UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCES
DEPARTMENT OF MATHEMATICS
AND INFORMATICS

Milena Kresoja

Modifications of Stochastic Approximation Algorithm Based on Adaptive Step Sizes

- PhD Thesis -

Modifikacije algoritma stohastičke aproksimacije zasnovane na prilagođenim dužinama koraka

- doktorska disertacija -

Novi Sad, 2017

Introduction

> Nothing takes place in the world
> whose meaning is not that of
> some maximum or minimum.
>
> Leonhard Euler

One of the major use of mathematics in the real world is solving problems as efficiently as possible. Finding optimal result under given circumstances is called optimization. It is present in almost every area of application and embedded as a fundamental approach in the analysis of decision making problems as well as in the analysis of physical, technical and many other systems. In business setting, investors seek to maximize profit, whereas to minimize loss and risk. While designing technical systems, engineers seek to optimize performance of their designs, to minimize effort or to maximize benefit.

An optimization problem is given in a form of minimizing or maximizing some objective function of one or several variables, possibly subject to constraints on these variables. In majority cases, obtaining optimal solution of optimization problem is very hard, expensive or even impossible. Optimization techniques are used to find an approximation of the optimal solution. Wide variety of numerical optimization methods have been developed for different types of problems. More than sixty years ago, the most of proposed optimization methods were deterministic. Deterministic methods assume a perfect information about the objective function and its derivatives. However, almost all real world problems are faced with

uncertainty and involve parameters which are unknown at the time of formulation. For example, making investment decisions in order to increase profit usually depends on future interest rates, future demands and future prices. As a consequence, mathematical models cannot be fully specified since the future outcomes are not deterministic. Minimization and maximization of a function with the presence of randomness are refereed to stochastic optimization. Randomness can enter the problem through the objective function or through the set of constraints. Stochastic optimization algorithms have been growing rapidly in popularity over the last years and have become widely available, [56]. Like in deterministic case, there is no single method available for solving all optimization problems efficiently.

Within the thesis we consider unconstrained minimization problems in noisy environment. In this set-up, the objective function and its gradient are random, i.e., disturbed by the random variables (stochastic noises). This means that only noisy observations of the objective function and its gradient are available. Noisy environment is modelled by adding a random variable and a random vector to the true values of the objective function and its gradient, respectively. The fundamental approach for solving unconstrained minimization problem in noisy environment is Stochastic Approximation (SA) algorithm. It is originally proposed for finding roots of nonlinear scalar function by Robbins and Monro, [45], and later extended to multidimensional systems by Blum, [5]. Iterative rule of SA algorithm is motivated by deterministic gradient decent method and uses only noisy gradient observation. Various modifications of SA algorithm are proposed to improve and accelerate the optimization process, [1, 4, 11, 24, 29, 30, 42, 55, 65, 67, 68]. These modifications are based on the step size and/or search direction selection which are fundamental issues in the iterative rule of the SA algorithm.

In the thesis, we focus on modifications of SA algorithm based on adaptive step sizes. The first SA algorithm with adaptive step size scheme is proposed by Kesten, [24], for one dimensional case and by Delyon and Juditsky, [11], for multidimensional problems. These adaptive step size schemes are based on monitoring frequency of sign changes of the differences between two successive iterates. We propose a class of adaptive step size schemes

for SA algorithms based only on previously observed noisy function values. Two main schemes are proposed. In both schemes, at each iterate, interval estimates of the optimal function value are constructed using fixed number of previously observed (noisy) function values. If the observed (noisy) function value in kth iterate is smaller than the lower limit of the interval, we consider this scenario as a good one, since it represents a sufficient decrease of the objective function. In this case, we suggest using a larger step size in the next $(k+1)$th iterate. If the function value in kth iterate is larger than the upper limit of the interval, we reject the current iterate by taking zero step size in the next iterate. Similar approach is implemented in [55, 66]. Otherwise, if the function value lies in the interval, we propose a small safe step size. In this manner, a faster progress of the algorithm is ensured when it is expected that larger steps will improve the performance of the algorithm. The proposed schemes differ in the intervals that we construct at each iterate. In the first scheme, we drew our inspiration from the interval estimation theory. We construct a symmetrical interval that can be viewed as a confidence-like interval for the optimal function value. The bounds of the interval are shifted means of the fixed number of previously observed function values. We suggest taking a value comparable to the standard deviation of the noise for the width of the interval. The generalization of this scheme is also presented. In the second scheme, we have used the Extreme Value Statistics to construct the intervals. For the lower and upper bounds of the interval, we suggest a minimum and a maximum of previous noisy function values, respectively. Using the proposed schemes, the generated sequences of step sizes are sequences of discrete random variables. However, they still keep desirable properties for the convergence in a stochastic sense. The almost sure convergence of SA algorithms with the proposed step size schemes is achieved under certain set of assumptions. A special case when the descent direction is a quasi-Newton direction is discussed separately.

The outline of the thesis is as follows. Fundamentals of numerical optimization are given in Chapter 1, while overview of optimization in noisy environment is presented in Chapter 2. The new adaptive step size schemes for SA algorithms are proposed in Chapter 3. Properties of the generated step size sequences are analysed. Convergence theory of the SA algorithms

with the new step size schemes is developed. The results of numerical experiments are given in Chapter 4. They verify efficiency of the proposed algorithms in comparison to classical SA algorithm as well as to other relevant adaptive SA algorithms.

Acknowledgement

During my studies, I have been encouraged and inspired by many people. Here I would like to take an opportunity to express my gratitude for their contribution in my academic development.

Undertaking this PhD would not have been possible without patient guidance, motivation and a lot of useful advice that I received from my advisor, Prof. Dr. Zorana Lužanin. I would like to thank her for believing in me and helping me to grow as a person, teacher and researcher. The person who is equally important for my academic development is Prof. Dr. Nataša Krejić. I am deeply grateful for all invaluable advice, understanding and endless support she has provided me. A very special thanks goes to Prof. Dr. Irena Stojkovska. I am very lucky to have so inspiring and enthusiastic co-author who has cared so much about my work. Moreover, I would like to thank Prof. Dr. Sanja Rapajić for her time, effort and valuable comments regarding this thesis. I am grateful to Prof. Dr. Mirko Savić and Prof. Dr. Andreja Tepavčević for providing me the continuous support during all these years. I would also like to thank Dr. Zoran Ovcin for helping me with Matlab issues.

Finally, I would like to thank my mother Snežana, sister Nataša and Milan for all their love and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Preliminaries on Optimization

> Begin at the beginning, the
> King said gravely, and go on till
> you come to the end: then stop.
>
> Lewis Carroll, Alice in Wonderland

The chapter presents a short summary of numerical optimization. Optimality conditions that verify whether some point is the optimal solution are derived. The most important numerical algorithms for unconstrained optimization are presented and analysed. Also, some basic notations and review of significant results for easier reference are introduced. This chapter mostly relies on [40].

1.1 Problem Statement

Optimization problems are given in a form of minimizing or maximizing a function of one or several variables, possibly subject to constraints on these variables. Therefore, three ingredients are necessary to form optimization

problem: decision variable, objective function that we want to maximize or minimize and the set of constraints that needs to be satisfied. Identifying the objective function, variable and constraints depends on a given problem and represents the first step in the optimization process. It the thesis, we focus only on unconstrained minimization problems. Even though we are considering only a minimization problem, there is no fundamental difference between a minimization and a maximization problem. If we are interested in maximizing the function $f(x)$, this would be equivalent to minimizing the function $-f(x)$. So, it suffices just to think in terms of minimization problems.

The problem that we consider is given by

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function. We assume that $f(x)$ is nonlinear, possibly noncovex and bounded from below.

1.2 Optimality Conditions

A point where the objective function $f(x)$ reaches its lowest value is called a global minimizer of $f(x)$. The formal definition is as follows.

Definition 1 *(Global minimizer) The point $x^* \in \mathbb{R}^n$ is a global minimizer of $f : \mathbb{R}^n \to \mathbb{R}$ if $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$. If this inequality is strict, then x^* is a strict global minimizer.*

According to the Definition 1, obtaining global minimizer requires some information about the function at every point. If it is difficult or impossible to find the global minimizer, then at least we would like to find a point where the function $f(x)$ achieves the smallest value in the open neighbourhood of x^*. Such point is called a local minimizer.

Definition 2 *(Local minimizer) The point $x^* \in \mathbb{R}^n$ is a local minimizer of $f : \mathbb{R}^n \to \mathbb{R}$ if there is an open neighbourhood $\mathcal{B}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathcal{B}$. If this inequality is strict, then x^* is a strict local minimizer.*

It seems that the only way to determine whether the point x^* is a local minimum is to examine all points in its neighbourhood and check whether some of them has smaller function value. Therefore, obtaining local minimum also requires information about the function at an infinite number of points. Since we assume $f(x)$ to be a smooth function, there are more efficient ways to identify local minimum.

The first-order necessary conditions for optimality are the following.

Theorem 1.2.1 *[40] (First-order necessary conditions) If x^* is a local minimizer and $f(x)$ is continuously differentiable in an open neighbourhood $\mathcal{B}$ of x^*, then $\nabla f(x^*) = 0$.*

The point x^* which satisfies the condition $\nabla f(x^*) = 0$ is called a stationary point of the function $f(x)$. Note that $\nabla f(x^*) = 0$ does not necessarily mean that x^* is a local minimizer. According to Theorem 1.2.1, any local minimizer must be a stationary point. The first derivatives do not provide enough information to claim whether the point is a minimizer, thus we need to make use of the second derivatives. We state the second-order necessary conditions below.

Theorem 1.2.2 *[40] (Second-order necessary conditions) If x^* is a local minimizer of $f(x)$ and $\nabla^2 f(x)$ exists and is continuous in an open neighbourhood $\mathcal{B}$ of x^*, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite matrix.*

The second-order sufficient conditions that guarantee that x^* is a local minimizer can be derived.

