm is presented in Section 3. The convergence results presented in Section 4 consists of two parts. First we prove that the SA methods defined by any descent direction in the sense specified below is almost surely convergent. Then we prove that the switching point between the SA and the line search method is well defined and that the algorithm generates a sequence which is almost surely convergent. Section 5 contains numerical results that confirm the efficiency of the approach proposed here. The issue of an appropriate implementation for quasi-Newton directions in stochastic environment is also addressed.

2 Line-search versus SA in stochastic environment

The iterative sequence generated by SA (stochastic approximation) for solving the problem (1) is defined by

$$x_{k+1} = x_k - a_k G_k, \tag{4}$$

where G_k is an approximation of $g(x_k) = \nabla f(x_k)$, and $a_k > 0$ are the gain coefficients. This method is introduced by Robbins and Monro [20] and is known as the Robbins-Monro Stochastic Approximation (RMSA).

The almost sure convergence of SA **for strongly convex objective func-**

tion can be proved if the gain coefficients a_k satisfy

$$a_k > 0,\ \sum_{k=0}^{\infty} a_k = \infty,\ \sum_{k=1}^{\infty} a_k^2 < \infty. \tag{5}$$

The convergence in mean square is proved in [20], i.e. $x_k \to x^*$ in m.s., that is $E[\|x_k - x^*\|^2] \to 0$ as $k \to \infty$. Other authors like Chen in [7] and Spall in [25] proved the almost sure convergence i.e. $x_k \to x^*$ a.s.

Let $\{x_k\}$ be a sequence generated by an SA method (4)-(5). Denote by $\mathcal{F}_k$ the σ-algebra generated by $x_0, x_1, \ldots, x_k$. The set of standard assumptions under which SA is convergent consist of the following.

A1 For any $\varepsilon > 0$ there is $\beta_\varepsilon > 0$ such that

$$\inf_{||x-x^*||>\varepsilon} (x - x^*)^T g(x) = \beta_\varepsilon > 0.$$

A2 The observation noise $(\varepsilon_k(x), \mathcal{F}_{k+1})$ is a martingale difference sequence with

$$E(\varepsilon_k(x)|\mathcal{F}_k) = 0 \text{ and } E||\varepsilon_k(x)||^2 < \infty \text{ a.s for all } k \text{ and } x \in D.$$

A3 The gradient g and the conditional second moment of the observation noise have the following upper bound

$$||g(x)||^2 + E(||\varepsilon_k(x)||^2|\mathcal{F}_k) < c(1 + ||x - x^*||^2) \text{ a.s. for all } k \text{ and } x \in D,$$

where $c > 0$ is a positive constant.

Assumption A1 represents the condition on the shape of $g(x)$, A2 is the standard mean-zero noise condition and A3 provides restrictions on the magnitude of $g(x)$, saying that $\|g(x)\|^2$ and the conditional second moment of the observation noise can not grow faster then a quadratic function of x.

In [7] it is proved that under assumptions A1-A3 and condition (5) for the gain sequence $\{a_k\}$, the sequence $\{x_k\}$ generated by SA method (4) converges a.s. to the solution x^* of the nonlinear system

$$g(x) = 0 \tag{6}$$

in noisy environment. Having in mind properties of the loss function f from the problem (1), we can conclude that assumptions A1-A3 and condition (5) for the gain sequence $\{a_k\}$, provide a.s. convergence of the SA method to the solution x^* of the problem (1). **Given that we are dealing with a method that combines SA and line search several other assumptions will be introduced later on. At this moment let us only briefly explain them. The conditions on search direction d_k determined by SA properties are stated in the assumptions A4-A5. In fact in the descent SA method A4-A5 are used instead of A1 as the true gradient g is unavailable.**

The assumptions A4-A5 limit the influence of noise and connect the descent direction with the noisy gradient. On the other hand line search requires several common assumptions like the Lipschitz continuity of the gradient and certain relationship between the gradient and the search direction, see A6, A8-A9. Finally, we are going to assume that the observation noise is bounded, A7. Detailed assumptions are presented in Section 4.

For practical implementation of SA, choice of the gain coefficients a_k plays the main role. The most simple choice is $a_k = a/(k+1)$, with $a > 0$. The optimal choice in the sense of asymptotic convergence is

$$a_k = \frac{\|H(x^*)^{-1}\|}{k+1}$$

where H is the Hessian matrix of f, while the most practical choice is

$$a_k = \frac{a}{(k+1+A)^\alpha},$$

where $a, \alpha > 0$ and $A \geq 0$ are parameters. A detailed discussion of parameter tuning is presented in [23, 24]. Non-asymptotic properties of SA method are the main focus of interest in practical applications. Unfortunately the asymptotically optimal methods behave badly in finite time: the choice $a_k = a/(k+1)$ is too "cautious" if the disturbance term is small with respect to the initial error $x_0 - x^*$. Although reliable, the method is quite slow and therefore expensive for practical purposes. These drawbacks prompted a number of modifications of SA method. Several generalizations of the SA method are based on adaptive step sizes that try to adjust the step size at each iteration to the progress achieved in the previous iterations, see [9, 14, 33]. An important choice for the gain sequence is a constant sequence. Although such sequences do not satisfy (5) and almost sure convergence to solution can not be obtained, it can be shown that a constant step size can conduct the iterations to a region that contains the solution. This result initiated development of a cascading step-length SA scheme in [31] where a fixed step size is used until some neighborhood of the solution is reached.

Let us now briefly introduce the Armijo line search procedure that will be incorporated into noisy environment. For details concerning line search methods one can consult [18]. Given an objective function f and a descent direction d_k line search procedure determines the step length a_k such that the Armijo condition

$$f(x_k + a_k d_k) \leq f(x_k) + c_1 a_k g(x_k)^T d_k, \tag{7}$$

where c_1 is a small positive constant, is satisfied. The implementation within the framework we consider here includes the need to modify the rule according to the fact that only the noisy values of the objective function and its gradient are available.

3 Descent Stochastic Line Search Algorithm

The presence of noise makes the definition of descent direction ambiguous, given that only noisy observations of the gradient are available and the condition $g(x_k)^T d_k < 0$ can not be checked. Furthermore, the negative noisy gradient might not be the best descend direction. One alternative possibility is considered in [34] where CG method is employed to find a better direction. On the other hand it is well known that second order descent directions, like Newton's and quasi-Newton directions, speed up convergence in deterministic problems. But capturing second order information in stochastic problems is a difficult task. Quasi-Newton approach, more precisely BFGS method is considered in [26] and [24] among others while [5] deals with the BFGS method in a slightly different problem of machine learning programs.

The definition of descent direction we use in this paper is based on the available gradient values. We say that d_k is a descent direction at x_k if

$$G_k^T d_k < 0, \tag{8}$$

where G_k is the approximation of the gradient defined by

$$G_k = G(x_k) = g(x_k) + \varepsilon_k(x_k).$$

The only available function values are noisy,

$$F_k = F_k(x_k) = f(x_k) + \xi_k(x_k).$$

Thus, the line search rule for accepting the step size a_k will be

$$F_k(x_k + a_k d_k) \leq F_k + c_1 a_k G_k^T d_k, \tag{9}$$

where d_k is a descent search direction at x_k.

We consider a combination of two methods here - a descent direction with line search at the beginning of the iterative procedure and the safe SA step sizes at the final stages of the procedure. Determining the appropriate switching point will be the main challenge in the implementation of this approach.

