A gradient method for unconstrained optimization in noisy environment

Nataša Krejić* Zorana Lužanin † Irena Stojkovska‡

January 22, 2013

Abstract

A gradient method for solving unconstrained minimization problems in noisy environment is proposed and analyzed. The method combines line-search technique with Stochastic Approximation (SA) method. A line-search along the negative gradient direction is applied while the iterates are far away from the solution and upon reaching some neighborhood of the solution the method switches to SA rule. The main issue is to determine the switching point and that is resolved both theoretically and practically. The main results is the almost sure convergence of the proposed method due to a finite number of line-search steps followed by infinitely many SA consecutive steps. The numerical results obtained on a set of standard test problems confirm theoretical expectations and demonstrate the efficiency of the method.

Key words. stochastic optimization, stochastic approximation, noisy function, gradient method, line-search method.

AMS subject classification. 90C15, 62L20, 60H40

1 Introduction

In this paper we consider the unconstrained minimization problem in noisy environment,

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

where $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function bounded below on D. Throughout the paper we will assume that there is a unique x^* that

*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. Research supported by Ministry of Education and Science of Serbia grant No. 174030.

†Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia, e-mail: zorana@dmi.uns.ac.rs. Research supported by Ministry of Education and Science of Serbia grant No. 174030.

‡Department of Mathematics, Faculty of Natural Sciences and Mathematics, St. Cyril and Methodius University, Gazi Baba b.b., 1000 Skopje, Macedonia, e-mail: irenatra@iunona.pmf.ukim.edu.mk

solves (1) and 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 component 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.

Noise is present whenever physical system measurements or computer simulations are used for approximation. A vast set of examples includes problems where estimates are formed by computer-based Monte Carlo sampling according to a statistical distribution, problems where data are collected while system is operating or problems where physical data are processed sequentially, with each sequential data point being used to estimate some overall (average) criterion, such as the mean-squared error (MSE). 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 [4, 18].

The notation we will use throughout the paper is

$$F_k = F_k(x_k) = f(x_k) + \xi_k(x_k), \quad G_k = G_k(x_k) = g(x_k) + \varepsilon_k(x_k) \tag{3}$$

The index k used with ε and ξ allow us to consider the noise depending on the current iteration x_k, *i.e.,* the noise-generating process may change with k.

One of the first stochastic optimization method is *Stochastic Approximation* (SA), also known as the *Robbins-Monro Stochastic Approximation* [16]. SA was modified by Kiefer and Wolfowitz [7] such that only noisy objective function measurements are used. That method is known as *Finite Difference Stochastic Approximation*. Originally, SA method was established for solving nonlinear algebraic systems, that is

$$g(x) = 0, \tag{4}$$

where $g : \mathbb{R}^n \to \mathbb{R}^n$. Mimicking the simplest gradient descent method and using only the noisy measurements of $g(x)$, SA algorithm generates an iterative sequence by the formula

$$x_{k+1} = x_k - a_k G_k, \tag{5}$$

where a_k is a nonnegative gain coefficient. The method is convergent if the gain sequence $\{a_k\}$ satisfies certain conditions. For solving (4) on a constrained set $X \subseteq \mathbb{R}^n$ one can use SA with projection. Starting from a chosen point $x_0 \in X$, the iterates are obtained by the formula

$$x_{k+1} = \Pi_X\big(x_k - a_k G_k\big), \tag{6}$$

where Π_X is a projection operator that maps iterates outside the constrained set X back to X.

Convergence of SA method in stochastic sense can be achieved under suitable conditions. In the pioneer work of Robbins and Monro [16], the convergence in mean square was proved, *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 [2] and Spall in [20] proved the almost sure convergence, *i.e.,* $x_k \to x^*$ a.s..

The performance of SA method depends mostly on the choice of the gain sequence $\{a_k\}$. The best-known example of the gain sequence is the scaled harmonic sequence $a_k = a/(k+1)$, where $a > 0$. A common generalization of the scaled harmonic sequence is $a_k = a/(k+1)^\alpha$, where $a > 0$, and $1/2 < \alpha \leq 1$. The gain coefficients are designed to yield convergence in noisy environment but their values tend to make the iterative process quite slow. Roughly speaking the step size is proportional to $1/k$ so the steps become small very fast and result in slow progress. That was the main reason for a number of modifications proposed in the literature. In particular, SA is often implemented in stochastic simulation methods as a modified SA method or as a final step of some stochastic method when convergence in stochastic sense is needed, see Delyon and Juditsky [3], Kesten [6], Kushner and Gavin [9], Monnez [12], Ruppert [17], Spall [19], Wei [24].

One of the first adaptive techniques, called *Accelerated Stochastic Approximation* is given by Kesten [6], for one dimensional problems. An adaptive gain sequence a_k definition is based on the frequency of sign changes of the differences $x_{k+1} - x_k$. Frequent sign changes are an indication that the current iterate is near the optimal solution x^*, and if the signs are not changing, than the iterate is far from x^*. A larger gain coefficient a_k is used if there are no sign changes and a smaller a_k is used if the signs change frequently. Kesten established a.s. convergence of the above accelerated SA to x^*, [6]. Further extensions of Kesten's accelerated SA were given by Kushner and Gavin [9], and by Delyon and Juditsky [3]. Further modifications are the subject of many studies. Andradótirr in [1] considered the scaled algorithm. In each iteration of the scaled algorithm two independent gradient estimates are sampled at the current iterate x_k to compute a scale-free estimate of the next iterate x_{k+1}. Gradient methods with the Armijo line-search rule are considered by Wardi [23]. A Quasi-Newton method with line-search is proposed by Kao, Chen [5]. To avoid misjudgment of the minimal point due to its stochastic nature, a t-test is performed instead of a simple comparison of the mean responses. Additional attempts to consider line-search in the context of SA methods are made in [10, 21, 22].

Methods that are SA analogues of the Newton-Raphson method replace the scalar gain a_k with a matrix that approximates the inverse Hessian. Some methods of this type are given by Ruppert [17], Wei [24], and Monnez [12]. One survey of stochastic approximation methods is given in Kushner [8].

Given that the performance of the gradient and other descent direction methods in classical optimization problems can be significantly improved if a line-search technique is applied, a natural question is wether the use of line-search techniques to determine the step size, in place of a fixed gain sequence, can be beneficial or not. The noise prevents direct application of line-search techniques, in particular when we are close to the solution. Steps determined by the

line-search rule tend to be too large and cause zig-zag behavior or even lead the iterative sequence out of the solution's neighborhood. However, when we are far away from the solution, large steps generated by a line-search rule are desirable in comparison with relatively small step generated by SA. So a natural way of exploiting the good properties of both approaches, SA and line-search, would be to combine them in such a way that a line-search rule is used at the initial stages of the optimization process and SA is applied afterwards. The key question is how to determine the switching point between the line-search procedure and SA. Given that the first-order optimality conditions are $\nabla f(x) = 0$, but only the noisy observation of the first derivative are available, the following reasoning is intuitively clear. If the observed (noisy) gradient components values G_k are large, we are far away from the stationary point and the decrease of the (noisy) functional value indicates a decrease of the real objective function f, *i.e.,*, the decrease is not only due to the noise, so it is safe to use the (large) line-search step size. But if the gradient components values are small we are probably close to the solution and hence we should switch to the safe gain coefficients of the SA procedure, which minimizes the influence of noise.

The main objective of this paper is to analyze and implement the above described procedure. We will show that a line-search procedure is called in a finite number of iterations, which ensures the almost sure convergence of proposed methods due to infinitely many SA steps. At the same time the convergence of the described combined method is significantly faster then the convergence of SA.

The paper is organized as follows. Section 2 contains the notation and an overview of known results regarding SA and line-search that will be used throughout the paper. The gradient-based algorithm is defined and analyzed in Section 3. Practical implementation issues are discussed in Section 4. In the same section the results of numerical experiments which demonstrate the efficiency of the proposed algorithm are given, while some conclusions are drawn in Section 5.

2 Preliminaries

SA method is the key ingredient in our consideration and for a given initial approximation x_0 it can be written as

$$x_{k+1} = x_k - a_k G_k,\ k = 0, 1, \ldots. \tag{7}$$

The standard convergence conditions for the gain sequence $\{a_k\}$ are

$$a_k > 0,\ \sum_k a_k = \infty \text{ and } \sum_k a_k^2 < \infty. \tag{8}$$

The choice of the gain coefficients is analyzed in many papers because they determine the rate of convergence (see Spall [20] for details).