Theorem 1.2.3 *[40] (Second-order sufficient conditions) Suppose that $\nabla^2 f(x)$ is continuous in an open neighbourhood $\mathcal{B}$ of x^*, $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite. Then, x^* is a strict local minimizer of f.*

If conditions from Theorem 1.2.3 are satisfied, we can say that the point x^* is a local minimizer. Note that the second-order sufficient conditions guarantee that the point is a strict local minimizer. Also, note that the second-order sufficient conditions are not necessary. A point x^* may be a strict local minimizer, and yet may fail to satisfy the sufficient conditions.

1.3 Overview of Algorithms

In most situations, we are not likely to find directly a solution of the problem (1.1). Numerical algorithms for unconstrained optimization are used to find an approximation of the optimal solution. These algorithms are iterative methods. They start from some initial point $x_0 \in \mathbb{R}^n$ and according to a certain iterative rule form a sequence of points $\{x_k\}_{k\in\mathbb{N}}$. Elements of this sequence are called iterates and represent estimates of the optimal solution. The iterative rule is determined by a mapping $\phi : \mathbb{R}^n \to \mathbb{R}^n$ such that $x_{k+1} = \phi(x_k)$. Algorithms often use information about the objective function at the current iterate $f(x_k)$, or sometimes use the sequence history $x_0, ..., x_k$ to generate the next iterate, x_{k+1}. Usually, we wish the mapping $\phi(x)$ to generate a sequence of points such that a decrease in the objective function at each iterate is achieved. Algorithms with this property are called descent. Ideally, the generated sequence $\{x_k\}$ converges to the optimal solution of the problem (1.1).

The initial point also has an important role in the optimization process. If an algorithm generates a sequence that converges to the optimal solution for an arbitrary starting point, then the algorithm is globally convergent. When we are able to localize the solution in some subset, the starting point should be chosen from that subset. In this case, if generated sequence converges to the optimal solution, the algorithm is locally convergent.

In practical implementations, algorithms are finite and use some termination criterion to stop generating a new iterate. They often stop when the problem has been solved with a desired accuracy, or when no further progress can be made. For example, the termination criteria can be reaching the maximum allowable number of iterates, maximum allowable number of function evaluations or gradient norm tolerance.

Through the thesis we consider algorithms of the following type - Algorithm 1.

The existing algorithms of this type differ in the way and on the criteria used to compute the search direction d_k and the step size a_k. In this section we will discuss how to choose a_k and d_k. There are two fundamental strategies for moving from the current iterate x_k to a new iterate x_{k+1}, [40].

Algorithm 1: Directional Search Algorithm

Step 0. Initialization. Specify an initial point $x_0 \in \mathbb{R}^n$.
Set $k = 0$.

Step 1. Direction selection. Determine the search direction d_k.

Step 2. Step size selection. Choose the step size $a_k > 0$.

Step 3. Update iterate. Calculate $x_{k+1} = x_k + a_k d_k$.

Step 4. If some termination criterion is satisfied, then stop.
Else, set $k = k + 1$ and go to Step 1.

These techniques are line search methods and trust region methods. In the thesis, we consider only line search methods. More about trust region methods can be found in [10, 40].

Assume that x_k is the current iterate. The line search method searches for a new iterate x_{k+1} with a lower function value along the line $x_k + ad_k$. So, the algorithm first chooses a search direction d_k from the current point x_k and then computes the step a_k along the direction d_k. The efficiency of the line search method depends on choices of both the direction d_k and the step length a_k.

The ideal choice of the step size a_k is the global minimizer of the one-dimensional minimization problem

$$\min_{a>0} m(a) = f(x_k + ad_k). \tag{1.2}$$

It is often expensive to solve (1.2) exactly, thus algorithms find an approximate solution in practice. In cases where the solution can be found exactly, the methods are called exact line search methods. In practice, we do not want to spend too much time on searching the step size. Typical line search algorithms try out a sequence of candidate values for the step size and accept one of these values when certain conditions are satisfied,

[40].

As stated above, a simple condition we could impose on the step size a_k is to require a reduction in the objective function

$$f(x_k + a_k d_k) < f(x_k). \tag{1.3}$$

The condition (1.3) is satisfied if d_k is a descent direction. The direction d_k is called descent direction if

$$\nabla f(x_k)^T d_k < 0. \tag{1.4}$$

Most line search algorithms require d_k to be descent direction because this property guarantees that the function $f(x)$ can be reduced along this direction, [40]. However, this requirement is not enough to produce convergence. One way to achieve convergence of the line search methods is to make additional assumptions on both, the step size a_k and the direction d_k. Since we want to decrease the function values, the step sizes should be small enough to get sufficient decrease and long enough to make progress. The most often used sufficient decrease condition is the Armijo rule

$$f(x_k + a_k d_k) \leq f(x_k) + \eta a_k \nabla f(x_k)^T d_k, \tag{1.5}$$

where $\eta \in (0, 1)$. In practice η is usually set to 10^{-4}. The curvature condition

$$\nabla f(x_k + a_k d_k)^T d_k \geq c \nabla f(x_k)^T d_k \tag{1.6}$$

where $0 < \eta < c < 1$, ensures that the step size is not too short. The conditions (1.5) and (1.6) together are called the Wolfe conditions. The condition (1.6) can be written as $m'(a_k) \geq m'(0)$. A step length may satisfy the Wolfe conditions without being particularly close to a minimizer of $m(a)$. Obtaining the step size that is in neighbourhood of the stationary point of function $m(a)$, can be done by imposing the strong Wolfe conditions. They consist of the Armijo condition (1.5) and

$$|\nabla f(x_k + a_k d_k)^T d_k| \leq c |\nabla f(x_k)^T d_k|$$

instead of (1.6).

The result of the existence of the step size sequence that satisfies the (strong) Wolfe conditions is given in the next theorem.

Lemma 1.3.1 *[40] Suppose that the function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and let d_k be a descent direction for the function $f(x)$ at the point x_k. Also, suppose that $f(x)$ is bounded below on $\{x_k + ad_k | a > 0\}$. If $0 < \eta < c < 1$, then there exist intervals of step lengths satisfying the (strong) Wolfe conditions.*

Lemma 1.3.1 states that if the objective function $f(x)$ is smooth and bounded below, there exist the step sizes that satisfy the Wolfe conditions.

Now, we discuss requirements on the search direction. Denote by θ_k the angle between the direction d_k and the negative gradient direction $-\nabla f(x_k)$ and define

$$\cos\theta_k = \frac{-\nabla f(x_k)^T d_k}{\|\nabla f(x_k)\|\|d_k\|}.$$

Theorem 1.3.1 *[40] (Zoutendijk theorem) Suppose that the function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on an open set $\mathcal{N}$ containing the level set $\mathcal{L} = \{x \in \mathbb{R}^n | \quad f(x) \leq f(x_0)\}$ where x_0 is the initial iterate. Furthermore, suppose that the gradient $\nabla f(x)$ is Lipschitz continuous on $\mathcal{N}$ and that d_k is a descent search direction. Also, suppose that $f(x)$ is bounded below on $\mathbb{R}^n$ and that the step size a_k satisfies the Wolfe conditions. Then,*

$$\sum_{k\geq 0} \cos^2\theta_k \|\nabla f(x_k)\|^2 < \infty.$$

This result also holds for the strong Wolfe conditions. Zoutendijk theorem implies that

$$\lim_{k\to\infty} \cos^2\theta_k \|\nabla f(x_k)\|^2 = 0.$$

Therefore, if we have a sequence of search directions d_k such that there exists a positive constant δ such that $\cos\theta_k \geq \delta$ for all k, then $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$. In other words, the gradient norms converge to zero, if the search directions are orthogonal with the gradient.

Next subsections are devoted to possible choices of the search direction.

1.3.1 Gradient Descent Algorithm

The most intuitive search direction is the negative gradient

$$d_k = -\nabla f(x_k). \tag{1.7}$$

The Algorithm 1 which uses the negative gradient direction (1.7) is called the Gradient descent algorithm. It is also called the Steepest descent algorithm because along this direction the objective function decreases most rapidly. Since we have assumed that $f(x)$ is smooth, we can use the Taylor expansion to approximate the function value $f(x_k + d)$. Observing only the first two terms of the Taylor series at the point x_k, we have

$$f(x_k + d) \approx f(x_k) + \nabla f(x_k)^T d. \tag{1.8}$$

The idea is to minimize the approximation (1.8) to obtain the search direction. Restricting the direction to be in the unit ball, solution of the problem

$$\min_{||d||=1} \nabla f(x_k)^T d \tag{1.9}$$

represents the unit direction d of the most rapid decrease. The minimum of (1.9) is reached for $d = -\nabla f(x_k)/||\nabla f(x_k)||$ and this direction makes the smallest inner product with the gradient $\nabla f(x_k)$. Consequently, unnormalized direction (1.7) is called the steepest descent direction. The negative gradient direction is orthogonal to the contour of objective function and satisfies descent direction condition unless x_k is a stationary point.

The following theorem ensures that under certain assumptions on $f(x)$, Gradient descent algorithm with exact line search converges regardless of the initial starting point x_0, i.e., it exhibits global convergence.

Theorem 1.3.2 *[15] (Convergence of Gradient Descent Algorithm with Exact Line Search) Suppose that $f(x)$ is continuously differentiable on the set $L = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$, where L is a closed and bounded set. Suppose further that the sequence $\{x_k\}$ is generated by the Gradient descent algorithm with step size a_k obtained by the exact line search. Then, every accumulation point $\overline{x}$ of the sequence $\{x_k\}$ satisfies $\nabla f(\overline{x}) = 0$.*

The Gradient descent algorithm is very applicable. The only cost is the cost of calculating the gradient at current iterate. However, it has several drawbacks. The main drawback is that it can be very slow. The convergence rate is at most linear. Even if we apply the exact line search, we cannot expect improvement of the convergence rate. Note that in terms of the Zountedijak theorem, $\cos\theta_k = 1$ for all k.