Roughly speaking, we will assume that the relative error in gradient estimation

$$r = \frac{g(x_k) - G_k}{\|g(x_k)\|}$$

is reasonably small, $\|r\| \approx 0$ and while $\|G_k\| > C$ we believe that we are far away from a solution. Therefore the noise should not have dominant influence in the gradient and functional values decrease and the stochastic line search (9) mimics deterministic behavior. In other words we expect it to generate large steps and approach a solution's neighborhood fast. Once we reach some small neighborhood of the solution, say when $\|x - x^*\| \leq \epsilon$ there is no more reason to believe that the line search (9) generates good steps. In fact the Armijo-type condition (9) does not mean real progress since the noise might have significant influence

on the estimates $F_k, F_k(x_k + a_k d_k), G_k$. Furthermore, **the estimates in consecutive iterations are calculated using different** samples and therefore, d_k obtained with one sample set is not likely to be "strong" enough descent direction so the Armijo-type condition is not meaningful anymore. At that moment we switch to the safe but slow SA method with descent direction and the step sizes satisfying (5).

The algorithm for the Descent Stochastic Line Search (DSLS) method is given as follows.

Algorithm DSLS

Input parameters: $x_0 \in \mathbb{R}^n$, $c_1 \in (0,1)$, $C, \underline{\delta}(C) > 0$ and $\{a_k\} \in \mathbb{R}$ such that (5) holds. Set $k = 0, phase = 1$.

Step 1 Take d_k such that (8) holds.

Step 2 Select a step size a_k.

Step 3 Define the new iterate $x_{k+1} = x_k + a_k d_k$.

Step 4 Set $k = k + 1$ and go to Step 1.

Clearly, Step 2 will determine the behavior of the algorithm and need to be specified. We apply the line search rule (9) as long as the gradient estimate is large enough. After that we switch to the predefined SA steps, as specified below.

Step size selection

Step 2

Step 2.1 If $phase = 1$ go to Step 2.2. Else go to Step 2.3

Step 2.2 If $\|G_k\| \geq C$ choose $a_k > \underline{\delta}(C)$ such that the inequality (9) is satisfied. Go to Step 3. Else set $phase = 2$.

Step 2.3 Take a_k from the predefined gain sequence.

4 Convergence results

Let us first establish the convergence of descent direction method with SA step sizes i.e. for a given x_k the next iterate x_{k+1} is

$$x_{k+1} = x_k + a_k d_k, \; k = 0, 1, \ldots, \tag{10}$$

where d_k is a descent direction defined by (8) and the gain sequence $\{a_k\}$ satisfies conditions (5). Similar problem is studied by Bertsekas and Tsitsiklis [2], where the iterative sequence is obtained through

$$x_{k+1} = x_k + a_k(s_k + w_k).$$

Here, w_k **is either deterministic error bounded by** $\|w_k\| \leq a_k(p + q\|g(x_k)\|)$ **for some positive scalars** p **and** q**, or is stochastic error with**

zero mean and satisfying $E[\|w_k\|^2|\mathcal{F}_k] \leq A(1+\|g(x_k)\|^2)$, **for some** $A > 0$. The convergence conditions include the assumption that s_k is gradient related in the sense that

$$c_1\|g(x_k)\|^2 \leq -g(x_k)^T s_k,\ \|s_k\| \leq c_2(1+\|g(x_k)\|).$$

The result we prove below claims a.s. convergence under the set of assumptions that are closely related to the SA standard set of assumptions, A1-A3.

Let $\{x_k\}$ be a sequence generated by (10). For any $k \geq 0$, denote with $\mathcal{F}_k$ the σ-algebra generated by iterations $x_0, x_1, ..., x_k$. The following assumptions A4-A5 are used instead of assumption A1, and together with assumptions A2-A3 provide the almost sure convergence of the descent direction method (10) with the SA step sizes.

A4 Direction d_k satisfies the following inequality

$$(x_k - x^*)^T E[d_k|\mathcal{F}_k] \leq -c_2\|x_k - x^*\| \text{ a.s. for all } k,$$

where $c_2 > 0$ is a constant.

A5 There exists $c_3 > 0$ such that for all k we have

$$\|d_k\| \leq c_3\|G_k\| \text{ a.s.}$$

The assumption A4 limits the influence of noise on d_k and is analogous to Assumption C4 used in [24]. If we take $d_k = -G_k$, then A2 implies that A4 is satisfied if there is a constant $c_2 > 0$ such that $(x_k - x^*)^T g_k \geq c_2\|x_k - x^*\|$ a.s. for all k. On the other hand the assumption A5 connects the available noisy gradient and descent direction. Clearly, taking $d_k = -G_k$ we get that A5 is satisfied with any $c_3 \geq 1$.

The following result from [21] is needed for the convergence proof below.

Theorem 4.1 *[21] If U_n, β_n, ξ_n and ζ_n, $n = 1, 2, ...$ are nonnegative $\mathcal{F}_n$-measurable random variables such that*

$$E(U_{n+1}|\mathcal{F}_n) \leq (1+\beta_n)U_n + \xi_n - \zeta_n, \quad n = 1, 2, ...$$

then on the set $\left\{\sum_n \beta_n < +\infty, \sum_n \xi_n < +\infty\right\}$, U_n converges a.s. to a random variable and $\sum_n \zeta_n < +\infty$ a.s.

Now, we can state and prove the following convergence theorem for the descent direction method with the SA step sizes (10).

Theorem 4.2 *Let A2-A5 hold and $\{x_k\}$ be a sequence generated by (10). Then $x_k \to x^*$ a.s.*

Proof. From $x_{k+1} = x_k + a_k d_k$ and assumptions A4 and A5 we have

$$\begin{aligned} E[\|x_{k+1} - x^*\|^2|\mathcal{F}_k] &= E[\|x_k + a_k d_k - x^*\|^2|\mathcal{F}_k] \\ &= \|x_k - x^*\|^2 + 2a_k(x_k - x^*)^T E[d_k|\mathcal{F}_k] + a_k^2 E[\|d_k\|^2|\mathcal{F}_k] \\ &\leq \|x_k - x^*\|^2 - 2c_2 a_k \|x_k - x^*\| + c_3^2 a_k^2 E[\|G_k\|^2|\mathcal{F}_k] \end{aligned}$$

Given that $G_k = g_k + \varepsilon_k$ assumptions A2 and A3 imply

$$\begin{aligned} E[\|G_k\|^2|\mathcal{F}_k] &= E[\|g_k + \varepsilon_k\|^2|\mathcal{F}_k] \\ &= \|g_k\|^2 + 2g_k^T E[\varepsilon_k|\mathcal{F}_k] + E[\|\varepsilon_k\|^2|\mathcal{F}_k] \\ &= \|g_k\|^2 + E[\|\varepsilon_k\|^2|\mathcal{F}_k] \\ &< c(1 + \|x_k - x^*\|^2) \end{aligned}$$

So,

$$\begin{aligned} E[\|x_{k+1} - x^*\|^2|\mathcal{F}_k] &\leq \|x_k - x^*\|^2 - 2c_2 a_k \|x_k - x^*\| + c_3^2 a_k^2 E[\|G_k\|^2|\mathcal{F}_k] \\ &< \|x_k - x^*\|^2 - 2c_2 a_k \|x_k - x^*\| + cc_3^2 a_k^2 (1 + \|x_k - x^*\|^2) \\ &= (1 + cc_3^2 a_k^2)\|x_k - x^*\|^2 - 2c_2 a_k \|x_k - x^*\| + cc_3^2 a_k^2 \end{aligned}$$