Let $\{x_k\}$ be a sequence generated by an SA method. Denote by $\mathcal{F}_k$ the σ-algebra generated by $x_0, x_1, \ldots, x_k$. The norm $||\cdot||$ refers to the Euclidean

norm. The set of standard assumptions under which SA is convergent consist of the following.

A1 For any $\varepsilon > 0$ there exists $\beta_\varepsilon > 0$ 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

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

where $\{\mathcal{F}_k\}$ is a family of nondecreasing σ-algebras.

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

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

where $c > 0$ is a constant.

Assumption A1 represents the condition on the shape of $g(x)$, A2 is the standard zero-mean 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 cannot grow faster then a quadratic function of x.

In [2] it is proved that under assumptions A1-A3, the sequence $\{x_k\}$ generated by SA method (5) converges a.s. to the solution x^* of the nonlinear system (4) in noisy environment.

Theorem 2.1. *Assume that A1-A3 hold. Let $\{x_k\}$ be a sequence generated by SA method*

$$x_{k+1} = x_k - a_k G_k$$

with the gain sequence $\{a_k\}$ satisfying (8) and assume that the Hessian of f is nonsingular at the solution x^*. Then the sequence $\{x_k\}$ converges to x^* for an arbitrary initial approximation x_0.

Line-search methods are powerful tools for globalization of descent-direction methods of unconstrained optimization. The main idea is to use a descent search direction and to determine the step size in such a way that enough decrease of the objective function is achieved. There are many rules for determining the step size in line-search procedure and for details of these methods one can look at [14]. In this paper we will consider the Armijo rule.

The method in noisy environment that mimics the Armijo rule assumes that we work with noisy observations of functional and gradient values. More precisely, the sufficient decrease condition we consider is governed by

$$F_k(x_k - a_k G_k) \leq F_k - c_1 a_k \|G_k\|^2, \tag{9}$$

where c_1 is a small positive constant.

The convergence conditions for line-search method include the Lipschitz condition on the gradient of the objective function. Therefore we state one additional assumption that will be necessary for the convergence.

A4 The gradient g is Lipschitz continuous, that is there exists a positive constant $L > 0$ such that

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

3 The gradient method

The algorithm we are proposing consists of two methods: line-search and SA. The basic idea is to take (larger) line-search steps at the beginning of the iterative procedure and switch to (smaller) SA steps when we approach the solution. The measure of proximity is the observed gradient vector. We will show that there exists a constant C such that we can safely apply the line-search at the kth iteration if

$$\|G_k\| \geq C. \tag{10}$$

The algorithm switches to SA if (10) is violated. This way we will preserve good properties of both methods, namely relatively large steps and fast progress of the gradient line-search methods and safe convergence of SA method. So the algorithm is defined as follows.

ALGORITHM 1. *GSLS method*

Step 0. Choose an initial point $x_0 \in \mathbb{R}^n$, constants $c_1 \in (0, 1)$, $C, \delta(C) > 0$, and a nonnegative SA gain sequence $\{a_k\}$ that satisfies (8). Set $k = 0$ and $p = 1$.

Step 1. Calculate the noisy gradient G_k.

Step 2. If $p = 1$ then calculate the noisy function F_k and go to Step 3, else go to Step 4.

Step 3. If $\|G_k\| \geq C$ choose $\alpha > \delta(C)$ such that the inequality

$$F_k(x_k - \alpha G_k) \leq F_k - c_1\alpha\|G_k\|^2 \tag{11}$$

is satisfied, set $a_k = \alpha$ and go to Step 5.

Else set $p = 2$.

Step 4. Take a_k from the predefined SA gain sequence.

Step 5. Define $x_{k+1} = x_k - a_k G_k$, set $k = k + 1$ and go to Step 1.

In practical implementation of GSLS Algorithm, at the kth iteration, the backtracking line-search procedure is applied in Step 3. If it fails within a predefined number of attempts then the algorithm switches to SA. Further details of practical implementation are discussed in Section 4.

Convergence analysis of Algorithm GSLS (Gradient Stochastic Line Search) consists of two parts. First we will show that the Algorithm is well defined, *i.e.,* that almost surely there exist constants C and $\delta(C)$ such that Step 3 will be performed if $\|G_k\| \geq C$. After that we will show that the line-search step is almost surely executed a finite number of times so the algorithm inevitably switches to SA and thus the almost sure convergence is obtained. The additional assumption stated below bounds the realized noise and does not imply any restriction on real problems. It is quite similar to the one used in Wardi [23]

A5 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 $x \in D$.

Theorem 3.1. *Suppose that A4-A5 hold. Let*

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

Then there exists $\delta(C) > 0$ a.s. such that the Algorithm 1 is well defined.

Proof. We will prove that almost surely there exist $\delta(C) > 0$ and $\alpha > \delta(C)$ such that (11) is satisfied whenever $\|G_k\| \geq C$.

Denote by f_k and g_k the objective function and gradient values at $x = x_k$ respectively. Let $\alpha > 0$ and $d \in R^n$ 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 $t \in (0, 1)$. Assumption A4 and $t \in (0, 1)$ imply

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

Taking $d = -G_k$ we have

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

Since f and g are subject to noise we have

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

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

where the abbreviation $\tilde{\xi}_k = \xi_k(x_k - \alpha G_k)$ is used. Furthermore

$$\begin{aligned} F(x_k - \alpha G_k) &= f(x_k - \alpha G_k) + \tilde{\xi}_k \qquad (12) \\ &\leq f_k - \alpha g_k^T G_k + \alpha^2 L \|G_k\|^2 + \tilde{\xi}_k \\ &= F_k - \xi_k - \alpha(G_k - \varepsilon_k)^T G_k + \alpha^2 L \|G_k\|^2 + \tilde{\xi}_k \\ &\leq F_k - \alpha\|G_k\|^2 + \alpha\|\varepsilon_k\|\|G_k\| + \alpha^2 L\|G_k\|^2 + \tilde{\xi}_k - \xi_k \\ &\leq F_k - \alpha\|G_k\|^2 + \alpha M\|G_k\| + \alpha^2 L\|G_k\|^2 + 2M, \text{a.s.} \qquad (13) \end{aligned}$$

with $\tilde{\xi}_k - \xi_k \leq 2M$ a.s. and $\|\varepsilon_k\| \leq M$ a.s..

We are looking for $\delta(C) > 0$ and $\alpha > \delta(C)$ such that

$$F(x_k - \alpha G_k) \leq F_k - c_1\alpha\|G_k\|^2$$

and therefore we need to prove that a.s. there exist $\delta(C) > 0$ and an upper bound $\bar{\alpha}$ to be specified later, $\bar{\alpha} > \delta(C)$, such that for $\alpha \in (\delta(C), \bar{\alpha})$ we have

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

This is equivalent to

$$0 \leq -\alpha^2 L\|G_k\|^2 + \alpha\|G_k\|^2 - \alpha M\|G_k\| - c_1\alpha\|G_k\|^2 - 2M. \qquad (14)$$

Let us first prove that there exists $\overline{\alpha} > 0$ such that (14) is satisfied for $\alpha \in (0, \overline{\alpha})$ a.s. and then we will prove that α can be uniformly bounded from below. Let

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

If $\phi(\alpha_1) = \phi(\alpha_2) = 0$ then $\phi(\alpha) \geq 0$ for $\alpha \in [\alpha_1, \alpha_2]$. So (14) will be valid for $\alpha \in (\alpha_1, \alpha_2)$ if we can prove that $\alpha_1 \neq \alpha_2$ and $\alpha_2 > 0$. With notation

$$A_\phi = -L\|G_k\|^2,$$

$$B_\phi = (1 - c_1)\|G_k\|^2 - M\|G_k\|,$$

$$C_\phi = -2M,$$

we have $B_\phi > 0$ due to $\|G_k\| > M/(1 - c_1)$ as $\|G_k\| \geq C$. Now

$$B_\phi^2 - 4A_\phi C_\phi = [(1 - c_1)\|G_k\|^2 - M\|G_k\|]^2 - 8ML\|G_k\|^2$$

so $B_\phi^2 - 4A_\phi C_\phi > 0$ is equal to

$$(1 - c_1)^2\|G_k\|^4 - 2M(1 - c_1)\|G_k\|^3 + M^2\|G_k\|^2 - 8ML\|G_k\|^2 > 0$$

and equivalent to

$$(1 - c_1)^2\|G_k\|^2 - 2M(1 - c_1)\|G_k\| + M^2 - 8ML > 0. \qquad (15)$$

Let us consider

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

If $\psi(u_1) = \psi(u_2) = 0$ then $\psi(u) > 0$ for $u > u_2$. As

$$M^2(1-c_1)^2 - (1-c_1)^2(M^2 - 8ML) = 8(1-c_1)^2ML > 0$$

we have that $u_1, u_2 \in \mathbb{R},\ u_1 \neq u_2$ and

$$u_2 = \frac{M(1-c_1) + \sqrt{8(1-c_1)^2ML}}{(1-c_1)^2},$$

so

$$u_2 = \frac{M + 2\sqrt{2ML}}{1-c_1}.$$

As $\|G_k\| \geq C > u_2$ we conclude that (15) is fulfilled and hence the function $\phi(\alpha)$ has two real zeroes α_1 and α_2, given by

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

Furthermore $\alpha_2 > 0$ (due to $B_\phi > 0$ and $A_\phi < 0$), so (14) is true for $\alpha \in (\alpha_1, \alpha_2)$.