There are many examples where the objective function $f(x)$ is expensive and calculating gradient is hard or not available analytically. In these situations, approximations of the true gradient at each iterate are used. These methods are called gradient-free methods. Approximations via finite difference are the most frequently used. For example, the central finite difference estimator of gradient is given by

$$\hat{\nabla} f(x_k) = \begin{bmatrix} \frac{f(x_k+he_1)-f(x_k-he_1)}{2h} \\ \frac{f(x_k+he_2)-f(x_k-he_2)}{2h} \\ \vdots \\ \frac{f(x_k+he_n)-f(x_k-he_n)}{2h} \end{bmatrix},$$

where e_i denotes the vector with 1 on the ith place and zeros elsewhere and $h > 0$. Instead of parameter h, a sequence of parameters $\{h_k\}_{k\in\mathbb{N}}$ which usually tends to zero can be used to obtain more accurate approximation, [10].

1.3.2 Newton's Algorithm

In this subsection, we assume that the objective function is twice continuously differentiable. Then, the objective function can be approximated around the current iterate x_k using the second order Taylor expansion

$$f(x_k + d) \approx f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d. \tag{1.10}$$

Our aim is to minimize the right hand side of (1.10) which is quadratic function of x. Assuming that the Hessian $\nabla^2 f(x_k)$ is nonsingular, differentiating right hand side of (1.10) with respect to x and setting the result

equal to zero, the Newton direction is derived

$$d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k). \tag{1.11}$$

The direction (1.11) is a descent direction if the Hessian $\nabla^2 f(x_k)$ is a positive definite matrix. The Algorithm 1 which uses Newton's direction (1.11) is called Newton's algorithm.

The classical Newton's algorithm does not apply line search. It takes full Newton's step $a_k = 1$ at each iterate.

Now, we state the main convergence result.

Theorem 1.3.3 *[40](Quadratic Convergence of Newton's Algorithm) Suppose that $f(x)$ is twice differentiable and that the Hessian $\nabla^2 f(x)$ is Lipschitz continuous in a neighbourhood of a solution x^* at which the sufficient conditions are satisfied. Consider the sequence of iterates generated by Newton's algorithm, $\{x_k\}$. Then*

(i) if the starting point x_0 is sufficiently close to x^*, the sequence of iterates $\{x_k\}$ converges to x^*;

(ii) the rate of convergence of $\{x_k\}$ is quadratic;

(iii) the sequence of gradient norms $\{||\nabla f(x_k)||\}$ converges quadratically to zero.

1.3.3 Quasi-Newton Algorithm

If we compare the Gradient descent and Newton's algorithm, the following conclusions can be drawn. The Gradient descent algorithm is much simpler than Newton's algorithm because it reduces the computation costs. It is more expensive to evaluate the Hessian of $f(x)$ than the gradient and each Hessian is used to solve only one linear system of equations in Newton's algorithm. Also, when $\nabla^2 f(x)$ is not positive definite, Newton's direction may not even be defined, since $[\nabla^2 f(x)]^{-1}$ may not exist. Even when it is defined, it may not satisfy the descent property in which case it is unsuitable

as a search direction. On the other side, the Gradient descent algorithm has a slower rate of convergence than Newton's algorithm.

In order to overcome the shortcomings of both methods, algorithms that mimic Newton's idea have been proposed. These algorithms use less expensive second-order approximations, but still outperform Gradient descent algorithm. The algorithms are called quasi-Newton algorithms and they are the most widely used for nonlinear optimization problems. There are many different quasi-Newton algorithms, but they are all based on approximating the Hessian by another matrix with lower evaluation and linear algebra costs.

Let us consider the following quadratic model of the objective function at current iterate x_k

$$q_k(d) = f_k + d^T \nabla f_k + \frac{1}{2} d^T B_k d, \tag{1.12}$$

where B_k is a symmetric positive definite approximation of the Hessian $\nabla^2 f(x_k)$ which is updated at each iterate and $f_k = f(x_k)$. The minimizer of (1.12) is called a quasi-Newton direction and it is given by

$$d_k = -B_k^{-1} \nabla f_k. \tag{1.13}$$

At the next iterate $x_{k+1} = x_k + a_k d_k$, the model (1.12) becomes

$$q_{k+1}(d) = f_{k+1} + d^T \nabla f_{k+1} + \frac{1}{2} d^T B_{k+1} d.$$

A logical condition is that ∇q_{k+1} should be equal to the gradient of the objective function at x_{k+1} and x_k. Since $\nabla q_{k+1}(0) = \nabla f_{k+1}$, one condition is already satisfied. The other condition states that

$$\nabla q_{k+1}(-s_k) = \nabla q_{k+1}(-a_k d_k) = \nabla f_{k+1} - a_k B_{k+1} d_k = \nabla f_k.$$

It follows that B_k should satisfy the following equation

$$B_{k+1} s_k = y_k, \tag{1.14}$$

where

$$s_k = x_{k+1} - x_k \quad \text{and} \quad y_k = \nabla f_{k+1} - \nabla f_k.$$

The equation (1.14) is known as the secant equation. As B_{k+1} is symmetric and positive definite, multiplying the secant equation with s_k^T yields to

$$0 < s_k^T B_{k+1} s_k = s_k^T y_k.$$

It follows that s_k and y_k satisfy the curvature condition. This inequality does not hold in general, except for strongly convex functions. When the curvature condition is satisfied, the secant equation always has a solution B_{k+1} but does not provide unique solution. Therefore, additional conditions have to be imposed. Obtaining a unique B_{k+1} can be done by requiring B_{k+1} to be the closest matrix to the current matrix B_k in a norm among all symmetric matrices satisfying the secant equation. That is, B_{k+1} should be a solution to the following problem

$$\min \|B - B_k\| \quad \text{subject to} \quad B^T = B, \quad Bs_k = y_k. \tag{1.15}$$

Depending on the used matrix norm, different updating formulas for B_{k+1} are obtained by solving problem (1.15). Using the weighted Frobenius norm, [40], the Davidon-Fletcher-Powell (DFP) formula for updating approximation of the Hessian is obtained

$$B_{k+1} = (E - \frac{1}{y_k^T s_k} y_k s_k^T) B_k (E - \frac{1}{y_k^T s_k} y_k s_k^T) + \frac{1}{y_k^T s_k} y_k y_k^T$$

where E is the identity matrix.

Using the Sherman-Morrison-Woodbury formula, the inverse Hessian approximation $H_k \approx [\nabla^2 f(x_k)]^{-1}$ can be derived. For the DFP update of B_k, the inverse approximation is the following

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}.$$

There are also other attractive updating formulas. The most used is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula, which is considered to

be the most effective of all quasi-Newton updating formulas. It is derived by solving the following problem

$$\min \|H - H_k\| \quad \text{subject to} \quad H^T = H, \quad s_k = Hy_k.$$

Using the weighted Frobenius norm just like in (1.15) the following unique solution is obtained

$$H_{k+1} = (E - \frac{1}{y_k^T s_k} s_k y_k^T) H_k (E - \frac{1}{y_k^T s_k} y_k s_k^T) + \frac{1}{y_k^T s_k} s_k s_k^T. \tag{1.16}$$

The Algorithm 1 which uses the direction (1.13), where B_k^{-1} is given by (1.16) is called the BFGS algorithm.

Theorem 1.3.4 *[40] (Convergence theorem; BFGS Algorithm) Suppose that $f(x)$ is twice continuously differentiable. The level set $\mathcal{L} = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is convex and there exist positive constants m and M such that*

$$m||z||^2 \leq z^T \nabla^2 f(x) z \leq M||z||^2, \tag{1.17}$$

for all $z \in \mathbb{R}^n$ and $x \in \mathcal{L}$. Let B_0 be any symmetric positive definite initial matrix and let x_0 be a starting point for which (1.17) is satisfied. Then, the sequence $\{x_k\}$ generated by BFGS Algorithm with a_k computed from a line search with Armijo rule converges to the minimizer x^* of $f(x)$.

Theorem 1.3.5 *[40] (Superlinear convergence of the BFGS Algorithm) Suppose that $f(x)$ is twice continuously differentiable and that the Hessian matrix $\nabla^2 f(x)$ is Lipschitz continuous at a minimizer x^*. Suppose also that the sequence $\{x_k\}$ generated by the BFGS algorithm with a_k computed from a line search with Armijo rule converges to x^* and that*

$$\sum_{k=0}^{\infty} ||x_k - x^*|| < \infty$$

holds. Then, $\{x_k\}$ converges to x^* at a superlinear rate.

Chapter 2

Stochastic Approximation

> Creativity is the ability to introduce order into the randomness of nature.
>
> Eric Hoffer

In this chapter, fundamentals of stochastic optimization are presented. Unconstrained minimization problem in noisy environment is introduced and a general framework of stochastic approximation (SA) algorithm as core approach for solving the stated problem is presented. The most important modifications of SA algorithm, based on step size sequence and/or on search directions, are formulated. The most relevant theoretical results are stated.

2.1 Optimization in Noisy Environment

The collection of algorithms for minimizing or maximizing an objective function when uncertainty is involved refers to stochastic optimization. Nowadays, stochastic optimization algorithms have become standard approaches for solving challenging optimization problems.

Even though stochastic optimization refers to all optimization problems with involved randomness, unconstrained optimization problem in noisy en-

vironment is considered in the thesis. In this set-up, true values of the objective function and its gradient are not available, but they are measurable with an error term of stochastic nature. The error term is called stochastic noise or simply noise in the literature. Hence, the objective function and its gradient depend on a random variable and a random vector, respectively, both of them belong to some probability space that might be known or unknown, depending on application.

A noise is present whenever physical system measurements or computer simulations are used for approximations. For example, it is present in problems of estimating quantiles or estimating Markov's chain schemes, i.e., in problems where estimates are formed by Monte Carlo simulations according to a statistical distribution. Furthermore, it arises in problems where data are collected while the system is still operating or in problems where physical data are processed sequentially, with each sequential data point being used to estimate some average criterion, [56].

Let us formulate the problem statement now. We consider the minimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable, possibly nonconvex function bounded below on $\mathbb{R}^n$. Additionally, we assume that the objective function $f(x)$ and its gradient $\nabla f(x) = g(x)$ are disturbed by the noise. Denote by ξ and ε a random variable and a random vector, respectively, defined on a probability space $(\Omega, \mathcal{F}, P)$. Then, noisy observations of the objective function and its gradient are given at each $x \in \mathbb{R}^n$ by

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

where ξ and ε represent the random noise terms. Also, we assume that there is a unique solution $x^* \in \mathbb{R}^n$ of the problem (2.1).