Denote by $U_n = \|x_n - x^*\|^2$, $\beta_n = cc_3^2 a_n^2$, $\xi_n = cc_3^2 a_n^2$ and $\zeta_n = 2c_2 a_n \|x_n - x^*\|$, **then** $\sum_k a_k^2 = \infty$ **imply** $\sum_n \beta_n < \infty$ **and** $\sum_n \xi_n < \infty$. **Theorem 4.1 yields that** U_n **converges a.s. to a random variable and** $\sum_n \zeta_n < \infty$. **Thus** $a_n \|x_n - x^*\| \to 0$ **a.s. Given that** $\sum_n a_n = \infty$ **there follows**

$$\|x_k - x^*\| \to 0 \text{ a.s.}$$

Consequently $x_k \to x^*$ a.s. ∎

Now, let $\{x_k\}$ be a sequence generated by Algorithm DSLS. For any $k \geq 0$, denote by $\mathcal{F}_k$ the σ-algebra generated by iterations $x_0, x_1, ..., x_k$.

The convergence conditions for line-search method include the Lipschitz condition on the gradient of objective function. Therefore the assumption A6 is necessary for the convergence of the proposed DSLS method.

A6 The gradient g is Lipschitz continuous, that is there exists a positive constant L such that

$$\|g(x) - g(y)\| \leq L\|x - y\| \quad \text{for all} \quad x, y \in \mathbb{R}^n.$$

An additional assumption which bounds the realized noise is needed. Notice that this assumption does not imply any restriction on real problems.

A7 Observation noises are bounded and there exists a positive constant M such that

$$\|\xi_k(x)\| \leq M, \ \|\varepsilon_k(x)\| \leq M \text{ a.s.}$$

for all k and $x \in D$.

Given that the DSLS method is based on descent direction line search approach, two additional assumptions A8-A9, common to the descent direction method in deterministic optimization [11], are necessary.

A8 There exists $\delta > 0$ such that for all k we have

$$G_k^T d_k \leq -\delta \|G_k\| \|d_k\| \text{ a.s.}$$

A9 There exists $\underline{\Delta} \in (0, \Delta)$ such that for all k we have

$$\|d_k\| \geq \underline{\Delta} \text{ a.s.}$$

The first theorem below shows that Algorithm DSLS is well defined and the second theorem shows that Algorithm DSLS generates finitely many step sizes that satisfy the line search rule (9). Their proofs are conceptually similar to the corresponding proofs in [15] with some technical differences. We include the proofs here for the sake of completeness.

Theorem 4.3 *Suppose that A5-A8 hold. Let*

$$C \geq \frac{M + 2\sqrt{2ML} + 1}{\delta(1 - c_1)}.$$

Then there exists $\underline{\delta}(C) > 0$ a.s. such that Algorithm DSLS is well defined.

Proof. Denote by f_k and g_k the objective function and gradient values at $x = x_k$ respectively. Let $\alpha > 0$ and $d \in \mathbb{R}^n$ be arbitrary. Then

$$\begin{aligned} f(x_k + \alpha d) &= f_k + \alpha g(x_k + t\alpha d)^T d \\ &= f_k + \alpha g(x_k + t\alpha d)^T d + \alpha g_k^T d - \alpha g_k^T d \\ &= f_k + \alpha g_k^T d + \alpha (g(x_k + t\alpha d) - g_k)^T d \\ &\leq f_k + \alpha g_k^T d + \alpha \|g(x_k + t\alpha d) - g_k\| \cdot \|d\| \end{aligned}$$

for some $t \in (0, 1)$. The assumption A6 and $t \in (0, 1)$ imply

$$f(x_k + \alpha d) \leq f_k + \alpha g_k^T d + \alpha^2 L \|d\|^2.$$

Since f and g are subject to noise we have

$$F_k(x_k + \alpha d_k) = f(x_k + \alpha d_k) + \tilde{\xi}_k,$$

$$F_k = f_k + \xi_k \quad \text{and} \quad G_k = g_k + \varepsilon_k,$$

where the abbreviation $\tilde{\xi}_k = \xi_k(x_k + \alpha d_k)$ is used.

Taking $d = d_k$ we have

$$\begin{aligned} F(x_k + \alpha d_k) &= f(x_k + \alpha d_k) + \tilde{\xi}_k \\ &\leq f_k + \alpha g_k^T d_k + \alpha^2 L \|d_k\|^2 + \tilde{\xi}_k \\ &= F_k + \alpha G_k^T d_k - \alpha \varepsilon_k^T d_k + \alpha^2 L \|d_k\|^2 + \tilde{\xi}_k - \xi_k \\ &\leq F_k + \alpha G_k^T d_k + \alpha M \|d_k\| + \alpha^2 L \|d_k\|^2 + 2M. \text{a.s.} \end{aligned} \tag{11}$$

So we need to show that a.s. there exists $\underline{\delta}(C) > 0$ and $\overline{\alpha} > \underline{\delta}(C)$ such that for $\alpha \in (\underline{\delta}(C), \overline{\alpha})$ we have

$$F_k + \alpha G_k^T d_k + \alpha M \|d_k\| + \alpha^2 L \|d_k\|^2 + 2M \leq F_k + c_1 \alpha G_k^T d_k, \qquad (12)$$

which is equivalent to

$$\alpha(1 - c_1) G_k^T d_k + \alpha M \|d_k\| + \alpha^2 L \|d_k\|^2 + 2M \leq 0.$$

The assumption A8 implies

$$\alpha(1 - c_1) G_k^T d_k + \alpha M \|d_k\| + \alpha^2 L \|d_k\|^2 + 2M$$
$$\leq \alpha^2 L \|d_k\|^2 - \delta\alpha(1 - c_1)\|G_k\|\|d_k\| + \alpha M \|d_k\| + 2M,$$

so the statement will be proved analyzing the quadratic function

$$\phi(\alpha) = \alpha^2 L \|d_k\|^2 - \delta\alpha(1 - c_1)\|G_k\|\|d_k\| + \alpha M \|d_k\| + 2M.$$

Defining

$$A_\phi = L\|d_k\|^2 > 0,$$
$$B_\phi = -\delta(1 - c_1)\|G_k\|\|d_k\| + M\|d_k\| < 0,$$
$$C_\phi = 2M > 0,$$

we want to prove that $B_\phi^2 - 4A_\phi C_\phi > 0$. That can be done analyzing the zeroes of another quadratic function

$$\psi(u) = \delta^2(1 - c_1)^2 u^2 - 2\delta M(1 - c_1)u + M^2 - 8ML.$$

There exist $u_1, u_2 \in \mathbb{R}$, $u_1 < u_2$, $\psi(u_1) = \psi(u_2) = 0$, and $\psi(u) > 0$ for $u > u_2$, where

$$u_2 = \frac{2\delta M(1 - c_1) + \sqrt{32\delta^2(1 - c_1)^2 ML}}{2\delta^2(1 - c_1)^2} = \frac{M + 2\sqrt{2ML}}{\delta(1 - c_1)}.$$

As $\|G_k\| \geq C > u_2$ we conclude that $B_\phi^2 - 4A_\phi C_\phi > 0$ is fulfilled and hence the function $\phi(\alpha)$ has two real zeros

$$\alpha_1 = \frac{-B_\phi - \sqrt{B_\phi^2 - 4A_\phi C_\phi}}{2A_\phi} \text{ and } \alpha_2 = \frac{-B_\phi + \sqrt{B_\phi^2 - 4A_\phi C_\phi}}{2A_\phi}$$

and the statement (12) is true for any $\alpha \in (\alpha_1, \alpha_2)$. Let us also note that $0 < \alpha_1 < -\frac{B_\phi}{2A_\phi} < \alpha_2$.