On the other hand $4A_\phi C_\phi > 0$ and $B_\phi^2 - 4A_\phi C_\phi > 0$ implies that $0 < \alpha_1 < -\frac{B_\phi}{2A_\phi} < \alpha_2$. In order to show that (14) is fulfilled for α that are uniformly bounded from bellow, it is sufficient to find a lower bound $\underline{\alpha} > 0$ that is independent of k such that $-\frac{B_\phi}{2A_\phi} \geq \underline{\alpha}$.

Since $\|G_k\| \geq C > 0$ we have that

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

Having in mind the condition $C \geq \frac{M+2\sqrt{2ML}+1}{1-c_1}$ we have

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

So for the lower bound that we are looking for we can take

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

Thus, we have demonstrated that (14) holds true for $\alpha \in (\alpha_1, \alpha_2)$, where $0 < \alpha_1 < -\frac{B_\phi}{2A_\phi} < \alpha_2$, and $-\frac{B_\phi}{2A_\phi} \geq \underline{\alpha}$. So we can conclude that for $\delta(C) = \max\{\alpha_1, \underline{\alpha}\}$ there exists $\alpha \in (\delta(C), \alpha_2)$ such that (14) is valid and α is uniformly bounded from below a.s.. ■

The previous Theorem shows that when the iterate is "far" from the solution (characterized by $\|G_k\| \geq C$), then Step 3 of GSLS Algorithm will be executed almost surely. The next theorem shows that this step is called almost surely only a finite number of times.

Theorem 3.2. *Suppose that assumptions A4-A5 hold. Let*

$$C \geq \max\left\{\frac{4(1-c_1)}{\underline{\alpha}c_1}, \frac{M+2\sqrt{2ML}+1}{1-c_1}\right\},$$

where

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

Let $\{x_k\}$ be an infinite sequence generated by Algorithm 1 and $\{x_j\}, j \in J$ be a subsequence such that

$$\|G_j\| \geq C. \tag{16}$$

Then J is finite a.s..

Proof. Let us assume the contrary, *i.e.,* the sequence $\{x_j\}$ is infinite. If (16) is satisfied, then we have that

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

and Theorem 3.1 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

$$F_j(x_{j+1}) \leq F_j - c_1 a_j \|G_j\|^2,\ x_{j+1} = x_j - a_j G_j,\ a_j > \delta(C) \geq \underline{\alpha}.$$

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

$$F_i(x_{i+1}) \leq F_i - c_1 a_i \|G_i\|^2,\ a_i > \underline{\alpha},\ i = 0, 1, \ldots, j. \text{ a.s..}$$

Since $\xi_i(x_i) - \xi_i(x_{i+1}) \leq 2M$, a.s. $\|G_i\| \geq C$, $a_i > \underline{\alpha}$, $M < (1-c_1)\|G_i\|$, we have

$$\begin{aligned} f(x_{i+1}) &\leq f(x_i) - c_1 a_i \|G_i\|^2 + \xi_i(x_i) - \xi_i(x_{i+1}) \\ &\leq f(x_i) - c_1 a_i \|G_i\|^2 + 2M \\ &\leq f(x_i) - c_1 a_i C \|G_i\| + 2M \\ &< f(x_i) - c_1 \underline{\alpha} C \|G_i\| + 2M \\ &< f(x_i) - c_1 \underline{\alpha} C \|G_i\| + 2(1-c_1)\|G_i\| \\ &= f(x_i) - (c_1 \underline{\alpha} C - 2(1-c_1))\|G_i\| \text{ a.s..} \end{aligned} \tag{17}$$

Let $K = c_1 \underline{\alpha} C - 2(1-c_1)$. Since

$$C \geq \frac{4(1-c_1)}{\underline{\alpha}c_1} > \frac{2(1-c_1)}{\underline{\alpha}c_1},$$

we have that

$$K = c_1 \underline{\alpha} C - 2(1 - c_1) > 0.$$

Now (17) implies

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

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

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

and

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

so

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

for some $K_1 > 0$, as f is bounded from below. As $K > 0$, we have

$$\sum_{i=0}^{j} \|G_i\| < K_1/K = K_2,$$

and $K_2 > 0$. On the other hand, $\|G_i\| \geq C$ and

$$\sum_{i=0}^{j} \|G_i\| \geq \sum_{i=0}^{j} C = (j+1)C.$$

Combining the last two inequalities we obtain

$$(j+1)C < K_2$$

for arbitrary $j \in J$. But J is an infinite subsequence by assumption, so there exists $j_0 \in J$ such that $(j_0+1)C \geq K_2$. This contradiction proves the statement. ■

Now we can state the main convergence result for GSLS method based on the previous Theorems and SA convergence.

Corollary 3.1. *Suppose that assumptions A1-A5 hold and that the Hessian matrix $\nabla f(x^*)$ exists and is nonsingular. Let*

$$C \geq \max\left\{ \frac{4(1-c_1)}{\underline{\alpha} c_1}, \frac{M + 2\sqrt{2ML} + 1}{1 - c_1} \right\},$$

where

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

Let $\{x_k\}$ be an infinite sequence generated by Algorithm 1. Then x_k convergenes to x^* a.s..

Proof. From Theorem 3.2 we have that there are finitely many iterates that are obtained by Step 3 of Algorithm 1. So, infinitely many successive iterates are obtained by Step 4 and Theorem 2.1 implies the statement. ∎

4 Numerical results

The algorithm proposed in this paper is tested and compared with SA using a collection of test problems in the form

$$\min_{x\in\mathbb{R}^n} f(x) = \sum_{i=1}^{m} f_i^2(x).$$

The set of 10 problems from [13] and [15] is selected. The problems are transformed to noisy problems using the rule suggested in Sirlantzis *et al.* [18],

$$F(x,\omega) = f(x) + \omega, \quad G_i(x,\omega) = (\nabla f(x))_i + \omega, \quad i = 1,\dots,n.$$

We assume that the noise ω is normally distributed with zero expectation and variance σ, *i.e.,* $w \sim N(0,\sigma^2)$. Normal distribution noise clearly does not satisfy A5 because one could have $\|\xi(x_k)\|$ and $\|\varepsilon(x_k)\|$ arbitrary large with positive probability. However that does not complicate the implementation of the algorithm in practice due to the small variance we use and thus the samples are in fact bounded. The same kind of reasoning is present in [11]. Each calculation of the functional and gradient values is performed by using an independent sample realization of the noise. Analogously to [5] and [19], we calculate the noisy functional and gradient values using the arithmetic mean with sample size p :

$$F(x) = \frac{\sum_{i=1}^{p} F(x,\omega_i)}{p}, \quad G(x) = \frac{\sum_{i=1}^{p} G(x,\omega_i)}{p}$$

All tests are performed with $p = 3$ and two values for the variance, $\sigma = 0.1, 0.01$.

For each of the problems we give the parameters n and m, functions f_i, initial approximation x_0, solution x^*, if available and the optimal value $f^* = f(x^*)$. If x^* is not known, we take f^* from [13]. Besides these data we also list $||x_0 - x^*||^2$ and $|f(x_0) - f(x^*)|^2$.