For convenience, we use the following notation. We observe measurements of the objective function and its gradient at current iterate x_k

$$F_k = f_k + \xi_k \quad \text{and} \quad G_k = g_k + \varepsilon_k, \tag{2.2}$$

where $F_k = F(x_k)$, $f_k = f(x_k)$, $G_k = G(x_k)$ and $g_k = g(x_k)$. Note that in this set-up, the noise terms depend on k. It means that we allow noise-generating processes to change with k. In most real applications, noise usually occurs independently, satisfying the classical statistical assumptions of being independent and identically distributed (i.i.d.). Also, it is important to distinguish noisy measurement presented in (2.2) and the term noisy data which is often present in the literature. For instance, consider the least squares or maximum likelihood estimation problems. If we have available only noisy input data, it does not entail noisy measurements of the objective function and/or gradient in the estimation process. These problems are solved on noisy data sets, but the sum of squares and maximum likelihood function are deterministic. This is often called off-line estimation method, [56].

Like in deterministic case, iterative algorithms are used for finding an approximation of optimal solution of the problem (2.1). The presence of noise affects an optimization algorithm throughout the entire process and it might mislead the optimization process which can result in false optimal solution.

2.2 Stochastic Approximation

In this section, we introduce one of the first and most used optimization methods for solving (2.1). The method is known as Stochastic Approximation (SA) algorithm. Originally, SA algorithm is proposed in the pioneer work of Robbins and Monro, [45], for solving the root-finding problem

$$g(x) = 0,$$

where $g : \mathbb{R}^n \to \mathbb{R}^n$ and only noisy measurements of $g(x)$ are available. SA algorithm is also known as Robins-Monro algorithm in the literature. This approach can be utilized for solving the problem (2.1) if the function $g(x)$ is the gradient of the objective function $f(x)$. Thus, SA algorithm represents iterative stochastic optimization algorithm that attempts to find zeroes of a

nonlinear systems of equation or stationary points of functions which cannot be computed directly, but only estimated via noisy observations. The iterative rule of SA algorithm is motivated by the deterministic Gradient descent algorithm. For a given initial approximation x_0, SA algorithm can be written as

$$x_{k+1} = x_k - a_k G_k,\ k = 0, 1, \ldots, \tag{2.3}$$

where $a_k > 0$. The sequence $\{a_k\}$ is called the sequence of step sizes or the gain sequence.

The rich convergence theory has been developed for the SA algorithm. Under suitable conditions, the convergence is achievable in a stochastic sense. Mean square convergence, $E[\|x_k - x^*\|^2] \to 0$ as $k \to \infty$, is established by Robbins and Monro, [45]. A stronger result, almost sure (a.s.) convergence is proved by Chen, [9] and Spall, [56].

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

$$a_k > 0\ \forall k,\ \sum_k a_k = \infty \text{ and } \sum_k a_k^2 < \infty. \tag{2.4}$$

The conditions (2.4) imply that the step size a_k should decay neither too fast nor too slow. The condition $\sum_k a_k = \infty$ requires the step size sequence to approach zero sufficiently slow in order to avoid false convergence of the algorithm. The condition $\sum_k a_k^2 < \infty$ provides sufficiently fast decay of the step size sequence in order to avoid influence of the noise when the iterates are close to the optimal solution.

Denote by $\{x_k\}$ a sequence generated by SA algorithm (2.3) and denote by $\mathcal{F}_k$ the σ-algebra generated by $x_0, x_1, \ldots, x_k$. The set of standard convergence assumptions is the following.

A1 For any $\varepsilon > 0$ there exists a constant $\beta_\varepsilon > 0$ such that

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

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

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

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

A3 There exists a constant $c > 0$ such that

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

Assumption A1 gives a strong condition on the shape of the true gradient $g(x)$. Assumption A2 represents a classical zero-mean noise condition. Under assumption A2, the noisy gradient $G(x)$ is an unbiased estimator of the true gradient $g(x)$. Assumption A3 provides restrictions on the magnitude of $g(x)$, i.e., $\|g(x)\|^2$ and the variance elements of observation noise can not grow faster than a quadratic function of x. By the zero-mean condition A2, the following relation can be obtained from assumption A3

$$E(||G_k||^2|\mathcal{F}_k) \leq c(1 + ||x_k - x^*||^2) \text{ a.s. for all } k.$$

Finally, we state the main convergence result for the SA algorithm.

Theorem 2.2.1 *[9] Assume that A1-A3 hold. Let $\{x_k\}$ be a sequence generated by SA algorithm (2.3), where the step size sequence $\{a_k\}$ satisfies the conditions* (2.4)*. Then, the sequence $\{x_k\}$ converges to x^* a.s. for an arbitrary initial approximation x_0.*

We also state the theorem of Robbins and Siegmund which is used in the proof of the Theorem 2.2.1.

Theorem 2.2.2 *[46] If U_k, β_k, ξ_k and ζ_k, $k = 1, 2, \ldots$ are nonnegative $\mathcal{F}_k$-measurable random variables such that*

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

then on the set $\left\{\sum_k \beta_k < +\infty, \sum_k \xi_k < +\infty\right\}$, U_k converges a.s. to a random variable and $\sum_k \zeta_k < +\infty$ a.s.

The conditions (2.4) are the most relevant conditions from user's input point of view. The step size sequence is crucial for performance of the SA algorithm and it affects the convergence rate. A common example of a sequence that satisfies the conditions (2.4) is the scaled harmonic sequence

$$a_k = \frac{a}{k+1}, \tag{2.5}$$

where $a > 0$. One of the most used sequences is the following generalization of the sequence (2.5)

$$a_k = \frac{a}{(k+1)^\alpha}, \tag{2.6}$$

where $a > 0$ and $0.5 < \alpha \leq 1$. In practical implementations, users have to choose the best values for a and α. Both of these sequences, (2.5) and (2.6), are designed to yield convergence of the SA algorithm. However, these step size sequences make the iterative process quite slow. The step sizes are proportional to $1/k$. Hence, they become small for large k and as a result, the progress is slow.

Under some regularity conditions, the asymptotic normality of iterates of the SA algorithm is proved, [16]. If the step sizes (2.6) are used, the following result can be obtained

$$k^{\frac{\alpha}{2}}(x_k - x^*) \xrightarrow{d} \mathcal{N}(0, \Sigma) \quad k \to \infty,$$

where $\xrightarrow{d}$ denotes convergence in distribution, α governs the decay rate for $\{a_k\}$ and Σ denotes the covariance matrix dependent on the step size sequence $\{a_k\}$ and on the Hessian $H(x)$ of the function $f(x)$. Therefore, iterates $\{x_k\}$ have asymptotic normal distribution with mean x^* and covariance matrix Σ/k^α. The maximum convergence rate is obtained for $\alpha = 1$ when the step size has the standard form (2.6) under conditions (2.4).

Although being optimal, $\alpha = 1$ is not the best choice in the practical implementation. Taking a lower value of α has a superior behaviour in finite time. In many situations, users choose a constant step size with $\alpha = 0$ to avoid small step sizes when k is large, [70]. The lower value of α will provide larger steps when iterates are close to the solution. On the other

hand, a larger a can produce unstable behaviour in early iterates when the denominator is still small. If we take a smaller a, we will ensure stability in earlier iterates, but have a slow progress later.

An asymptotically optimal step size can be obtained by minimizing the covariance matrix Σ and it is given by

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

where $H(x^*)$ is the Hessian of the function $f(x)$ at x^*, [3].

Spall, [55], has suggested introducing a stability constant $A \geq 0$ in the denominator of the step size (2.6) to improve performance of the algorithm

$$a_k = \frac{a}{(k+1+A)^\alpha}. \tag{2.7}$$

Choosing $A > 0$ in (2.7) allows taking a larger a without risking unstable behaviour in the early iterates. A reasonable choice for A is about $5-10\%$ of the total number of expected or allowed iterates.

Due to its simplicity, SA algorithm has become popular among researchers. Various modifications have been proposed to improve optimization process and they are mainly based on the step size selection and/or search direction. The following sections review some of the most important modifications.

2.3 Stochastic Approximation with Descent Direction

In Section 1.3, we have introduced the algorithms which use descent directions. It is impossible to check the condition $g(x_k)^T d_k < 0$ in noisy environment because only noisy measurements of the gradient are available. However, the idea of descent direction can be mimicked. We use the definition of descent direction proposed in [30]. According to the definition, the direction d_k is a descent if

$$G_k^T d_k < 0, \tag{2.8}$$

where G_k is the available noisy measurement given by (2.2). For a given initial approximation x_0, iterates of the descent direction SA algorithm are defined by

$$x_{k+1} = x_k + a_k d_k,\ k = 0, 1, \ldots, \tag{2.9}$$

where d_k satisfies (2.8) and $a_k > 0$. The gradient SA algorithm (2.3) is a special case of the algorithm (2.9).

Convergence of the descent direction SA algorithm (2.9) is also achievable in the stochastic sense. Instead of assumption A1, two additional assumptions on the direction d_k are used.

Let $\{x_k\}$ be a sequence generated by descent direction SA algorithm (2.9) and $\mathcal{F}_k$ the σ-algebra generated by $x_0, x_1, \ldots, x_k$. The additional convergence conditions are the following:

A4 For all k, there exists $c_1 > 0$ such that direction d_k satisfies

$$(x_k - x^*)^T E(d_k|\mathcal{F}_k) \leq -c_1||x_k - x^*|| \text{ a.s.}$$

A5 For all k, there exists $c_2 > 0$ such that

$$||d_k|| \leq c_2||G_k|| \text{ a.s.}$$

The assumption A4 limits the influence of noise on d_k and it is analogous to assumption C4 used in [55]. On the other hand, the assumption A5 connects the available noisy gradient with descent direction. If $d_k = -G_k$, assumption A5 is satisfied with any $c_2 \geq 1$.