Now, let us show that (12) is fulfilled for α uniformly bounded from below. It is sufficient to find a lower bound $\underline{\alpha} > 0$ independent of k such that $-\frac{B_\phi}{2A_\phi} \geq \underline{\alpha}$.

From the assumption A5 we have

$$-\frac{B_\phi}{2A_\phi} = \frac{\delta(1 - c_1)\|G_k\| - M}{2L\|d_k\|} \geq \frac{\delta(1 - c_1)\|G_k\| - M}{2Lc_3\|G_k\|}.$$

Since $\|G_k\| \geq C > 0$,

$$\frac{\delta(1-c_1)\|G_k\| - M}{2Lc_3\|G_k\|} = \frac{\delta(1-c_1) - M/\|G_k\|}{2Lc_3} \geq \frac{\delta(1-c_1) - M/C}{2Lc_3},$$

and from $C \geq \frac{M+2\sqrt{2ML}+1}{\delta(1-c_1)}$ we have

$$\frac{\delta(1-c_1) - M/C}{2Lc_3} \geq \frac{\delta(1-c_1)(2\sqrt{2ML}+1)}{2Lc_3(M+2\sqrt{2ML}+1)}.$$

thus we can take

$$\underline{\alpha} = \frac{\delta(1-c_1)(2\sqrt{2ML}+1)}{2Lc_3(M+2\sqrt{2ML}+1)}.$$

So we can conclude that for $\underline{\delta}(C) = \max\{\alpha_1, \underline{\alpha}\}$ we can a.s. take $a_k \in (\underline{\delta}(C), \alpha_2)$ such that (12) is valid and a_k is uniformly bounded from below a.s. ■

Let us now prove that Algorithm DSLS eventually ends up with the SA steps.

Theorem 4.4 *Let A6-A9 hold and take*

$$C \geq \max\{\frac{2M+1}{\underline{\alpha}c_1\delta\underline{\Delta}}, \frac{M+2\sqrt{2ML}+1}{\delta(1-c_1)}\},$$

$$\underline{\alpha} = \frac{\delta(1-c_1)(2\sqrt{2ML}+1)}{2Lc_3(M+2\sqrt{2ML}+1)}$$

Let $\{x_k\}$ be an infinite sequence generated by Algorithm DSLS. Let $\{x_j\}, j \in J$ be a subsequence such that

$$\|G_j\| \geq C. \tag{13}$$

Then $\{x_j\}$ is finite a.s

Proof. Let us assume the contrary i.e. the sequence $\{x_j\}$ is infinite. If (13) is satisfied then we have that

$$\|G_j\| \geq \frac{M+2\sqrt{2ML}+1}{\delta(1-c_1)}$$

and Theorem 4.3 implies that for any $j \in J$ the next iterative point x_{j+1} is obtained a.s. by the line-search rule such that

$$F_j(x_{j+1}) \leq F_j(x_j) + c_1 a_j G_j^T d_j,\ x_{j+1} = x_j + a_j d_j,\ a_j > \underline{\delta}(C) \geq \underline{\alpha}.$$

Further more all the previous points are also obtained a.s. by the line-search rule and thus

$$F_i(x_{i+1}) \leq F_i(x_i) + c_1 a_i G_i^T d_i,\ a_i > \underline{\alpha},\ i = 0, 1, \ldots, j.$$

Since

$$F_i(x_{i+1}) = f(x_{i+1}) + \tilde{\xi}_i,\ F_i(x_i) = f(x_i) + \xi_i,$$

$\xi_i - \tilde{\xi}_i \leq 2M$ a.s., $a_i > \underline{\alpha}$, $\|G_i\| \geq C$ we have

$$\begin{aligned} f(x_{i+1}) &\leq f(x_i) + c_1 a_i G_i^T d_i + \xi_i - \tilde{\xi}_i \\ &< f(x_i) - c_1 \delta \underline{\alpha} C \underline{\Delta} + 2M \quad \text{a.s.} \end{aligned} \tag{14}$$

Take $K = c_1 \delta \underline{\alpha} C \underline{\Delta} - 2M > 0$. Clearly

$$f(x_{i+1}) < f(x_i) - K,\ i = 0, 1, 2, \ldots, j.$$

Summing up the above inequalities for an arbitrary $j \in J$, we have

$$\sum_{i=0}^{j} f(x_{i+1}) < \sum_{i=0}^{j} f(x_i) - \sum_{i=0}^{j} K$$

and

$$f(x_{j+1}) - f(x_0) < -\sum_{i=0}^{j} K,$$

which implies

$$\sum_{i=0}^{j} K < f(x_0) - f(x_{j+1}) \leq K_1$$

since the function f is bounded from below. As $K > 0$, we have

$$j + 1 < K_1/K = K_2.$$

for arbitrary $j \in J$. This is contrary to the assumption that the sequence $\{x_j\}$, $j \in J$ is infinite so the statement is proved. ■

Now, we can state the main convergence theorem for DSLS method based on the previous theorems.

Theorem 4.5 *Suppose that assumptions A2-A9 hold. Let*

$$C \geq \max\{\frac{2M+1}{\underline{\alpha} c_1 \delta \underline{\Delta}}, \frac{M + 2\sqrt{2ML} + 1}{\delta(1 - c_1)}\},$$

$$\underline{\alpha} = \frac{\delta(1 - c_1)(2\sqrt{2ML} + 1)}{2Lc_3(M + 2\sqrt{2ML} + 1)}$$

Let $\{x_k\}$ be an infinite sequence generated by Algorithm DSLS. Then x_k converges a.s. to x^*.

Proof. From Theorem 4.4 we have that almost surely there are only finitely many step sizes that satisfy the stochastic line search rule (9). So, Algorithm DSLS generates infinitely many successive step sizes that satisfy conditions (5) and due to Theorem 4.2 we have that $x_k \to x^*$ a.s. ■

5 Numerical Experiments

We are interested here in computational implementation of Algorithm DSLS assuming that the problem we are solving is (1) and that the noisy values of the objective function and the gradient are calculated as the sample average approximation. In other words for given i.i.d. samples $\{\xi_k\}$ and $\{\varepsilon_k\}$ we assume that

$$F_k = F(x_k) = \frac{1}{p}\sum_{i=1}^{p} F(x_k, \xi_{k^i}),\ G_k = \nabla F(x_k, \varepsilon_{k^i})$$

where p is a reasonably small number and the samples used in each iteration are independent.

5.1 Computational Implementation

In practical implementation of Algorithm DSLS several issues need to be specified. First of all, the convergence results proved in the previous Section depend on the constant C that actually determines the switching point between the two phases of Algorithm DSLS. Given that C depends on the Lipschitz constant L and the realized noise bound M, it is clear that in general we can not estimate C with reasonable precision. The second issue we are interested in is the implementation of a second-order direction, mainly an efficient implementation of Quasi-Newton directions in noisy environment. Finally the SA gain sequence can be defined in many ways and the behavior of the algorithm DSLS depends on that sequence as well. The implementation we tested is sketched as a pseudocode in Procedure DSLS below.