Problem 1. [13] Biggs EXP6 function; $n = 6$, $m = 13$

$$f_i(x) = x_3 e^{-t_i x_1} - x_4 e^{-t_i x_2} + x_6 e^{-t_i x_5} - y_i,$$
$$t_i = 0.1i, \;\; y_i = e^{-t_i} - 5e^{-10t_i} + 3e^{-4t_i}$$

$x_0 = (10,10,1,1,10,1)$; $x^* = (1,10,1,5,4,3)$; $f^* = 0$; $||x_0 - x^*||^2 = 137$; $|f(x_0) - f(x^*)|^2 = 74.3672$

Problem 2. [13] Gaussian function; $n = 3$, $m = 15$

$$f_i(x) = x_1 \exp(\tfrac{-x_2(t_i - x_3)^2}{2}) - y_i, \text{ and } t_i = (8 - i)/2,$$

$$y_1 = y_{15} = 0.0009,\ y_2 = y_{14} = 0.004,\ y_3 = y_{13} = 0.0175,\ y_4 = y_{12} = 0.0540,$$

$$y_5 = y_{11} = 0.1295,\ y_6 = y_{10} = 0.2420,\ y_7 = y_9 = 0.3521 \text{ and } y_8 = 0.3989$$

$x_0 = (0, 0, 0)$; x^* unknown; $f^* = 1.12793 \cdot 10^{-8}$; $|f(x_0) - f(x^*)|^2 = 0.3183$

Problem 3. [13] Box three-dimensional function; $n = 3$, $m = 10$

$f_i(x) = \exp[-t_i x_1] - \exp[-t_i x_2] - x_3(\exp[-t_i] - \exp[-10t_i])$, $t_i = \frac{i}{10}$, $i = 1, ..., m$

$x_0 = (0, 10, 20)$; $x^* = (1, 10, 1)$ or $(10, 1, -1)$ or $(x_1 = x_2 \text{ and } x_3 = 0)$; $f^* = 0$
$|f(x_0) - f(x^*)|^2 = 1.0633 \cdot 10^6$

Problem 4. [13] Penalty function I; $n = 10$, $m = 11$

$$f_i(x) = 10^{-5/2}(x_i - 1),\ 1 \le i \le 10, \quad f_{n+1}(x) = (\textstyle\sum_{j=1}^{n} x_j^2) - \frac{1}{4}$$

$x_0 = (1, 1, ..., 1)$, x^* unknown; $f^* = 7.08765 \cdot 10^{-5}$ $|f(x_0) - f(x^*)|^2 = 9.0369 \cdot 10^3$

Problem 5. [13] Penalty function II; $n = 4$, $m = 8$

$$f_1(x) = x_1 - 0.2,\ f_i(x) = 10^{-5/2}(\exp(\tfrac{x_i}{10}) + \exp(\tfrac{x_{i-1}}{10}) - y_i),\ 2 \le i \le n,$$

$$f_i(x) = 10^{-5/2}(\exp(\tfrac{x_{i-n+1}}{10}) - \exp(\tfrac{-1}{10})),\ n < i < 2n,$$

$$f_{2n}(x) = (\textstyle\sum_{j=1}^{n}(n - j + 1)x_j^2) - 1 \text{ and } y_i = \exp(\tfrac{i}{10}) + \exp(\tfrac{i-1}{10}).$$

$x_0 = (1/2, 1/2, ..., 1/2)$; x^* unknown; $f^* = 9.37629 \cdot 10^{-6}$; $|f(x_0) - f(x^*)|^2 = 5.4756$

Problem 6. [13] Trigonometric function; $n = 10$, $m = 10$

$$f_i(x) = n - \textstyle\sum_{j=1}^{n} \cos x_j + i(1 - \cos x_i) - \sin x_i$$

$x_0 = (1, 0, ..., 1, 0)$; x^* unknown; $f^* = 0$; $|f(x_0) - f(x^*)|^2 = 1.1106 \cdot 10^4$

Problem 7. [13] Beale function; $n = 2$, $m = 3$

$$f_i(x) = y_i - x_1(1 - x_2^i),\ y_1 = 1.5,\ y_2 = 2.25,\ y_3 = 2.625$$

$x_0 = (1, 1)$; $x^* = (3, 0.5)$; $f^* = 0$; $||x_0 - x^*||^2 = 4.25$; $|f(x_0) - f(x^*)|^2 =$

201.7288

Problem 8. [13] Chebyquad function; $n = 10$, $m = 10$

$$f_i(x) = \frac{1}{n}\sum_{j=1}^{n} T_i(x_j) - \int_0^1 T_i(x)dx,$$

T_i is the ith Chebyshev polynomial shifted to the interval $[0, 1]$

$$\int_0^1 T_i(x)dx = 0 \text{ for } i \text{ odd}, \int_0^1 T_i(x)dx = \frac{-1}{i^2-1} \text{ for } i \text{ even}.$$

$x_0 = (1/(n+1), 2/(n+1), ..., n/(n+1))$; x^* unknown; $f^* = 6.50395 \cdot 10^{-3}$; $|f(x_0) - f(x^*)|^2 = 0.0965$

Problem 9. [15] Strictly Convex 1; $n = 10$

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

$x_0 = (1/n, ..., i/n, ..., 1)$; $x^* = (0, ..., 0)$; $f^* = 10$; $||x_0 - x^*||^2 = 3.85$, $|f(x_0) - f(x^*)|^2 = 6.5345$

Problem 10. [15] Strictly Convex 2, $n = 10$

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

$x_0 = (1, ..., 1)$; $x^* = (0, ..., 0)$; $f^* = 5.5$; $||x_0 - x^*||^2 = 10$, $|f(x_0) - f(x^*)|^2 = 15.6068$

The algorithm applies either the line-search rule taking $\alpha = \beta^m$, for $m = 0, 1, \ldots$ until the Armijo rule (11) is satisfied in Step 3, or the SA method in Step 4. Which one of these steps is taken clearly depends on the constant C that determines the switching point. The constant C introduced in Theorem 3.1 and later modified in Theorem 3.2 is a theoretical value that depends on the Lipschitz constant L and the noise bound M. It is very difficult to estimate such value. Therefore we implemented the algorithm without estimating C but taking a more practical criteria, the maximal number of backtracking steps $\overline{m}$. In fact, one can easily see that setting the maximal number of the backtracking steps to

$$\beta^{\overline{m}} < \underline{\alpha}, \quad i.e., \quad \overline{m} > \ln\Big(\frac{(1-c_1)(2\sqrt{2ML}+1)}{2L(M+2\sqrt{2ML}+1)}\Big)/\ln\beta \tag{18}$$

is in fact equivalent to the condition involving C. This bound does not solve the estimation problem as $\overline{m}$ still depends on L and M, but allows one to test different values for $\overline{m}$ in practical implementation. Taking $\overline{m}$ too small would of course result in a switching point that is not so close to the optimal solution,

so the line-search method would not be fully used. On the other hand, taking $\overline{m}$ too large will yield unproductive line-search steps but the switching point will be reached eventually. In all our experiments we took $\overline{m} = 5$ and such a choice appears to be robust. So all results reported here are obtained with the following rule: if the sufficient decrease condition is not satisfied for $\alpha = \beta^{\bar{m}}$, the algorithm switches to SA method. Other parameter values in the line-search procedures are selected as $c_1 = 10^{-4}$ and $\beta = 0.5$. The new algorithms switch to SA at some iteration, say j. So two possibilities for the gain coefficients are considered

(I) $a_k = (k+1)^{-1}$, $k = j, j+1, \ldots$.

(II) $a_k = (k-j)^{-1}$, $k = j+1, j+2, \ldots$.

The stopping criteria in all tests is either $\|G_k\| \leq 10^{-5}$ or the maximal number of 1000 function evaluations is reached.

Each test consists of 50 independent runs. Thus for each method and each problem we have a sample

$$(||G^{(i)}||, x^{(i)}, y^{(i)}), \quad i = 1, \ldots, 50.$$

where $G^{(i)}, x^{(i)}, y^{(i)}$ are the last estimates of the gradient value, optimal point and optimal functional value. A run is considered successful if the condition $||G^{(i)}|| < 1$ is satisfied.

Two performance measures are reported. The empirical standard deviation (ESD), σ_e, is defined by

$$\sigma_e^2 = \sum_{i:||G^{(i)}||<1} ||x^{(i)} - x^*||^2 / N_s,$$

where $N_s \leq 50$ is the number of successful runs. The Empirical Standard Deviation σ_e can serve as a characteristic of the estimation quality (see [4]). There are problems for which the empirical standard deviation cannot be evaluated since the optimal point is not known or it is not unique (Problems 2, 3, 4, 5, 6 and 8). Another measure for estimation quality that we report is the Mean-Squared Error (MSE) of the objective function estimator, given by

$$MSE(f) = \sum_{i:||G^{(i)}||<1} (y^{(i)} - f^*)^2 / N_s.$$

The mean-squared error of the objective function estimator is calculated for each problem.

The results for GSLS method and comparison with the SA method are given in the following tables. Table 1 contains the number of successful runs for SA and GSLS methods, depending on the choice of the gain sequence after switching to SA (choices I and II).