Theorem 2.3.1 *[30] Assume that A2-A5 hold. Let $\{x_k\}$ be a sequence generated by descent direction SA algorithm (2.9), where the step size sequence $\{a_k\}$ satisfies the conditions* (2.4)*. Then the sequence $\{x_k\}$ converges to x^* a.s. for an arbitrary initial approximation x_0.*

Similar method, a descent direction form of SA algorithm, is studied by Bertsekas and Tsitsiklis, [4].

2.4 Stochastic Approximation with Line Search

The line search methods can significantly improve the performance of algorithms in deterministic framework. The implementation of the line search methods in noisy environment produces large step sizes, [36, 57, 63, 64]. The large step sizes can cause zig-zag behaviour or even lead the iterative sequence out of the solution's neighbourhood. On the other hand, SA step sizes that satisfy the conditions (2.4) become very small very fast. The larger steps are advantageous in earlier stages when iterates are far away from the optimal solution. The smaller steps are desirable when the iterates are close to the solution, i.e., when the iterates reach some neighbourhood of the solution. The two-phase algorithms that consist of both algorithms, line-search and SA, are proposed in [29, 30]. These algorithms combine SA steps that satisfy conditions (2.4) and steps determined by the line search method. The line search rule is used at the initial stages of the optimization process and SA algorithm is used afterwards. The algorithms are formulated using the negative gradient direction and with the general descent direction.

Let us consider the gradient form of the two-phase algorithm. The algorithm is called Gradient Stochastic Line Search (GSLS) algorithm.

The GSLS algorithm uses the Armijo line search rule adjusted for noisy environment given by

$$F_k(x_k - a_k G_k) \leq F_k - \widehat{c} a_k \|G_k\|^2,$$

where $\widehat{c}$ is a small positive constant. The algorithm switches from the line search to the SA method, if the following inequality is violated

$$\|G_k\| \geq C,$$

where C is some positive constant.

The convergence analysis of the GSLS algorithm consists of two parts. The first part shows that there is a finite number of line search steps. After the line search step is executed in a finite number of steps, the algorithm switches to SA steps almost surely. The almost sure convergence of the proposed method is ensured due to infinitely many SA consecutive steps.

Algorithm 2: Gradient Stochastic Line Search (GSLS) Algorithm

Step 0. Choose $x_0 \in \mathbb{R}^n$, $\widehat{c} \in (0,1)$, $C, \delta(C) > 0$, and $\{a_k\}$ that satisfies (2.4). Set $k = 0$ and $p = 1$.

Step 1. Calculate G_k.

Step 2. If $p = 1$ then calculate 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

$$F_k(x_k - \alpha G_k) \leq F_k - \widehat{c}\alpha\|G_k\|^2$$

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.

The additional convergence assumptions are the following.

A6 The gradient $g(x)$ is Lipschitz continuous, i.e., there exists a positive constant $L > 0$ such that

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

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 $x \in \mathbb{R}^n$.

Assumption A7 is similar to the one used by Wardi, [64].

Theorem 2.4.1 *[29] Suppose that assumptions A1-A3 and A6-A7 hold and that the Hessian $\nabla f^2(x^*)$ exists and is nonsigular. Let*

$$C = \max\{\frac{4(1-\widehat{c})}{\underline{\alpha}\widehat{c}}, \frac{M + 2\sqrt{2ML} + 1}{1-\widehat{c}}\},$$

where

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

Then, the sequence $\{x_k\}$ generated by GSLS algorithm converges a.s. to x^*.

The two phases algorithm which uses a descent direction is called Descent stochastic line search (DSLS) algorithm, [30].

The authors use the descent direction given by (2.8) and the line search rule

$$F_k(x_k + a_k d_k) \leq F_k + \widetilde{c} a_k G_k^T d_k,$$

where $\widetilde{c}$ is a small positive constant.

The convergence analysis of DDLS algorithm is analogous to the analysis of GSLS algorithm. Two additional assumptions, A8 and A9, common to the descent direction method in deterministic optimization, are imposed.

Algorithm 3: Descent Direction Line Search (DDLS) Algorithm

Step 0. Choose $x_0 \in \mathbb{R}^n$, $\widetilde{c} \in (0,1)$, $C, \underline{\delta}(C) > 0$, and $\{a_k\}$ that satisfies (2.4). Set $k = 0$ and $p = 1$.

Step 1. Take d_k such that (2.8) holds.

Step 2. Select a_k.

Step 2.1. If $p = 1$ go to Step 2.2, else to Step 2.3.

Step 2.2. If $||G_k|| \geq C$ chose $a_k > \underline{\delta}(C)$ such that (2.10) holds. Go to Step 3. Else, p=2.

Step 2.3. Take a_k from the predefined gain sequence.

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

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

A8 There exists a positive constant δ such that

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

A9 There exists a positive constant $\underline{\Delta} \in (0, \Delta)$ such that

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

The main convergence theorem is the following.

Theorem 2.4.2 *[30] Suppose that assumptions A2-A9 hold. Let*

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

where

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

Then, the sequence $\{x_k\}$ generated by DDLS algorithm converges a.s. to x^*.

2.5 Stochastic Approximation with Adaptive Step Sizes

2.5.1 Accelerated Stochastic Approximation

One of the first algorithms with an adaptive step size scheme is Accelerated SA algorithm. It is introduced by Kesten, [24], for problems in one dimension and extended to multidimensional case by Delyon and Judicky, [11]. Kesten's idea is that frequent changes of the sign of the difference $x_{k+1} - x_k = a_k G_k$ indicate that the current iterate is near the optimal solution x^*. In this case, a smaller step size in the next iterate is proposed. If the changes are not frequent, a larger step size should be used in the next iterate.

The iterative sequence of the Accelerated SA algorithm is generated by the iterative rule of the gradient SA algorithm (2.3) with the step sizes $\{a_k\}$ defined by

$$z_k = z_{k-1} + I(G_k^T G_{k-1} < 0)$$

$$a_k = a(z_k) \quad k = 1, 2, \ldots \tag{2.10}$$

where $z_0 = 0$, $I(\cdot)$ stands for an indicator function and $a(\cdot)$ is some deterministic sequence.
For example, the following step size sequence

$$a_k = \frac{a}{z_k + 1} \tag{2.11}$$

can be used, where $a > 0$.

An almost sure convergence of the Accelerated SA algorithm is established under a certain set of assumptions, [11].

The most important issue in establishing convergence of the Accelerated SA algorithm is to show that infinitely many sign changes occur, i.e., that $z_k \to \infty$ when $k \to \infty$. The authors have obtained an estimate of the convergence rate of $\frac{z_k}{k}$ which stands for the change of the sign frequency of $G_k^T G_{k-1}$

$$\lim_{k\to\infty} \frac{z_k}{k} - P(\varepsilon_k^T \varepsilon_{k-1}) \to 0 \text{ a.s.}$$

Under additional conditions, asymptotic normality of the Accelerated SA algorithm for the special choice of the step sizes (2.11) is established

$$\sqrt{k}(x_k - x^*) \xrightarrow{d} \mathcal{N}(0, V) \quad k \to \infty,$$

where V is unique positive solution of certain Lyapunov equation, [11]. Thus, the Accelerated SA algorithm is an asymptotic equivalent to the SA algorithm.

2.5.2 Switching Stochastic Approximation

Kesten's idea of adjusting the step sizes at each iterate has recently been modified by Xu and Dai, [68]. The switching step size scheme based on the quantity $\frac{z_k}{k}$ for the gradient SA algorithm (2.3) is proposed. The suggested step sizes are random variables that satisfy the condition (2.4) almost surely.

The switching step size scheme is defined by

$$a_k = \begin{cases} \frac{a}{(k+1+A)^{\alpha}}, & l_k \geq v \\ \frac{a}{(k+1+A)^{\beta}}, & l_k < v \end{cases}, \tag{2.12}$$

where $l_k = |\frac{z_k}{k} - P(\varepsilon_1^T \varepsilon_2 < 0)|$, $0.5 \leq \alpha < \beta \leq 1$ and v is a small positive constant.

According to (2.12), a relatively small value of l_k indicates that the iterates are close to the solution and that a smaller step size should be used in the next iterate. If the value of l_k is relatively large, the iterates are far away from the optimal solution and a larger step size should be used. The dividing criterion l_k is formed using the probability $P(\varepsilon_1^T \varepsilon_2 < 0)$. In practice, this probability is unknown. Since it represents probability and belongs to the $[0, 1]$, authors propose taking $\frac{1}{2}$. When distribution of $\varepsilon_1^T \varepsilon_2$ is symmetric around zero, then $P(\varepsilon_1^T \varepsilon_2 < 0) = \frac{1}{2}$. The authors show that if $g(x_k) \to 0$, then $l_k \to 0$ in L^2. Almost sure convergence of l_k is still an open problem.

The gradient SA algorithm generated by (2.3) with the step sizes scheme (2.12) is called the Switching SA algorithm. Almost sure convergence is proved under assumptions A1-A3 and the following additional assumption on the noise terms

A10 $\{\varepsilon_k\}$ is a sequence of i.i.d. continuous random variables which are independent of x_k and g_k.

We state the main convergence results.

Theorem 2.5.1 *[68] Let assumptions A1-A3 and A10 hold. Then, the sequence $\{x_k\}$ generated by Switching SA algorithm with the step sizes (2.12) converges a.s. to x^*.*

The step size choice can be improved by taking

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

where $q(\frac{z_k}{k})$ is some function of $\frac{z_k}{k}$. Here, the step size a_k changes together with the dividing criterion l_k at each iterate. The authors have also proposed

$$q(\frac{z_k}{k}) = \max\,[1 - |\frac{z_k}{k} - \frac{1}{2}|, 0.501]$$

and

$$q(\frac{z_k}{k}) = \min\,[1, \frac{z_k}{k} + 0.501].$$

The quantity $q(\frac{z_k}{k})$ is large when the iterates are far away from the optimal solution. Otherwise, $q(\frac{z_k}{k})$ is small when the iterates are close to the optimum.