As common in stochastic optimization the main exit criteria for the algorithm is the budget for function evaluation. This exit criteria motivates the definition of switching point. The classical line search procedure based on interpolation, [18, 10] is applied within each iteration with a limited number of trial step lengths. If the number of maximal trial step lengths is exhausted without a satisfactory step length we conclude that the line search is not successful any more i.e. that the switching point is reached and the method switches to the SA step lengths. The pseudo code is listed in Procedure LS below. The input and output parameters are discussed later on.

The SA step lengths we use are

$$a_k = \frac{a}{(k+1+A)^{\alpha}}, \tag{15}$$

where parameters are fine tuned to $a = 1$, $A = 0$, $\alpha = 0.602$, as in [25].

The remaining question is the implementation of Quasi-Newton directions. We consider two directions, BFGS and SR1 obtained as

$$d_k = -B_k^{-1} G_k.$$

Thus for given x_k, x_{k+1} the **BFGS** update is

$$B_{k+1} = B_k - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} + \frac{\Delta_k \Delta_k^T}{\Delta_k^T \delta_k}, \tag{16}$$

where

$$\delta_k = x_{k+1} - x_k,\ \Delta_k = G_{k+1} - G_k,$$

and **the SR1 update is**

$$B_{k+1} = B_k + \frac{(\Delta_k - B_k\delta_k)(\Delta_k - B_k\delta_k)^T}{(\Delta_k - B_k\delta_k)^T\delta_k}. \tag{17}$$

Both of the considered QN directions are known for their good performance and exhibit super linear convergence if implemented with a suitable line search. However, in the noisy environment several issues might arise. The most important is perhaps the calculation of Δ_k. If we take

$$\Delta_k = G(x_{k+1}, \varepsilon_{k+1}) - G(x_k, \varepsilon_k)$$

then the noise terms are interfering with the difference in gradients and might yield very inaccurate and unstable information. One possible remedy is proposed in [24] where the update is taken with weights that limit the influence of noise. Other possibility that we adopted here is successfully tested in [26]. The main point is to compute the gradient difference Δ_k using the same sample set. Such computation of Δ_k clearly doubles the number of gradient calculations but the experiments we performed clearly indicated that the additional computation is well spent. Thus in both updates, BFGS and SR1 we compute

$$\Delta_k = G(x_{k+1}, \varepsilon_k) - G(x_k, \varepsilon_k). \tag{18}$$

The pseudo code of the update procedure is given in **Procedures** updateBFGS and updateSR1.

The input parameters of Procedure DSLS are the initial values x_0, F_0, G_0 denote by x, F_x, G_x respectively and three parameters: $itnlimit, maxfcalcls$ and $maxfcalc$. These parameters are the maximal number of iterations, the maximal number of trial step lengths within the line search procedure and the budget for function evaluation within the whole iterative procedure. As already mentioned, $maxfcalcls$ is in fact used to determine the switching point. The parameter *near* in DSLS procedure is $false$ while we try to find the step length by the line search procedure. Once it takes the value $true$ the algorithm switches to the SA step lengths. In the notation of Algorithm DSLS, this corresponds to setting $phase = 2$ in Step 2.2. The exit parameters are the final iteration x_{end} and the function and gradient values F_{end} and G_{end}, denoted by y, F_y and G_y, as well are the termination code *termcode*. The termination code stops the algorithms as follows:

- $termcode = 1$ if the gradient value is small enough, $\|G_k\| \leq gradtol$;
- $termcode = 2$ if the maximal number of function evaluations is reached.

Thus the algorithm stops with x_{end} either if we reach a stationary point or if the maximal number of function evaluations is used.

Procedure: DSLS

Input: $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \to \mathbb{R}$, $G : \mathbb{R}^n \to \mathbb{R}^n$, $itnlimit \in \mathbb{Z}$, $maxfcalcls \in \mathbb{Z}$, $maxfcalc \in \mathbb{Z}$

Output: $y \in \mathbb{R}^n$, $F_y \in \mathbb{R}$, $G_y \in \mathbb{R}^n$, $termcode \in \mathbb{Z}$

initiate_variables;

$termcode := 0$; $[G_x, z_x] := G(x)$; `% remember the noise` z_x
$near := false$; `% when testing SA steps put` $near := true$ `here`

if $\neg near$ **then** $F_x := F(x)$;
for $itncount := 1$ **to** $itnlimit$ **do**

$d :=$ find_direction;

if $\neg near$ **then**

$[y, F_y, retcode, \lambda, fcalcls] := \mathrm{LS}(x, F_x, F(\cdot), G_x, d, maxfcalcls)$;

if *retcode = 2* **then** `%` $fcalcls \geq maxfcalcls$`, switch to SA`
$near := true$;
$\lambda := a/(itncount + 1 + A)^\alpha$;
else if *retcode = 1* **then** `% too small step`
$\lambda := \lambda_{\mathrm{minfix}}$;
end

else
$\lambda := a/(itncount + 1 + A)^\alpha$;
end

$y := x + \lambda * d$;

$[G_y, z_y] := G(y)$; `% find the gradient and remember the noise`

if $|G_y| \leq gradtol$ **then**
$termcode := 1$; **return**;
else if $number\_of\_function\_calculations \geq maxfcalc$ **then**
$termcode := 2$; **return**;
else
update_quasi_Newton; `% Not needed with GRAD directions`
$x := y$; $G_x := G_y$; $z_x := z_y$; `% prepare for next iteration`
if $\neg near$ **then** $F_x := F_y$;
end
end
return;

The line search procedure input parameters are the current values x_k, F_k, G_k, d_k denoted by x, F, G, d, the function $F(\cdot, \xi_{k+1})$ denoted by $\hat{F}$ and the parameters $maxstep, steptol$ and $maxfcalcls$. The role of $maxfcalcls$ is already explained, $maxstep$ is the maximal allowed step length and $steptol$ is a scaling parameter. The output parameters are the new iteration $y = x_{k+1}$ and the objective function value $F_y = F_{k+1}$, the parameter $retcode$, the step size $\lambda = a_k$ and the number of function evaluations used in the line search procedure $fcalcls$. The parameter $retcode$ can have the following values:

- $retcode = 0$ - x_{k+1} satisfying the Armijo condition is found
- $retcode = 1$ - unable to find x_{k+1} satisfying Armijo condition and sufficiently distinct from x_k
- $retcode = 2$ - maximal allowed number of function evaluations reached not giving x_{k+1} that satisfies Armijo condition

The case $retcode = 1$ might be a consequence of the noise. In that case we continue with the iteration regardless of the Armijo condition taking the predefined minimal step size λ_{minfix} in Procedure DSLS.

Both QN directions are implemented with the difference in gradients calculated using (18). In the absence of noise BFGS Hessian approximation is generating descent directions and has the so-called self-correction property. In other words, BFGS tends to correct bad approximations of the Hessian and it is reasonable to expect that after one bad approximation the method itself will correct the update, if combined with well defined line search. An additional care is necessary if the noise is present as we may face singularity due to a particularly bad sample. Therefore we skip the update if $|\Delta_k \delta_k| < safeguard$ in order to prevent the accumulation of errors due to small values of $\Delta_k \delta_k$. The $safeguard$ parameter is initialized within Procedure initiate_variables as $eps^{1/4}$ where eps is the machine precision. The another variable to be initiated within this procedure is the initial QN approximation taken as the identity matrix.