The GSLS method proposed in this paper is significantly better than SA when one compares the number of successful runs. The first algorithm, GSLS(I)

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	50	50	46	50	50	46
2	50	50	50	50	50	50
3	0	50	50	0	50	50
4	0	50	50	0	50	50
5	0	45	2	0	50	1
6	0	50	46	0	50	46
7	0	48	0	0	50	0
8	15	50	28	2	50	48
9	50	50	50	50	50	50
10	50	50	50	50	50	50
success:	43%	98.6%	74.7%	40.4%	100%	78.2%

Table 1: Number of successful runs

is successful in 493 out of 500 runs for the noise with variance $\sigma = 0.1$, while for the smallest noise $\sigma = 0.01$ it is successful in all 500 runs. The SA algorithm is successful only in 215 and 202 runs for $\sigma = 0.1$ and $\sigma = 0.01$, respectively. The second line-search method GSLS(II) is somewhere between these two, with success in 372 and 391 runs, respectively. The obtained results confirm the assumption that the line-search will yield large steps when the noise is weak. On the other hand, the SA method does not recognized the level of noise and decreases steps independently of the noise. We can also see that the switching point is well defined as the proposed combination of line-search and SA significantly improves the number of successful runs, if compared with SA. The switch at a later iteration might have taken the iterative sequence out of the solution's neighborhood.

Significantly better results are obtained with GSLS(I), *i.e.,* with the method that takes $a_k = (1 + k)^{-1}$ after switching to SA. This is in line with the expectation that the last line-search iteration will be in proximity of the solution and thus the noise will significantly interfere with the process, so smaller SA steps are preferable afterwards. SA is successful only if the initial approximation is close to the solution (problems 1,7,9 and 10). But as we will see in Tables 2 and 3, where the values of ESD and MSE as well as the average number of line-search iterations are reported, in all tests the new method is better in terms of computational efforts versus quality of approximation. Considering the common stopping criteria for problems without noise, $\|G_k\| \leq 10^{-5}$, only GSLS(II) fulfilled that criterion in 38 out of 50 runs for Problem 9 and $\sigma = 0.01$. In all other cases the algorithms stopped due to the maximal number of functional evaluations. Thus the reported results are obtained with the same computational effort for all tested methods in terms of functional evaluations.

The square of the empirical standard deviation, σ_e^2 and the average number of line search iterations before switching to SA, itnAR, are reported in Table

2. The empirical standard deviation is calculated only for the problems with known solutions. The weakest results are obtained for Problem 1, but even in that case GSLS is better than SA, particularly for the case $\sigma = 0.01$. The number of Armijo steps is relatively large for this problem (around 10 for the smallest noise and around 30 for the largest noise) in both GSLS. That is explained with the fact that x_0 is far away from the solution and thus the globalization with the line-search rule yields significantly better results in comparison with SA. The difference in the number of line-search steps (10 versus 30) is in line with the intuitive reasoning: small noise allows us to use line-search steps for longer, as the noise is important only in a tight neighborhood of the solution. The only successful method for Problem 7 is GSLS(I) and the approximate solution is quite close to the exact solution. In all tests the line-search is used at initial stages of the iterative process and there is no example where SA is applied immediately. Thus, introducing line-search steps is numerically justified.

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	125.627124	88.9515396	10.5	85.22198	9.3
7	fail	0.0051343	7.79	fail	fail
9	0.0017502	0.00189925	4.46	0.0015737	4.46
10	0.692694	0.2619552	4.78	0.14223548	4.78
$\sigma = 0.01$					
1	125.422589	10.931906	32.92	11.028006	32.5
7	fail	0.0041855	13.3	fail	fail
9	0.0008237	0.0000253	5.1	0.000231	4.94
10	0.685479	0.053169	8.52	0.0997031	8.52

Table 2: σ_e^2 and average number of Armijo steps (itnAR)

Table 3 contains the Mean Square Error values, MSE(f), and the average number of Armijo steps, itnAR for all examples. As expected MSE values are similar to σ_s values in the sense that problems with large MSE values have large σ_s values too, but MSE values are calculated for all test examples. The results confirm that the number of Armijo steps is significantly larger for problems with small noise ($\sigma = 0.01$) than for problems with large noise, $\sigma = 0.1$. The average number is almost twice as large as for the smallest noise: for example in Algorithms of type GSLS(I) we have 6.74 steps per successful run for $\sigma = 0.1$ and 11.82 steps per successful run for $\sigma = 0.01$. The advantages of GSLS method are quite obvious at Problem 1. SA can not approach the solution, while both new methods use a large number of Armijo steps (32.92 and 32.5 iterations) and yield good approximations of the solution for $\sigma = 0.01$.

The plots given in Figures 1 and 2 show the values of $\|G_k\|$ and step size a_k for the first 20 iterations, on Problem 10 with $\sigma = 0.1, 0.01$, for SA, GSLS(I) and GSLS(II) methods. The gradient values decrease significantly faster with

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	42.978024	28.741947	10.5	29.80379	9.3
2	0.3349472	0.0035007	5.98	0.01756547	5.98
3	fail	0.0115834	12.12	0.0114113	12.12
4	fail	0.002752	3.48	0.002758	3.48
5	fail	0.00290395	6.91	0.00385406	8
6	fail	0.00289889	7.08	0.00311755	7.24
7	fail	0.004229	7.79	fail	fail
8	0.017301023	0.010854015	4.34	0.01431826	4.68
9	0.00397999	0.0033668	4.46	0.00336359	4.46
10	0.0096439	0.003697	4.78	0.00338097	4.78
$\sigma = 0.01$					
1	42.7774287	0.0008434	32.92	0.0235777	32.5
2	0.3193908	0.0000374	10.54	0.0002833	10.54
3	fail	0.0077512	13.82	0.0075879	13.82
4	fail	0.00005147	5.54	0.00002859	5.54
5	fail	0.00009268	10.94	0.00000276	11
6	fail	0.00004513	8.96	0.00004815	8.96
7	fail	0.0000319	13.3	fail	fail
8	0.0091224794	0.007754975	8.54	0.007408019	8.04
9	0.00004266	0.000033967	5.1	0.00003608	4.94
10	0.0051316	0.0000437	8.52	0.0000792	8.52

Table 3: MSE(f) and average number of Armijo steps (itnAR)

line-search steps for both σ values. The switching iteration is market by a dot.

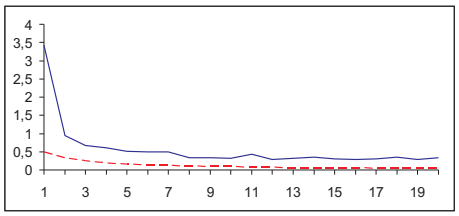

Figure 1.1: SA

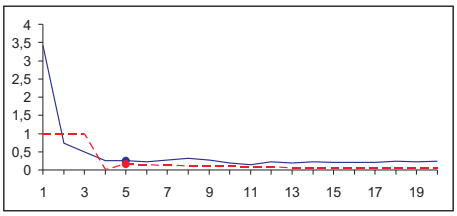

Figure 1.2: GSLS(I)

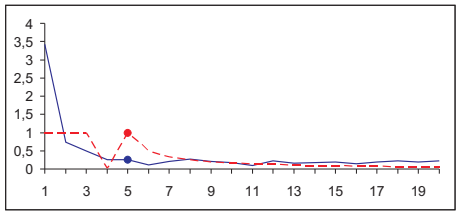

Figure 1.3: GSLS(II)

Figure 1: Problem 10, $\sigma = 0.1$, $\|G_k\|$ (solid line) and a_k (dashed line) values

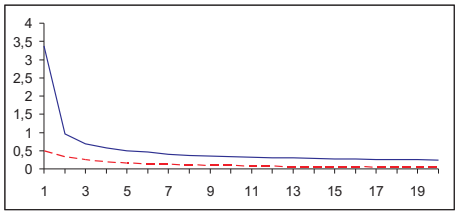

Figure 2.1: SA

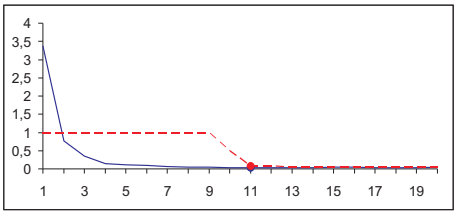

Figure 2.2: GSLS(I)