Chapter 3

Stochastic Approximation with New Adaptive Step Sizes

> When it is obvious that the goals cannot be reached, don't adjust the goals, adjust the action steps.
>
> Confucius

In this chapter, a new class of adaptive step size schemes for the SA algorithms is introduced. The two main schemes are introduced and their properties are derived. The schemes are based on the previously observed noisy function values. The convergence theory of SA algorithms with the new schemes is developed.

3.1 Preliminaries

For convenience, let us formulate the main problem again. We consider the following minimization problem in noisy environment

$$\min_{x\in\mathbb{R}^n} f(x), \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable, nonlinear and possibly nonconvex function bounded below on $\mathbb{R}^n$. A unique solution $x^* \in \mathbb{R}^n$ of (3.1) exists. We assume that observations of the objective function and its gradient disturbed by the noise

$$F(x) = f(x) + \xi \quad \text{and} \quad G(x) = g(x) + \varepsilon, \tag{3.2}$$

are available for all $x \in \mathbb{R}^n$, where ξ and ε represent the random noise terms defined on a probability space $(\Omega, \mathcal{F}, P)$. Moreover, we make an additional assumption on the noise terms $\{\xi_k\}$

A11 $\{\xi_k\}$ is a sequence of i.i.d. continuous random variables with a common probability density function (pdf), $p(y) > 0$ a.s. $\forall y \in \mathbb{R}$.

From the formulation of problem (3.1) and according to the definition of noisy function value $F(x)$, (3.2), the noise terms $\xi_k, k = 0, 1, 2, \ldots$. are already identically distributed. One example of the noise terms that satisfy A11 is a sequence of i.i.d. normal random variables.

In this chapter, a new class of adaptive step size schemes which are based on the fixed number of previously observed noisy function values is presented. At each iterate, using the proposed schemes, interval estimations of the optimal objective function are constructed. The bounds of the intervals are used to determine whether the objective function has been improved. If the current objective function value is larger than the upper interval bound, we declare the iterate as unsuccessful. A zero step size is used in the next iterate. If the function value is smaller than the lower bound of the interval, we declare the iterate as successful one and propose a larger step size in the next iterate. In this manner, we will ensure a

faster progress of the algorithm and avoid small steps especially when the number of iterates gets large. In other words, the schemes avoid using the step sizes proportional to $1/k$ when it is expected that the larger steps will improve the process. If the function value lies in the interval, the step size is obtained by a harmonic rule.

We introduce two main adaptive step size schemes that can be applied to both, gradient and descent direction, SA algorithms. The first scheme estimates optimal function value at each iterate by forming a confidence-like interval for the optimal function values. The interval bounds are shifted means of the previously observed noisy function values. This scheme can be generalized by using a convex combination of the previously observed noisy function values instead of the mean. The second scheme uses Extreme Value Statistics to form the intervals, i.e., maximum and minimum of the previously observed noisy function values as the interval bounds.

The SA algorithms with the proposed step size schemes require an additional measurement, F_k, at each iterate compared to the standard SA algorithms. However, we believe that tracking the objective function values may considerably improve the knowledge of the optimization process. The similar reasoning that using the observed function values to accept or reject steps can improve the algorithm's stability is discussed in [56, 66]. This is also a feature by which our algorithms differ from the Accelerated and Switching SA algorithms. So, the additional measurement at each iterate might be sometimes a good decision, as our numerical results will demonstrate. On the other hand, the proposed schemes also might be good choice for derivative-free settings, when we can only rely on the noisy functional values. In this case, the gradient will be approximated using only functional values, for example with finite differences. We did not consider this case in our numerical experiments, since we suppose that the noisy gradient measurements are known.

3.2 Mean-Sigma Stochastic Approximation

In this section, we present the first adaptive step size scheme.

3.2.1 Step Size Scheme

As mentioned above, we monitor the previously observed function values to get an insight into whether the objective function is improving. In the kth iterate, we construct a confidence-like interval

$$J_k = (\frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} - \sigma, \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} + \sigma),$$

using $m(k)$ previously observed (noisy) function values $F_{k-1}, F_{k-2}, ..., F_{k-m(k)}$, where $m(k) = \min\{k, m\}$. If the observed (noisy) function value in kth iterate F_k, is smaller than the lower bound of the interval, we consider this scenario as a good one, i.e., we consider that a sufficient decrease of the objective function is achieved. In this case, we propose taking a larger step size in the next $(k+1)$th iterate. Inspired by [42] we chose $a_k = a\theta^{s_k}$, which for large k and large θ, remains large in comparison to step size of the form (2.7). We can still obtain properties of the sequence $\{a_k\}$ suitable for convergence analysis. As it will be demonstrated later, we recommend taking θ close to 1. Note that θ is the key parameter in controlling the length of the step size when good scenario occurs. The step size $a_k = a\theta^{s_k}$ with $\theta \cong 1$ will produce longer steps than steps of SA form while the iterates are far away from the solution, but also when the number of iterates becomes large. This can be suitable when we believe that there is strong influence of the noise. If F_k is greater than the upper limit of the interval, we reject the current iterate. Zero step size is used, as implemented, for example in [66]. Otherwise, if F_k lies in the interval, we propose a small safe step size of the form similar to the classical SA step size (2.7).

The formal formulation of the adaptive step size scheme is the following

$$a_k = \begin{cases} a\theta^{s_k}, & F_k < \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} - \sigma \\ 0, & F_k > \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} + \sigma, \\ \frac{a}{(t_k+1+A)^\alpha}, & otherwise \end{cases} \tag{3.3}$$

where $m(k) = \min\{k, m\}$, $\sigma > 0$ and

- $m \in \mathbb{N}$, $\theta \in (0,1)$, $a > 0$, $A \geq 0$, $0.5 < \alpha \leq 1$,
- $s_k = s_{k-1} + I\left\{F_k < \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} - \sigma\right\}$ for $k = 1, 2, \ldots$ and $s_0 = 0$,
- $t_k = t_{k-1} + I\left\{\frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} - \sigma \leq F_k \leq \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} + \sigma\right\}$ for $k = 1, 2, \ldots$, and $t_0 = 0$.

The scheme (3.3) is called mean-sigma step size scheme. SA algorithm (2.9) with the steps generated by the mean-sigma adaptive step size scheme (3.3) is called Mean-Sigma SA algorithm.

Algorithm 4: Mean-Sigma SA Algorithm

Step 0. Choose $x_0 \in \mathbb{R}^n$, $m \in \mathbb{N}$, $\sigma > 0$, $\theta \in (0,1)$, $a > 0$, $A \geq 0$ and $0.5 < \alpha \leq 1$. Set $k = 0$.

Step 1. Choose d_k such that (2.8) holds.

Step 2. Calculate F_k and select a_k according to the criterion (3.3).

Step 3. Calculate $x_{k+1} = x_k + a_k d_k$.

Step 4. If some termination criterion is satisfied then stop. Else, set $k = k + 1$ and go to Step 1.

A special case of Algorithm 4 is when a negative noisy gradient is chosen as the search direction, i.e., $d_k = -G_k$.

The inspiration for intervals J_k is drawn from the interval estimation theory. If the observed function value F_k is considered as an estimate of the optimal function value $f^* = f(x^*)$, then the sequence of the observed function values $F_{k-1}, F_{k-2}, ..., F_{k-m(k)}$ can be considered as its sample of length $m(k)$. The interval J_k can be viewed as a confidence-like interval for the expected optimal function value f^*, since it is symmetrical around the

sample mean $\frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j}$. Therefore, if the next estimate F_k of f^* is in the interval J_k, we decide to proceed with slow but safe steps.

The convergence of Mean-Sigma SA algorithm is established for an arbitrary constant $\sigma > 0$. In practical implementation, the choice of the constant σ is closely related to the noise level. It can be easily shown that in the case of an i.i.d. white noise with variance σ^2, i.e., $E(\xi_k) = 0$ and $Var(\xi_k) = \sigma^2$, for all k, the mean-square error (MSE) of the function value estimator F_k of the optimal value f^* is equal to $\sigma^2 + (f_k - f^*)^2$, where $f_k = f(x_k)$ is the true function value at x_k. Now, since the variance of the sampling distribution of F_k is often approximated reasonably well by MSE of F_k, [28], it is justified to relate the noise level σ to constant in the interval J_k. Although the noise level may not be known, in many real phenomenon the order of magnitude of the noise is known. For example, in physical measurements, it is usually the error of the measuring instrument. In cases when there is no information about the magnitude, procedures that estimate noise are applied first.

3.2.2 Properties of the Adaptive Step Size Sequence

In this subsection, we will show that the sequence $\{a_k\}$ generated by the mean-sigma adaptive step size scheme (3.3) satisfies the conditions (2.4) a.s. under assumptions A11.

The mean-sigma scheme (3.3) generates a sequence of random variables. The distribution of the step size a_k is the following

$$a_k : \begin{pmatrix} 0 & \frac{a}{(t_k+1+A)^\alpha} & a\theta^{s_k} \\ p_k^1 & p_k^2 & p_k^3 \end{pmatrix},$$

where

$$p_k^1 = P(a_k = 0) = P(F_k > \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} + \sigma),$$

$$\begin{aligned} p_k^2 &= P(a_k = \frac{a}{(t_k+1+A)^\alpha}) \\ &= P(\frac{1}{m(k)}\sum_{j=1}^{m(k)} F_{k-j} - \sigma \le F_k \le \frac{1}{m(k)}\sum_{j=1}^{m(k)} F_{k-j} + \sigma), \end{aligned}$$

$$p_k^3 = P(a_k = a\theta^{s_k}) = P(F_k < \frac{1}{m(k)}\sum_{j=1}^{m(k)} F_{k-j} - \sigma),$$

and $p_k^1 + p_k^2 + p_k^3 = 1$. The probabilities p_k^1, p_k^2 and p_k^3 cannot be derived explicitly, not even in the i.i.d. case, i.e., when $f_k = \frac{1}{m(k)}\sum_{j=1}^{m(k)} f_{k-j}$. Also, they depend on the distribution of the noise terms.