5.2 Numerical Results

There are two questions we are interested in here. First one is to test the efficiency of the algorithm proposed here i.e. to test the efficiency of the line search at the initial phase of the iterative methods. Thus, we compare the methods defined by Algorithm DSLS and the SA method (10) with the same direction. The second question is the relationship between the negative gradient direction and second order directions, in particular the BFGS and SR1 directions implemented as described above. So, we are reporting the results for 6 different methods. In all SA methods the gain sequence is given by (15) as well as in the second phase of LS methods.

SA-G - the SA method (10) with $d_k = -G_k$.

LS-G - the method defined by Algorithm DSLS and $d_k = -G_k$;

Procedure: LS

Input: $x \in \mathbb{R}^n$, $F_x \in \mathbb{R}$, $F : \mathbb{R}^n \to \mathbb{R}$, $G_x \in \mathbb{R}^n$, $d \in \mathbb{R}^n$, $maxstep \in \mathbb{R}$, $steptol \in \mathbb{R}$, $maxfcalcls \in \mathbb{Z}$

Output: $y \in \mathbb{R}^n$, $F_y \in \mathbb{R}$, $retcode \in \mathbb{Z}$, $\lambda \in \mathbb{R}^+$, $fcalcls \in \mathbb{Z}$

if $|d| > maxstep$ **then** $d := d * maxstep/|d|$;

$initslope := G^T\, d$;

$rellength := \max_i \dfrac{|d[i]|}{\max(x[i], 1)}$; $\lambda_{\min} := \dfrac{steptol}{rellength}$;

$retcode := 3$; $fcalcls := 0$; $\alpha := 1E-4$; $\lambda := 1$;

while $retcode > 2$ **do**

  $y := x + \lambda\, d$;

  $F_y := F(y)$; `% calculated using new` p `size sample`

  $fcalcls := fcalcls + p$;

  if $y \le x + \alpha\, \lambda\, initslope$ **then** `% satisfactory` λ

    $retcode := 0$;

  else if $\lambda < \lambda_{\min}$ **then** `% too small` λ

    $retcode := 1$; $y := x$; $F_y := F_x$;

  else if $fcalcls \ge maxfcalcls$ **then** `%` $maxfcalcls$ `reached`

    $retcode := 2$;

    $y := x + \lambda\, d$; `% returned value` y `is not valid now`

  else `% we iterate`

    if $\lambda = 1$ **then** `% quadratic fit`

      $\lambda_{temp} := -\frac{initslope}{2(F_y - F_x - initslope)}$;

    else `% cubic fit`

$$\begin{bmatrix} a \\ b \end{bmatrix} := \frac{1}{\lambda - \lambda_{prev}} \begin{bmatrix} \frac{1}{\lambda^2} & \frac{-1}{\lambda_{prev}^2} \\ \frac{-\lambda_{prev}}{\lambda^2} & \frac{\lambda}{\lambda_{prev}^2} \end{bmatrix} * \begin{bmatrix} F_y - F_x - \lambda\, initslope \\ F_{prev} - F_x - \lambda_{prev}\, initslope \end{bmatrix};$$

      $disc := b^2 - 3\, a\, initslope$;

      if $a = 0$ **then** $\lambda_{temp} := -\frac{initslope}{2b}$;

      else $\lambda_{temp} := \frac{-b + \sqrt{disc}}{3a}$;

      if $\lambda_{temp} > 0.5\lambda$ **then** $\lambda_{temp} := 0.5\lambda$;

    end

    $\lambda_{prev} := \lambda$; $F_{prev} := F_y$;

    if $\lambda_{temp} \le 0.1\lambda$ **then** $\lambda := 0.1\lambda$;

    else $\lambda := \lambda_{temp}$;

  end

end

return;

Procedure: initiate_variables

$B := I$; % starting with the identity matrix
$safeguard := eps^{(1/4)}$; % treshold for QN singularity
$maxstep := \max(|x|, 1)$; % for LS
$steptol := eps^{(2/3)}$; % for $\lambda_{\min}$ in LS
return;

Procedure: updateBFGS

$G'_y := G(y, z_x)$; % find the gradient using old noise
$\delta := y - x$; $\Delta := G'_y - G_x$;
if $|\Delta^T \delta| \geq safeguard$ **then** % no division by zero
    if $itncount = 1$ **then** $B := \dfrac{\Delta\Delta^T}{\Delta^T\delta} I$;
    $B := B - \dfrac{B\delta\delta^T B}{\delta^T B\delta} + \dfrac{\Delta\Delta^T}{\Delta^T\delta}$;
end
return;

Procedure: updateSR1

$G'_y := G(y, z_x)$; % find the gradient using old noise
$\delta := y - x$; $\Delta := G'_y - G_x$;
if $|(\Delta - B\delta)^T \delta| \geq safeguard\, |\delta|\, |B\delta|$ **then** % no division by zero
    $B := B + \dfrac{(\Delta - B\delta)(\Delta - B\delta)^T}{(\Delta - B\delta)^T \delta}$;
end
return;

SA-BFGS - the SA method (10) with BFGS d_k;

LS-BFGS - the method defined by Algorithm DSLS and BFGS direction d_k;

SA-SR1 - the SA method (10) with SR1 d_k;

LS-SR1 - the method defined by Algorithm DSLS and SR1 direction d_k.

The test collection we considered consists of problems taken from the collection of J. Burkhardt, at `http://people.sc.fsu.edu/~jburkardt/m_src/test_opt/test_opt.html`, [6] and four additional problems tested in [24], [16] and [19].

The collection in [6] consists of 43 problems, mainly described in [17], in the form of nonlinear least squares,

$$\min f(x) = \sum_{i=1}^{m} f_i^2(x).$$

The problems 4,10 and 11 are excluded as all tested algorithms failed on these three problems. The original problems are transformed into noisy ones adding the noise at the objective function and the gradient in the form

$$\xi(x) = u,\ u \sim \mathcal{N}(0, \sigma^2),\ \varepsilon(x) = v,\ v \sim \mathcal{N}(0, \sigma^2 I_{n\times n}),$$

where $I_{n\times n}$ is the identity matrix. The initial approximations are taken as in [6].

The additional four problems are given below.
Problem 1, [24]

$$f(x) = x^T A^T x + 0.1 \sum_{i=1}^{n} (Ax)_i^3 + 0.01 \sum_{i=1}^{n} (Ax)_i^4$$

where A is $n \times n$ upper triangular matrix of ones. The initial point is $x^0 = 0.2(1, 1, \ldots, 1)$ and the minimizer is $x^* = (0, 0, \ldots 0)$, $f(x^*) = 0$.
Problem 2, [16].

$$f(x) = 200\, \phi_b(x_1, x_2)/\phi_b(10, 10)$$

where

$$\phi_b(x_1, x_2) = \exp\left(0.1\sqrt{x_1^2 + bx_2^2}\right) + \exp\left(-0.1\sqrt{x_1^2 + bx_2^2}\right) - 2.$$

The initial point is $x^0 = 0.5(1, 1)$ and the minimizer is $x^* = (0, 0)$, $f(x^*) = 0$ $(b = 1)$.