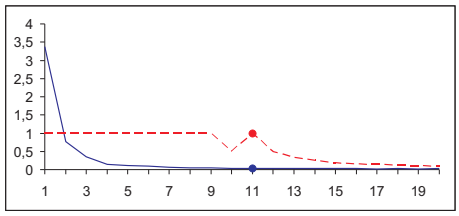

Figure 2.3: GSLS(II)

Figure 2: Problem 10, $\sigma = 0.01$, $\|G_k\|$ (solid line) and a_k (dashed line) values

Additional numerical results are given in Appendix A. The results are obtained with different gain coefficients for the SA method. Some other results are also available at *http://sites.dmi.rs/personal/krejicn/Natasa%20Krejic%20-%20Curriculum%20Vitae.htm*

5 Consclusion

The proposed method for unconstrained minimization in noisy environment combined two different approaches: the gradient line-search with the Armijo rule

and the SA method. It aims at employing the good properties of both methods. A large step size obtained by the Armijo rule at the beginning of the iterative procedure yields fast progress while we are far away from the solution. The safe but slow SA method is applied afterwards to ensure almost sure convergence and prevent zig-zag behavior of line search methods in the proximity of the solution. The key point in applying the proposed method is the rule that determines the switching point between the line-search and the SA method. Under a set of standard assumption in noisy optimization we proved that such a point exists almost surely. Furthermore we proved that SA will be employed almost surely after a finite number of line search steps and thus almost sure convergence of the sequence is guaranteed. An additional assumption that limits the realized noise does not influence the applicability of the proposed method in real-life problems. The question of suppressing noise is without doubt an important question in solving noisy problems. As one of the referees suggested, taking an average value of the gradients in several subsequent iterations could be an appealing strategy. Such a strategy would imply a change in the Armijo rule within line-search iterations and might be beneficial in the case of large noise. We are planning to consider such an approach in future research.

The method is tested on a set of examples from literature, by adding the Gaussian noise to the objective function and its gradient and compared with the SA method. The results confirm that the combination of line-search and SA method improves the robustness as well as the quality of approximate solutions within a given constraint on the computational effort, measured in functional evaluations.

The main drawbacks of the proposed method are inherited from SA convergence conditions. These are the assumption that x^* is unique and the need to use gradient values. Gradient values are not always available in real-life problems and avoiding their calculation is an important point. In fact line-search is a powerful tool for many other search directions, not only the negative gradient but also quasi-Newton directions and other descent directions. Further research is needed to develop methods that can successfully work with gradient approximations in line search procedures.

Aknowledgements: We are grateful to the two anonymous referees, whose suggestions helped us to improve this paper.

References

[1] S. Andradóttir, *A scaled stochastic approximation algorithm,* Management Science, Vol. 42 No. 4 (1996), 475-498.

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

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

[4] H. T. Fang, H. F. Chen, *Almost surely convergent global optimization algorirhm using noise-corrupted observations,* J. Optim. Theory Appl., Vol. 104, No. 2 (2000), 343-376.

[5] Ch. Kao, Sh.-P. Chen, *A stochastic quasi-Newton method for simulation response optimization,* European Journal of Operational Research, Vol. 173 No.1 (2006), 30-46.

[6] H. Kesten, *Accelerated stochastic approximation,* Ann. Math. Stat., Vol. 29 No. 1 (1958), 41-59.

[7] J. Kiefer, J. Wolfowitz, *Stochastic estimation of the maximum of a regression function,* Ann. Math. Stat., Vol. 23 No.3 (1952), 462-466.

[8] H. Kushner, *Stochastic approximation: a survey,* Wiley Interdiciplinary Reviews: Computational Statistics, Vol.2, No. 1 (2010), 87-96.

[9] H. J. Kushner, T. Gavin *Extensions of Kesten's adaptive stochastic approximation method,* Ann. Stat., Vol. 1, No. 5 (1973), 851-861.

[10] B. Li, Y. Ong, M. Le, C. Goh, *Memetic Gradient Search*, Proc. of IEEE Congress on Evolutionary Computation 2008 (CEC2008), 2894-2901.

[11] S. Lucidi, M. Sciandrone, *A derivative-free algorithm for bound constrained optimization,* Computational Optimization and Applications, Vol. 21 No. 2 (2002), 119-142

[12] J.-M. Monnez, *Almost sure convergence of stochastic gradient processes with matrix step sizes,* Statistics & Probability Letters Vol. 76, No. 5 (2006), 531-536

[13] J. J. Moré, B. S. Garbow, K. E. Hillstrom *Testing Unconstrained Optimization Software,* ACM Transactions on Mathematical Software, Vol. 7, No. 1 (1981), 17-41.

[14] J. Nocedal, S. J. Wright, *Numerical optimization,* Springer-Verlag, New York, 1999

[15] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem,* SIAM. J. Optim., Vol.7, No. 1 (1997), 26-33.

[16] H. Robbins, S. Monro, *A stochastic approximation method,* Ann. Math. Stat., Vol. 22 No. 3 (1951), 400-407.

[17] D. Ruppert, *A Newton-Raphson version of the multivariate Robbins-Monro procedure,* Ann. Stat., Vol. 13, No. 1 (1985), 236-245.

[18] 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), 325-340.

[19] J. C. Spall, *Adaptive stochastic approximation by the simultaneous perturbarion method,* IEEE Trans. Automat. Contr., Vol. 45, No. 10 (2000), 1839-1853.

[20] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control,* John Wiley & Sons, Inc., Hoboken, New Jersey, 2003.

[21] J.C. Spall, *Feedback and weighting Mechanism for Improving Jacobian Estimates in the Adaptive Simultaneous Perturbation Algorithm*, IEEE Transactions on Automatic Control, Vol. 54, No. 6, (2009), 1216-1229.

[22] Xiumei Yue, *Improved Simulataneous Perturbation Stochastic Approximation and Its Application in Reinforcement Learning*, Vol.1, 329-339, International Conference on Computer Science and Software Engineering, 2008.

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

[24] C. Z. Wei, *Multivariate adaptive stochastic approximation,* Ann. Stat., Vol. 15, No. 3 (1987), 1115-1130.

[25] Zi Xu, Yu-Hong Dai, *New stochastic approximation algorithms with adaptive step size*, Optim. Lett., Vol. 6, No. 8 (2012), 1831-1846.

A Additional numerical results

gain coefficients A, [20]: $a_k = \dfrac{a}{(k+1+A)^\alpha},\ A = 100,\ \alpha = 0.501,\ a = 1$

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	50	50	50	50	50	50
2	50	50	50	50	50	50
3	50	50	50	50	50	50
4	0	50	50	0	50	50
5	0	50	50	0	50	50
6	0	50	50	0	50	50
7	4	2	0	0	0	0
8	50	50	50	50	50	50
9	50	50	50	50	50	50
10	50	50	50	50	50	50
success:	60.8%	90.4%	90%	60%	90%	90%

Table A.1: Number of successful runs

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	121.0426	82.899469	10.5	82.582009	10.5
7	0.0144607	0.0142281	5	fail	fail
9	0.0065941	0.0018238	4.46	0.001833	4.46
10	0.832608	0.2223136	4.78	0.218987	4.78
$\sigma = 0.01$					
1	120.8994	10.996916	32.92	11.008855	32.92
7	fail	fail	fail	fail	fail
9	0.0053672	0.00001934	5.1	0.00001924	5.1
10	0.823739	0.0834911	8.82	0.0819886	8.52

Table A.2: σ_e^2 and average number of Armijo steps (itnAR)

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	42.2433	27.589528	10.5	27.52283	10.5
2	0.002926	0.0034218	5.98	0.0034213	5.98
3	0.00295	0.011389	12.12	0.0113746	12.12
4	fail	0.0027555	3.48	0.0027554	3.48
5	fail	0.0028692	6.26	0.0028678	6.26
6	fail	0.0029015	7.08	0.0029009	7.08
7	0.0029660	0.0026383	5	fail	fail
8	0.01314	0.0108821	4.34	0.0108835	4.34
9	0.004012	0.0033682	4.46	0.0033682	4.46
10	0.013923	0.00353367	4.78	0.0035224	4.78
$\sigma = 0.01$					
1	42.0478	0.0007886	32.92	0.00077969	32.92
2	0.0000309	0.00002061	10.54	0.00002055	10.54
3	0.0000334	0.0075707	13.82	0.007556	13.82
4	fail	0.00003489	5.54	0.000034091	5.54
5	fail	0.00003225	10.94	0.00003228	10.94
6	fail	0.00004591	8.96	0.00004599	8.96
7	fail	fail	fail	fail	fail
8	0.008507	0.0077535	8.54	0.00775349	8.54
9	0.0000578	0.00003396	5.1	0.00003396	5.1
10	0.0092167	0.00006379	8.52	0.00006257	8.52