Let us denote by A_k the event that $m(k)$ consecutive zero steps occur

$$A_k = \left\{a_{k-1} = a_{k-2} = \ldots = a_{k-m(k)} = 0\right\}. \tag{3.4}$$

Lemma 3.2.1 *Assume that A11 holds. Let the step size sequence $\{a_k\}$ be defined by the mean-sigma step size scheme (3.3). Then, for $k = 1, 2, \ldots$ and $m \in \mathbb{N}$ the following inequality holds*

$$P(A_k) > 0, \tag{3.5}$$

where A_k is defined by (3.4).

Proof. This lemma states that $m(k)$ consecutive zero steps occur with nonzero probability. We will prove it by assuming the contrary, that there exists $k \in \mathbb{N}$ such that

$$0 = P(A_k) = P(F_{k-i} > \frac{1}{m(k)}\sum_{j=1}^{m(k)} F_{k-i-j} + \sigma, i = 1, 2, \ldots, m(k)).$$

Consider the events $\left\{F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} + \sigma\right\}, i = 1, 2 \ldots, m(k)$. Obviously,

$$\left\{F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} + \sigma\right\} \subseteq \left\{F_{k-i} > \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-i-j} + \sigma\right\},$$

for $i = 1, 2 \ldots, m(k)$ which further implies

$$\bigcap_{i=1}^{m(k)} \left\{F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} + \sigma\right\} \subseteq \bigcap_{i=1}^{m(k)} \left\{F_{k-i} > \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-i-j} + \sigma\right\}.$$

Thus, we obtain

$$P(\bigcap_{i=1}^{m(k)} F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} + \sigma) \le P(\bigcap_{i=1}^{m(k)} F_{k-j} > \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-i-j} + \sigma).$$

Consequently,

$$P(F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} + \sigma,\ i = 1, 2, ..., m(k))$$
$$\le P(F_{k-i} > \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-i-j} + \sigma,\ i = 1, 2, ..., m(k)).$$

We have

$$P(F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} + \sigma, i = 1, 2, \ldots, m(k)) = 0.$$

Let us define *δ-neighbourhood of the optimal value* $f^* = f(x^*)$. We say that y is in δ-neighbourhood of the optimal value f^* if $|y - f^*| < \delta$, where $\delta > 0$. Denote by $B^k_{\frac{\delta}{2}}$ the event

$$B^k_{\frac{\delta}{2}} = \left\{f_{k-i} \text{ is in } \frac{\delta}{2} - \text{neighbourhood of } f^*,\ i = 1, 2, \ldots, 2m(k)\right\}.$$

Now, we chose $\delta > 0$ such that

$$P(B^k_{\frac{\delta}{2}}) > 0. \tag{3.6}$$

Note that such $\delta > 0$ exists. For example, we can take

$$\delta = 2 \cdot \max_{1 \le i \le 2m(k)} |f_{k-i} - f^*| + 1.$$

For this choice of δ, we have $P(B^k_{\frac{\delta}{2}}) = 1$.
Now,

$$\begin{aligned} 0 &= P\left(F_{k-i} > \max_{1 \le j \le m(k)} F_{k-i-j} + \sigma, i = 1, 2, \ldots, m(k)\right) \\ &\ge P\left(F_{k-i} > \max_{1 \le j \le m(k)} F_{k-i-j} + \sigma, i = 1, 2, \ldots, m(k) \Big| B^k_{\frac{\delta}{2}}\right) P(B^k_{\frac{\delta}{2}}). \end{aligned} \tag{3.7}$$

So, (3.6) and (3.7) imply

$$P\left(F_{k-i} > \max_{1 \le j \le m(k)} F_{k-i-j} + \sigma,\ i = 1, 2, \ldots, m(k) \Big| B^k_{\frac{\delta}{2}}\right) = 0.$$

Under the realization of the event $B^k_{\frac{\delta}{2}}$, it can be shown that

$$f_{k-i} - \delta < f_{k-j} < f_{k-i} + \delta, \tag{3.8}$$

for all $i, j = 1, 2, ..., 2m(k)$. Now, using (3.8), under the realization of the event $B^k_{\frac{\delta}{2}}$, the inequality

$$\xi_{k-i} > \max_{1 \le j \le m(k)} \xi_{k-i-j} + \sigma + \delta$$

implies

$$F_{k-i} > \max_{1 \le j \le m(k)} F_{k-i-j} + \sigma,$$

and this is true for any $i = 1, 2, \ldots, m(k)$. Therefore,

$$\begin{aligned} 0 &= P\left(F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} + \sigma,\ i = 1, 2, \ldots, m(k) \middle| B^k_{\frac{\delta}{2}}\right) \\ &\ge P\left(\xi_{k-i} > \max_{1\le j\le m(k)} \xi_{k-i-j} + \sigma + \delta,\ i = 1, 2, \ldots, m(k) \middle| B^k_{\frac{\delta}{2}}\right) \\ &= P\left(\xi_{k-i} > \max_{1\le j\le m(k)} \xi_{k-i-j} + \sigma + \delta,\ i = 1, 2, \ldots, m(k)\right), \end{aligned} \tag{3.9}$$

since the last conditional probability is independent of the condition. Relation (3.9) implies

$$P\left(\xi_{k-i} > \max_{1\le j\le m(k)} \xi_{k-i-j} + \sigma + \delta,\ i = 1, 2, \ldots, m(k)\right) = 0.$$

Now,

$$\begin{aligned} 0 &= P\left(\xi_{k-i} > \max_{1\le j\le m(k)} \xi_{k-i-j} + \sigma + \delta, i = 1, 2, \ldots, m(k)\right) \\ &= P\left(\xi_{k-i} > \xi_{k-i-j}, i, j = 1, 2, \ldots, m(k)\right) \\ &\ge P\left(\xi_{k-1} > \xi_{k-2} + \sigma + \delta > \ldots > \xi_{k-2m(k)} + (2m(k) - 1)(\sigma + \delta)\right) \\ &= I(\sigma + \delta). \end{aligned} \tag{3.10}$$

On the other hand,

$$\begin{aligned} I(\sigma + \delta) &= \int_{-\infty}^{\infty} p(x_{k-1})dx_{k-1} \int_{-\infty}^{x_{k-1}-(\sigma+\delta)} p(x_{k-2})dx_{k-2} \cdots \\ &\qquad \int_{-\infty}^{x_{k-2m(k)+1}-(2m(k)-1)(\sigma+\delta)} p(x_{k-2m(k)})dx_{k-2m(k)} > 0 \end{aligned}$$

almost surely for all $\delta > 0$, since $p(x) > 0$ a.s. by A11, and $I(\delta)$ is a decreasing function with

$$\lim_{\delta\to 0} I(\delta) = \frac{1}{(2m(k))!} \quad \text{and} \quad \lim_{\delta\to+\infty} I(\delta) = 0,$$

which is a contradiction with (3.10). Therefore, the relation (3.5) holds for all k. ■

Under realization of the event A_k we have that $f_k = \frac{1}{m(k)} \sum_{j=1}^{m(k)} f_{k-j}$. Now, when we know that $m(k)$ consecutive zero steps occur with nonzero probability, we can state the following lemma for conditional distribution of the step size a_k.

Lemma 3.2.2 *Assume that A11 holds. Let the step size sequence $\{a_k\}$ be defined by the mean-sigma step size scheme (3.3). Then, for all $k = 1, 2, \ldots$*

$$P(a_k = 0|A_k) > 0,$$

$$P(a_k = a\theta^{s_k}|A_k) > 0$$

and

$$P(a_k = \frac{a}{(t_k + 1 + A)^\alpha}|A_k) > 0,$$

where A_k is the event defined by (3.4). Moreover, for all $k = 1, 2, ...$

$$P(a_k = 0) > 0.$$

$$P(a_k = a\theta^{s_k}) > 0$$

and

$$P(a_k = \frac{a}{(t_k + 1 + A)^\alpha}) > 0.$$

Proof. First note that the conditional probabilities are well defined because of the Lemma 3.2.1. Under the realization of the event A_k we have that $f_k = \frac{1}{m(k)} \sum_{j=1}^{m(k)} f_{k-j}$.

Let us start with the first inequality $P(a_k = 0|A_k) > 0$. According to the step size rule (3.3) we have

$$\begin{aligned} P(a_k = 0|A_k) &= P(F_k > \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} + \sigma|A_k) \\ &= P(f_k + \xi_k > \frac{1}{m(k)} \sum_{j=1}^{m(k)} (f_{k-j} + \xi_{k-j}) + \sigma|A_k) \\ &= P(\xi_k > \frac{1}{m(k)} \sum_{j=1}^{m(k)} \xi_{k-j} + \sigma|A_k) \\ &= P(\xi_k - \frac{1}{m(k)} \sum_{j=1}^{m(k)} \xi_{k-j} > \sigma), \end{aligned} \tag{3.11}$$

since the conditional probability is independent of the condition. Let us define Y_k by

$$Y_k = \xi_k - \frac{1}{m(k)} \sum_{j=1}^{m(k)} \xi_{k-j},$$

and let $p_{Y_k}(\cdot)$ be its pdf. We can think of Y_k as a difference of two random variables, ξ_k with pdf $p(\cdot)$ and $Z_{k,m(k)} = \frac{1}{m(k)} \sum_{j=1}^{m(k)} \xi_{k-j}$ with pdf $p_{k,m(k)}(\cdot)$. By the convolution formula for two independent random variables X and Y, the pdf of their sum $X + Y$ is

$$p_{X+Y}(z) = \int_{-\infty}^{\infty} p_Y(z - t) p_X(t) dt, \tag{3.12}$$

where $p_X(\cdot)$ is pdf of X, and $p_Y(\cdot)$ is pdf of Y. Now, using (3.12) we can derive recursively the distribution of the random variable $Z_{k,m(k)}$, since ξ_k are all independent random variables, by A11. The derived pdf $p_{k,m(k)}(\cdot)$ is always positive because it only depends on $p(\cdot)$ which is, by A11, always positive. The pdf of Y_k is

$$p_{Y_k}(y) = \int_{-\infty}^{\infty} p(t) p_{k,m(k)}(y - t) dt,$$

and it is always positive, since $p(\cdot)$ and $p_{k,m(k)}(\cdot)$ are always positive. Therefore, by (3.11), we have

$$P(a_k = 0|A_k) = P(Y_k > \sigma) = \int_{\sigma}^{\infty} p_{Y_k}(y)dy > 0. \tag{3.13}$$