The noise term for both Problems 1 and 2 is added as

$$\xi(x) = [x^T, 1]\ z,\ \text{where } z \sim \mathcal{N}(0, \sigma^2 I_{(n+1)\times(n+1)}),$$

$$\varepsilon(x) = z,\ \text{where } z \sim \mathcal{N}(0, \sigma^2 I_{n\times n}).$$

Problem 3, [19]

$$f(x) = \sum_{i=1}^{n}(e^{x_i} - x_i),\ x = (x_1, ..., x_n).$$

The initial point is $x_0 = (1/n, ..., i/n, ..., 1)$ and the minimiser is $x^* = (0, ..., 0), f^* = n$.
Problem 4, [19]

$$f(x) = \sum_{i=1}^{n} \frac{i}{10}(e^{x_i} - x_i), x = (x_1, ..., x_n).$$

The initial iteration is $x_0 = (1, ..., 1)$ and the minimiser is $x^* = (0, ..., 0),\ f^* = \frac{n(n+1)}{20}$.

Problems 3 and 4 are modified adding the noise like in the problems from [6],

$$\xi(x) = u,\ u \sim \mathcal{N}(0, \sigma^2),\ \varepsilon(x) = v,\ v \sim \mathcal{N}(0, \sigma^2 I_{n\times n}).$$

Different level of noise are tested and here we report results obtained for $\sigma = 1, \sigma = 0.2$ and $\sigma = 0.04$. In each experiment we made $N = 50$ independent runs starting from the same initial point. The *itnlimit* value set to ∞ and the maximal number of function evaluations is $maxfcalc = 200n$. The parameter $maxfcalcls = 4$ is used so the line search procedure contains at most 4 function evaluations. Each gradient evaluation is counted as n function evaluations. In all calculations we have used the sample size $p = 3$ for the sample average approximation of the objective function and the gradient.

The exit parameters for all algorithms are

$$gradtol = \min\{\sqrt{n}\sigma, 1\}$$

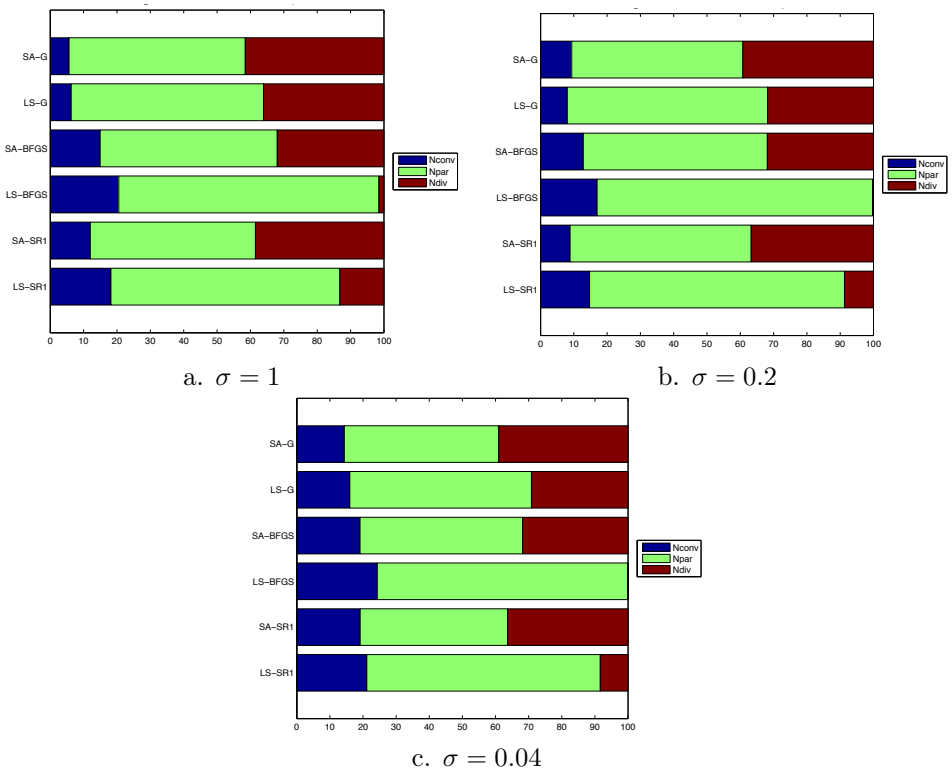
and the maximal number of function evaluations

$$maxfcalc = 200n.$$

If a method stops due to $\|G_{end}\| \leq gradtol$ we consider the run successful. The number of successful runs is denoted by $Nconv$. If

$$\|G_{end}\| > 200\sqrt{n}$$

we declare divergence. The number of divergent runs is denoted by $Ndiv$. Finally the runs which stopped due to reaching the maximal number of allowed function evaluations are considered partially successful and their number is denoted by $Npar$. Our conjecture is that in such cases $maxfcalc$ was not large enough to achieve convergence with the given $gradtol$ but the function values is nevertheless decreased so the algorithm made some progress. In total we have 2200 runs of each algorithm and we report the percentage of $Nconv, Npar$ and $Ndiv$ for all three noise levels at Figure 1.



a. $\sigma = 1$ b. $\sigma = 0.2$

c. $\sigma = 0.04$

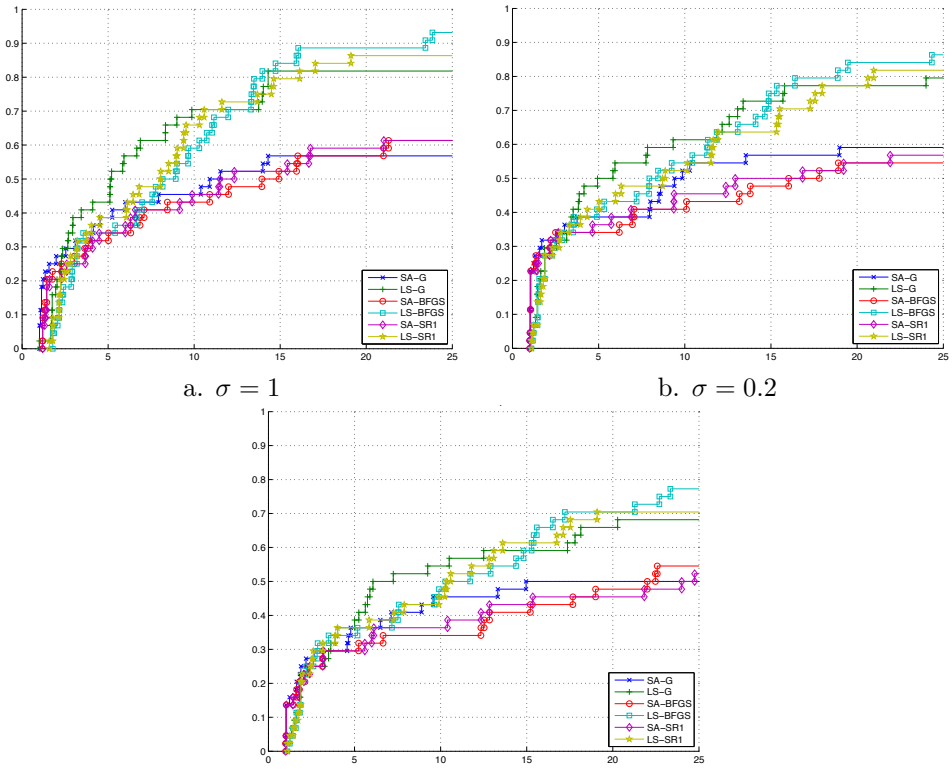
Figure 1: Percentage of successful, partially successful and divergent runs

The results shown at Figures 1 clearly demonstrate that Algorithm DSLS have significantly smaller number of divergent runs with respect to the corresponding SA methods regardless of the direction for all noise levels. Furthermore, the second order directions, BFGS and SR1 are significantly better than the negative gradient directions in both approaches, within the SA and DSDS algorithms. DSLS algorithm with the BFGS direction appears to be the most successful for all three noise levels.