Table A.3: MSE(f) and average number of Armijo steps (itnAR)

gain coefficients B, [20]: $a_k = \dfrac{a}{(k+1+A)^\alpha},\ A = 100,\ \alpha = 0.501,\ a = 0.1$

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	0	50	50	0	50	50
2	50	50	50	50	50	50
3	0	50	50	0	50	50
4	50	50	50	50	50	50
5	50	50	50	50	50	50
6	50	50	50	50	50	50
7	50	50	50	50	50	50
8	50	50	50	50	50	50
9	0	50	50	0	50	50
10	0	50	50	0	50	50
success:	60%	100%	100%	60%	100%	100%

Table B.1: Number of successful runs

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	fail	95.7944784	10.5	95.7665068	10.5
7	0.261215	0.0538276	8.1	0.0520003	8.1
9	fail	0.017043	4.46	0.0168833	4.46
10	fail	0.4304804	4.78	0.4296233	4.78
$\sigma = 0.01$					
1	fail	10.908572	32.92	10.910184	32.92
7	0.2614917	0.0255486	13.3	0.0244098	13.3
9	fail	0.00025191	5.1	0.000249239	5.1
10	fail	0.131444	8.52	0.131192	8.52

Table B.2: σ_e^2 and average number of Armijo steps (itnAR)

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	fail	29.8742945	10.5	29.8696448	10.5
2	0.010195	0.004398	5.98	0.0043807	5.98
3	fail	0.0116541	12.12	0.011652	12.12
4	0.010936	0.002735	3.48	0.0027345	3.48
5	0.002566	0.0124461	6.26	0.012206	6.26
6	0.007077	0.0029019	7.08	0.0029014	7.08
7	0.00658	0.0036576	8.1	0.00363827	8.1
8	0.0208914	0.012674	4.34	0.0126472	4.34
9	fail	0.0035372	4.46	0.0035353	4.46
10	fail	0.0050786	4.78	0.00506645	4.78
$\sigma = 0.01$					
1	fail	0.0008795	32.92	0.0008754	32.92
2	0.0056422	0.0001420	10.54	0.0001389	10.54
3	fail	0.0078115	13.82	0.0078099	13.82
4	0.0064634	0.00026495	5.54	0.00026477	5.54
5	0.0000417	0.0012029	10.94	0.00117512	10.94
6	0.002765	0.0000491	8.96	0.00004897	8.96
7	0.00523	0.00006156	13.3	0.00005879	13.3
8	0.0160763	0.0078465	8.54	0.0078438	8.54
9	fail	0.00003401	5.1	0.00003401	5.1
10	fail	0.0001281	8.52	0.0001277	8.52

Table B.3: MSE(f) and average number of Armijo steps (itnAR)

gain coefficients C, [20]: $a_k = \dfrac{a}{k+1}$, $a = 0.1$

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	0	50	50	0	50	50
2	50	50	50	50	50	50
3	0	50	50	0	50	50
4	50	50	50	50	50	50
5	50	50	50	50	50	50
6	10	50	50	7	50	50
7	0	50	50	0	50	50
8	50	50	50	50	50	50
9	0	50	50	0	50	50
10	0	50	50	0	50	50
success:	42%	100%	100%	41.4%	100%	100$

Table C.1: Number of successful runs

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	fail	96.342625	10.5	95.561843	10.5
7	fail	0.125638	8.1	0.078354	8.1
9	fail	0.018855	4.46	0.012795	4.46
10	fail	0.4397850	4.78	0.4049329	4.78
$\sigma = 0.01$					
1	fail	10.900485	32.92	10.932018	32.92
7	fail	0.0521027	13.3	0.0326230	13.3
9	fail	0.0002832	5.1	0.0001871	5.1
10	fail	0.133937	8.52	0.125421	8.52

Table C.2: σ_e^2 and average number of Armijo steps (itnAR)

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	fail	29.963021	10.5	29.833755	10.5
2	0.018291	0.005420	5.98	0.004676	5.98
3	fail	0.0119313	12.12	0.011655	12.12
4	0.0039711	0.002736	3.48	0.002712	3.48
5	0.0248071	0.022087	6.26	0.014402	6.26
6	0.0093885	0.00291	7.08	0.00290176	7.08
7	fail	0.0052070	8.1	0.0040522	8.1
8	0.020718	0.012961	4.34	0.011934	4.34
9	fail	0.0035588	4.46	0.0034795	4.46
10	fail	0.0052181	4.78	0.0047484	4.78
$\sigma = 0.01$					
1	fail	0.00090981	32.92	0.00083693	32.92
2	0.012812	0.0002263	10.54	0.0001574	10.54
3	fail	0.0078491	13.82	0.0078136	13.82
4	0.0000412	0.0002667	5.54	0.0002567	5.54
5	0.0234929	0.0016016	10.94	0.0011419	10.94
6	0.0038791	0.0000506	8.96	0.0000466	8.96
7	fail	0.0001925	13.3	0.00008712	13.3
8	0.0158904	0.0078800	8.54	0.0078043	8.54
9	fail	0.00003403	5.1	0.00003400	5.1
10	fail	0.0001325	8.52	0.0001181	8.52

Table C.3: MSE(f) and average number of Armijo steps (itnAR)

gain coefficients D, [6, 3]: $a_k = \dfrac{a}{s_k + 1}, \quad s_{k+1} = s_k + I(G_{k+1}^T G_k),$
$(I(t) = 1 \text{ if } t < 0,\ \ I(t) = 0 \text{ if } t \geq 0), \quad a = 1$

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	46	50	38	50	50	45
2	50	50	50	50	50	50
3	0	50	0	0	50	0
4	0	50	50	0	50	50
5	0	28	0	0	50	0
6	0	50	0	0	50	50
7	0	27	0	0	50	0
8	0	48	0	0	50	0
9	50	50	50	50	50	50
10	50	50	50	50	50	50
success:	39.2%	90.6%	47.6%	40%	100%	59%

Table D.1: Number of successful runs

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	38.536352	36.676602	10.5	94972.44284	10.34
7	fail	0.0016100	9.22	fail	fail
9	0.0015442	0.0016006	4.46	0.0345863	4.46
10	0.115646	0.1136127	4.78	0.0559252	4.78
$\sigma = 0.01$					
1	13.9445	11.02922	32.92	29465.36399	32.44
7	fail	0.00026264	13.3	fail	fail
9	0.0000165	0.000017	5.1	0.000318	4.94
10	0.0190116	0.0132143	8.52	0.0018534	8.52

Table D.2: σ_e^2 and average number of Armijo steps (itnAR)

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	0.004862	0.014505	10.5	51.74475	10.34
2	0.3349523	0.0034784	5.98	0.3297073	5.98
3	fail	0.011529	12.12	fail	fail
4	fail	0.002755	3.48	0.002737	3.48
5	fail	0.0026771	7.75	fail	fail
6	fail	0.0029073	7.08	fail	fail
7	fail	0.0016704	9.22	fail	fail
8	fail	0.0104589	3.92	fail	fail
9	0.00398386	0.00336686	4.46	0.0036238	4.46
10	0.0041752	0.0033385	4.78	0.00342104	4.78
$\sigma = 0.01$					
1	0.001699	0.0007788	32.92	73.86254	32.44
2	0.3193908	0.0000209	10.54	0.317417	10.54
3	fail	0.0076047	13.82	fail	fail
4	fail	0.0000285	5.54	0.0000303	5.54
5	fail	0.00003166	10.94	fail	fail
6	fail	0.0000463	8.96	0.0000502	8.96
7	fail	0.0000319	13.3	fail	fail
8	fail	0.0077533	8.54	fail	fail
9	0.0000411	0.00003396	5.1	0.0000364	4.94
10	0.00004616	0.00003212	8.52	0.00003144	8.52

Table D.3: MSE(f) and average number of Armijo steps (itnAR)

gain coefficients E, [6, 3]: $a_k = \dfrac{a}{s_k + 1}, \quad s_{k+1} = s_k + I(G_{k+1}^T G_k),$
$(I(t) = 1 \text{ if } t < 0, \ I(t) = 0 \text{ if } t \geq 0), \quad a = 0.1$

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	50	50	50	50	50	50
2	50	50	50	50	50	50
3	50	50	50	50	50	50
4	0	50	50	0	50	50
5	50	50	50	50	50	50
6	2	50	50	0	50	50
7	50	50	0	50	50	0
8	50	50	50	50	50	50
9	50	50	50	50	50	50
10	50	50	50	50	50	50
success:	80.4%	100%	90%	80%	100%	90%

Table E.1: Number of successful runs

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	118.3953499	82.222250	10.5	80.965305	10.5
7	0.0081745	0.0191389	8.1	fail	fail
9	0.0073673	0.0081150	4.46	0.0019346	4.46
10	0.7454309	0.3411463	4.78	0.2101301	4.78
$\sigma = 0.01$					
1	118.233373	10.9157290	32.92	11.0123029	32.92
7	0.0004904	0.0186394	13.3	fail	fail
9	0.0032182	0.00009375	5.1	0.000019704	5.1
10	0.7357191	0.1043706	8.52	0.0796481	8.52

Table E.2: σ_e^2 and average number of Armijo steps (itnAR)

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	41.798171	27.424956	10.5	27.193449	10.5
2	0.0033206	0.0037454	5.98	0.0034165	5.98
3	0.0036856	0.0116663	12.12	0.0113331	12.12
4	fail	0.0027207	3.48	0.0027545	3.48
5	0.0079767	0.0032263	6.26	0.0028036	6.26
6	0.00005802	0.0028907	7.08	0.0029003	7.08
7	0.0025004	0.00338343	8.1	fail	fail
8	0.013114	0.0109367	4.34	0.0108823	4.34
9	0.0039959	0.0033837	4.46	0.0033677	4.46
10	0.0113042	0.0041788	4.78	0.0035002	4.78
$\sigma = 0.01$					
1	41.600274	0.0008710	32.92	0.00077782	32.92
2	0.00007962	0.00010291	10.54	0.0000205	10.54
3	0.0003867	0.0078148	13.82	0.0075176	13.82
4	fail	0.00021048	5.54	0.00003131	5.54
5	0.0026003	0.00097469	10.94	0.000032315	10.94
6	fail	0.00004745	8.96	0.000046077	8.96
7	0.00002901	0.00004126	13.3	fail	fail
8	0.00850702	0.00781918	8.54	0.00775355	8.54
9	0.0000493	0.00003403	5.1	0.00003396	5.1
10	0.0066991	0.00008631	8.52	0.00006052	8.52

Table E.3: MSE(f) and average number of Armijo steps (itnAR)

gain coefficients F, [25]: $a_k = \dfrac{a}{(k+1+A)^{q(\frac{s_k}{k})}}, \quad q(\dfrac{s_k}{k}) = \max\left(1 - \left|\dfrac{s_k}{k} - \dfrac{1}{2}\right|, 0.501\right)$

$$(I(t) = 1 \text{ if } t < 0,\ \ I(t) = 0 \text{ if } t \geq 0), \quad a = 1$$

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	50	50	50	50	50	50
2	50	50	50	50	50	50
3	50	50	50	50	50	50
4	0	50	50	0	50	50
5	50	50	50	50	50	50
6	5	50	50	6	50	50
7	50	50	0	50	50	0
8	50	50	50	50	50	50
9	50	50	50	50	50	50
10	50	50	50	50	50	50
success:	80%	100%	90%	81.2%	100%	90%

Table F.1: Number of successful runs

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	121.0425578	83.4379495	10.5	82.5820087	10.5
7	0.0022018	0.0034627	8.1	fail	fail
9	0.0073457	0.0060687	4.46	0.001833	4.46
10	0.832608	0.3128886	4.78	0.218987	4.78
$\sigma = 0.01$					
1	120.899406	10.941975	32.92	11.008855	32.92
7	0.0001114	0.0020164	13.3	fail	fail
9	0.0053672	0.0000644	5.1	0.00001924	5.1
10	0.8237388	0.0944149	8.52	0.0819886	8.52

Table F.2: σ_e^2 and average number of Armijo steps (itnAR)

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	42.2432718	27.6883046	10.5	27.5228296	10.5
2	0.0030965	0.00352996	5.98	0.0034213	5.98
3	0.0087767	0.01163409	12.12	0.0113746	12.12
4	fail	0.0027195	3.48	0.0027554	3.48
5	0.0025681	0.002965	6.26	0.0028678	6.26
6	0.0006954	0.0028992	7.08	0.002909	7.08
7	0.0025116	0.0033131	8.1	fail	fail
8	0.0130993	0.0109146	4.34	0.0108835	4.34
9	0.0040162	0.0033809	4.46	0.0033682	4.46
10	0.0139235	0.0039766	4.78	0.0035224	4.78
$\sigma = 0.01$					
1	42.0478177	0.00083982	32.92	0.0007797	32.92
2	0.0000422	0.000037	10.54	0.00002055	10.54
3	0.0043837	0.0077682	13.82	0.00755600	13.82
4	fail	0.00016708	5.54	0.00003409	5.54
5	0.00004143	0.0002453	10.94	0.0000323	10.94
6	0.00061929	0.0000462	8.96	0.00004599	8.96
7	0.00002896	0.0000315	13.3	fail	fail
8	0.00850702	0.0077957	8.54	0.0077535	8.54
9	0.00005783	0.00003401	5.1	0.00003396	5.1
10	0.0092167	0.00007500	8.52	0.00006257	8.52

Table F.3: MSE(f) and average number of Armijo steps (itnAR)

gain coefficients G, [25]: $a_k = \dfrac{a}{(k+1+A)^{q(\frac{s_k}{k})}}, \quad q(\dfrac{s_k}{k}) = \max\left(1 - \left|\dfrac{s_k}{k} - \dfrac{1}{2}\right|, 0.501\right)$

$(I(t) = 1 \text{ if } t < 0,\ I(t) = 0 \text{ if } t \geq 0), \quad a = 0.1$

	$\sigma = 0.1$			$\sigma = 0.01$		
pr	SA	GSLS(I)	GSLS(II)	SA	GSLS(I)	GSLS(II)
1	0	50	50	0	50	50
2	50	50	50	50	50	50
3	0	50	50	0	50	50
4	50	50	50	50	50	50
5	50	50	50	50	50	50
6	50	50	50	50	50	50
7	50	50	50	50	50	50
8	50	50	50	50	50	50
9	0	50	50	0	50	50
10	0	50	50	0	50	50
success:	60%	100%	100%	60%	100%	100%

Table G.1: Number of successful runs

	SA	GSLS(I)		GSLS(II)	
prb	σ_e^2	σ_e^2	itnAR	σ_e^2	itnAR
$\sigma = 0.1$					
1	fail	95.841642	10.5	95.766507	10.5
7	0.261215	0.091945	8.1	0.0520003	8.1
9	fail	0.021593	4.46	0.016883	4.46
10	fail	0.4470755	4.78	0.4296233	4.78
$\sigma = 0.01$					
1	fail	10.901866	32.92	10.910184	32.92
7	0.2614917	0.0460891	13.3	0.0244098	13.3
9	fail	0.0003052	5.1	0.0002492	5.1
10	fail	0.1331996	8.52	0.131192	8.52

Table G.2: σ_e^2 and average number of Armijo steps (itnAR)

	SA	GSLS(I)		GSLS(II)	
prb	MSE(f)	MSE(f)	itnAR	MSE(f)	itnAR
$\sigma = 0.1$					
1	fail	29.8817907	10.5	29.869645	10.5
2	0.010195	0.004974	5.98	0.0043807	5.98
3	fail	0.012035	12.12	0.0116516	12.12
4	0.010936	0.002772	3.48	0.002734	3.48
5	0.002565	0.015744	6.26	0.012206	6.26
6	0.0102667	0.002969	7.08	0.002901	7.08
7	0.0065804	0.004209	8.1	0.0036383	8.1
8	0.0208914	0.013657	4.34	0.012647	4.34
9	fail	0.003586	4.46	0.003535	4.46
10	fail	0.0053331	4.78	0.0050664	4.78
$\sigma = 0.01$					
1	fail	0.0009065	32.92	0.0008754	32.92
2	0.0056422	0.0002221	10.54	0.0001389	10.54
3	fail	0.007859	13.82	0.0078099	13.82
4	0.006463	0.000271	5.54	0.0002648	5.54
5	0.0000417	0.001659	10.94	0.001175	10.94
6	0.005773	0.0000547	8.96	0.00004897	8.96
7	0.0052300	0.0001443	13.3	0.0000588	13.3
8	0.016076	0.0079555	8.54	0.0078438	8.54
9	fail	0.0000341	5.1	0.00003401	5.1
10	fail	0.0001316	8.52	0.0001277	8.52

Table G.3: MSE(f) and average number of Armijo steps (itnAR)