At Figure 2 we report the performance profiles, [8]. The performance measure is the average number of function evaluations needed in convergent and partially successful runs normalized to include the problem dimension i.e.

$$fc_{ij} = \frac{1}{|Ncon_{ij} U Npar_{ij}|} \sum_{r \in Ncon_{ij} U Npar_{ij}} \frac{fcalc_{ij}^r}{n_j},$$

where $fcalc_{ij}^r$ is the the number of function evaluations spent by the method i to solve the problem j in r-th run, n_j is the dimension of the j-th problem, and $Ncon_{ij}$ and $Npar_{ij}$ are convergent and partially successful runs for the method i to solve the problem j respectively.



a. $\sigma = 1$ b. $\sigma = 0.2$

c. $\sigma = 0.04$

Figure 2: Performance profiles for all 6 methods

The performance profile clearly indicate clustering of SA methods and DSLS methods with DSLS being significantly better. The difference appears to be more obvious for larger values of σ. Among DSLS methods BFGS and SR1 perform better that the negative gradient method as expected so the second order information decreases the number of function evaluations. We can conclude that both considered measures, the number of function evaluations and the number of successful runs demonstrate the advantages of the algorithm proposed in this paper.

Acknolegements. We are grateful to the anonymous referees and the associate editor for their constructive comments which helped us to improve this paper.

References

[1] **A.S. Bandeira, K. Scheinberg, L.N. Vicente, Convergence of trust region methods based on probabilistic methods, SIAM Journal on Optimization, 24 (2014) 1238-1264.**

[2] D. P. Bertsekas, J. N. Tsitsiklis, Gradient convergence in gradient methods with errors, SIAM J. Optim., Vol. 10, No. 3 (2000), pp.627-642

[3] R. H. Byrd, G. M. Chin, W. Neveitt, J. Nocedal, On the Use of Stochastic Hessian Information in Optimization Methods for Machine Learning, SIAM J. Optim., 21 (3), (2011) pp. 977-995.

[4] R. H. Byrd, G. M. Chin, J. Nocedal, Y. Wu, Sample size selection in optimization methods for machine learning, Math. Program. 134(1), (2012) pp. 127-155.

[5] R. H. Byrd, S. L. Hansen, J. Nocedal, Y. Singer, A Stochastic Quasi-Newton Method for Large-Scale Optimization, *Technical report, arXiv:1401.7020 [math.OC].*

[6] J. Burkhardt, TEST_OPT, `http://people.sc.fsu.edu/~jburkardt/m_src/test_opt/test_opt.html`

[7] H.-F. Chen, Stochastic Approximation and Its Application, Kluwer Academic Publishers, New York, 2002

[8] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, Math. Program., Ser. A, Vol. 91 (2002), pp.201-213

[9] B. Delyon, A. Juditsky, *Accelerated stochastic approximation,* SIAM J. Optim., Vol. 3, No. 4 (1993), pp.868-881

[10] J. E. Dennis, R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[11] M. A. Diniz-Ehrhardt, J. M. Martinez, M. Raydan, A derivative free nonmonotone line-search technique for unconstrained optimization, J. Comp. Appl. Math., Vol.219, No.2 (2008), pp.389-397

[12] **S. Gratton, C.W. Royer, L.N. Vicente, Z. Zhang, Direct search based on probabilistic descent, preprint 11-14, Department of Mathematics, University of Coimbra.**

[13] H. T. Fang, H. F. Chen, Almost surely convergent global optimization algorirhm using noise-corrupted observations, J. Optim. Theory Appl., 104,2 (2000), pp.343-376

[14] H. Kesten, Accelerated stochastic approximation, Ann. Math. Stat., 29 (1958), pp.41-59

[15] N. Krejić Z. Lužanin, I. Stojkovska , A gradient method for unconstrained optimization in noisy environment, Appl. Numer. Math. 70 (2013) 1-21.

[16] M. N. Levy, M. W. Trosset, R. R. Kincaid, Quasi-Newton methods for stochastic optimization, Proceedings of the Fourth Symposium on Uncertainty Modeling and Analysis (ISUMA '03), 2003.

[17] J. J. Moré, B. S. Garbow, K. E. Hillstrom, Testing Unconstrained Optimization Software, ACM Trans. Math. Soft., Vol. 7, No. 1 (1981), pp.17-41

[18] J. Nocedal, S. J. Wright, Numerical Optimization, Springer-Verlag, New York, 1999

[19] M. Raydan, The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem. SIAM J. Optim. Vol.7 No.1 (1997), pp. 26-33

[20] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Statist., 22 (1951), pp.400-407

[21] H. Robbins, D. Siegmund, A convergence theorem for nonnegative almost supermatringales and some applications, Optimizing Methods in Statistics, Academic Press, New York (1971), pp.233-257

[22] K. Sirlantzis, J. D. Lamb, W. B. Liu, Novel algorithms for noisy minimization problems with applications to neural networks training, J. Optim. Theory Appl., Vol.129, No.2 (2006), pp.325-340

[23] J. C. Spall, An Overview of the Simultaneous Perturbation Method for Efficient Optimization, John Hopkins Applied Technical Digest, Vol. 19, No. 4 (1998), pp.482-492.

[24] J. C. Spall, Adaptive stochastic approximation by the simultaneous perturbation method, IEEE Trans. Autom. Contr., Vol. 45, No. 10 (2000), pp.1839-1853

[25] J. C. Spall, Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003

[26] N. N. Schraudolph, J.Yu, S. Günter, A Stochastic Quasi-Newton Method for Online Convex Optimization, Proceedings of 11th International Conference on Artificial Intelligence and Statistics (2007), 433-440.

[27] **H. Valpola, J. Karhunen, An Unsupervised Ensemble Learning Method for Nonlinear Dynamic State-Space Models, Neural Computation 14(11), (2002) 2647 - 2692.**

[28] **L. Venkataramanan, R. Kuc, F.J. Sigworth, Identification of Hidden Markov Models for Ion Channel Currents - Part II: State-Dependent Excess Noise, IEEE Transactions on Signal Processing 46,7 (1998), 1916-1929.**

[29] Y. Wardi, A stochastic steepest-decent algorithm, J. Optim. Theory Appl., Vol. 59, No. 2 (1988), pp.307-323

[30] Y. Wardi, Stochastic algorithms with Armijo stepsizes for minimization of functions, J. Optim. Theory and Appl., Vol. 64, No. 2 (1990), pp.399-417

[31] F. Yousefian, A. Nedic, U.V. Shanbhag, On stochastic gradient and subgradient methods with adaptive steplength sequences, Automatica 48 (1),2012, pp. 56-67.

[32] Zi Xu, Yu-H. Dai, A stochastic approximation frame algorith with adaptive directions, Numer. Math. Theor. Meth. Appl., Vol.1, No.4 (2008), pp.460-474

[33] Zi Xu, Yu-H. Dai, New stochastic approximation algorithms with adaptive spet sizes, Optim Lett., Vol.6, No.8 (2012), pp.1831-1846

[34] Zi Xu, A combined direction stochastic approximation algorithm, Optimization Letters, Vol.4, No.1 (2010), pp.117-129