Similarly, we have

$$P(a_k = a\theta^{s_k}|A_k) = P(Y_k < -\sigma) = \int_{-\infty}^{-\sigma} p_{Y_k}(y)dy > 0 \tag{3.14}$$

and

$$P(a_k = \frac{a}{(t_k+1+A)^{\alpha}}|A_k) = P(-\sigma \leq Y_k \leq \sigma) = \int_{-\sigma}^{\sigma} p_{Y_k}(y)dy > 0, \tag{3.15}$$

since $\sigma > 0$. Additionally, from Lemma 3.2.1 and (3.13)-(3.15), for all $k = 1, 2, ...$ we have

$$P(a_k = 0) \geq P(a_k = 0|A_k) \cdot P(A_k) > 0,$$

$$P(a_k = a\theta^{s_k}) \geq P(a_k = a\theta^{s_k}|A_k) \cdot P(A_k) > 0$$

and

$$P(a_k = \frac{a}{(t_k+1+A)^{\alpha}}) \geq P(a_k = \frac{a}{(t_k+1+A)^{\alpha}}|A_k) \cdot P(A_k) > 0,$$

which completes the proof. ∎

The previous lemma leads to the important result which is stated below.

Lemma 3.2.3 *Assume that A11 holds. Let the step size sequence $\{a_k\}$ be defined by the mean-sigma step size scheme (3.3). Then, there are infinitely many steps $a_k = \frac{a}{(t_k+1+A)^{\alpha}}$ and infinitely many steps $a_k = a\theta^{s_k}$ almost surely.*

Proof. Let us first consider the sequence of events $T_k = \left\{a_k = \frac{a}{(t_k+1+A)^\alpha}\right\}$, $k = 1, 2, \ldots$. Define $\{T_k \text{ i.o.}\}$ as the event that an infinite number of events T_k, $k = 1, 2, \ldots$ occur. The abbreviation i.o. means for infinitely often. We will show that the event $\{T_k \text{ i.o.}\}$ occurs almost surely, i.e.,

$$P(\{T_k \text{ i.o.}\}) = P(\{w | w \in T_k \text{ for infinitely many } k \in \{1, 2, \ldots\}\}) = 1. \tag{3.16}$$

Let us consider the subsequence $\left\{T_{k(m+1)}\right\}_k$ of the sequence $\{T_k\}_k$. It is a sequence of independent events, because they depend on different random variables ξ_k, which are independent by A11. Analogously, we define the event $\{T_{k(m+1)} \text{ i.o.}\}$ as the event that an infinite number of events $T_{k(m+1)}$, $k = 1, 2, \ldots$ occur. The event $\{T_{k(m+1)} \text{ i.o.}\}$ is a member of the σ-algebra $\bigcap_{k=1}^{\infty} \left\{\sigma(T_{n(m+1)}), n \geq k\right\}$. Therefore, we can apply the Kolmogorov $0-1$ law which states that σ-algebra $\bigcap_{k=1}^{\infty} \left\{\sigma(T_{n(m+1)}), n \geq k\right\}$ contains only events of probability 0 or 1, [13]. According to the Kolmogorov $0-1$ law

$$P(\{T_{k(m+1)} \text{ i.o.}\}) \in \{0, 1\}. \tag{3.17}$$

Let us assume that

$$P(\{T_{k(m+1)} \text{ i.o.}\}) = 0.$$

Because of the inclusion

$$\bigcap_{k=1}^{\infty} T_{k(m+1)} \subseteq \{T_{k(m+1)} \text{ i.o.}\},$$

we have that

$$P\left(\bigcap_{k=1}^{\infty} T_{k(m+1)}\right) \leq P(\{T_{k(m+1)} \text{ i.o.}\}),$$

which together with (3.18), imply

$$P\left(\bigcap_{k=1}^{\infty} T_{k(m+1)}\right) = 0. \tag{3.18}$$

As we mentioned before, $T_{k(m+1)}$, $k = 1, 2, \ldots$ are independent events, so (3.18) is equivalent to

$$\prod_{k=1}^{\infty} P(T_{k(m+1)}) = 0,$$

which implies that there exists $k_0 \in \mathbb{N}$ such that $P(T_{k_0(m+1)}) = 0$, i.e., $P(a_{k_0} = \frac{a}{(t_{k_0}+1+A)^{\alpha}}) = 0$, which is in contradiction to Lemma 3.2.2. Therefore,

$$P(\{T_{k(m+1)} \text{ i.o.}\}) > 0. \tag{3.19}$$

The relation (3.19) together with (3.17), imply

$$P(\{T_{k(m+1)} \text{ i.o.}\}) = 1. \tag{3.20}$$

Now, because of the inclusion

$$\{T_{k(m+1)} \text{ i.o.}\} \subseteq \{T_k \text{ i.o.}\},$$

we have that

$$P(\{T_{k(m+1)} \text{ i.o.}\}) \leq P(\{T_k \text{ i.o.}\}).$$

The last inequality together with (3.20) imply (3.16), i.e., almost surely there are infinitely many steps $a_k = \frac{a}{(t_k+1+A)^{\alpha}}$. Analogously, we can show that almost surely there are infinitely many steps $a_k = a\theta^{s_k}$, which completes the proof. ∎

Remark 3.2.1 *As a consequence of Lemma 3.2.3 we have that almost surely infinitely many consequtive steps $a_k = 0$ cannot occur, since almost surely there are infinitely many nonzero steps. This finding will help us during the practical implementation. We can impose a correction condition and limit the number of consecutive zero steps in the following way. If there is some predefined number of successive steps $a_k = 0$, then in the next iterate we are going to take a nonzero safe step of the form (2.7).*

Theorem 3.2.1 *Assume that A11 holds. Then, the step size sequence* $\{a_k\}$ *defined by the mean-sigma step size scheme (3.3) satisfies the conditions (2.4) almost surely.*

Proof. Let us denote events C and D by $C = \{k | F_k < \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} - \sigma\}$ and $D = \{k | \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} - \sigma \leq F_k \leq \frac{1}{m(k)} \sum_{j=1}^{m(k)} F_{k-j} + \sigma\}$. By the definition of the sequence $\{a_k\}$, (3.3), the following relations hold a.s.

$$\sum_k a_k = \sum_{k \in C} a\theta^{s_k} + \sum_{k \in D} \frac{a}{(t_k + 1 + A)^\alpha} = \infty,$$

and

$$\sum_k a_k^2 = \sum_{k \in C} (a\theta^{s_k})^2 + \sum_{k \in D} (\frac{a}{(t_k + 1 + A)^\alpha})^2 < \infty.$$

The first relations hold since there are infinitely many steps $a_k = \frac{a}{(t_k+1+A)^\alpha}$ by Lemma 3.2.3. The second relation holds because there are infinitely many steps $a_k = a\theta_k^s$, also by Lemma 3.2.3. So, the step size sequence $\{a_k\}$ satisfies the conditions (2.4) a.s. ∎

We will conclude this section with a short discussion about the parameter σ in the step size scheme (3.3). In each iterate, we construct a confidence-like interval J_k with the sample size $m(k)$. Instead of a standard deviation of the mean of $m(k)$ previously observed noisy function values, standard deviation of the noise terms ξ is taken. Note that

$$\begin{aligned} \frac{1}{m(k)} \sum_{j=1}^{m(k)} [F_{k-j} - E(F_{k-j})]^2 &= \frac{1}{m(k)} \sum_{j=1}^{m(k)} [f_{k-j} + \xi_{k-j} - E(f_{k-j} + \xi_{k-j})]^2 \\ &= \frac{1}{m(k)} \sum_{j=1}^{m(k)} [\xi_{k-j} - E(\xi_{k-j})]^2. \end{aligned} \tag{3.21}$$

Now, taking expectation of (3.21) we have

$$\begin{aligned} E[\frac{1}{m(k)}\sum_{j=1}^{m(k)}[F_{k-j}-E(F_{k-j})]^2] &= E[\frac{1}{m(k)}\sum_{j=1}^{m(k)}[\xi_{k-j}-E(\xi_{k-j})]^2] \\ &= \frac{1}{m(k)}\sum_{j=1}^{m(k)}E[[\xi_{k-j}-E(\xi_{k-j})]^2] \\ &= \frac{1}{m(k)}\sum_{j=1}^{m(k)}D[\xi_{k-j}] \\ &= \sigma^2. \end{aligned} \tag{3.22}$$

We can interpret the result (3.22) in the following way. If we consider F_{k-j} as an estimator of the true function value f_{k-j}, then the mean value of any $m(k)$ consecutive mean square errors (MSE) of the objective function is equal to the variance of noise ξ.

3.2.3 Generalization of the Mean-Sigma Scheme

The mean-sigma step size scheme (3.3) can be extended to allow using information from the previous iterates in a more general way, [33]. The proposed generalized step size scheme allows past function values to have different influence on the selection of a new step size length. It allows constructing the bigger steps when a sufficient decrease in the objective function is monitored. Let x_k be the current iterate. We wish to determine the step size a_k for the next iterate. Denote by $\sum_{j=1}^{m(k)}\lambda_{k,j}F_{k-j}$ a convex combination of $m(k)$ previous noisy function values F_{k-1}, F_{k-2}, ..., $F_{k-m(k)}$, where $m(k)=\min\{k,m\}$, $m\in\mathbb{N}$ and $\lambda_{k,j}\geq\lambda>0,\ j=1,2,\ldots,m(k)$ such that $\sum_{j=1}^{m(k)}\lambda_{k,j}=1$, for all k.

The generalized mean-sigma step size scheme is given by

$$a_k=\begin{cases} b\theta^{s_k}, & F_k<\sum_{j=1}^{m(k)}\lambda_{k,j}F_{k-j}-\sigma \\ 0, & F_k>\sum_{j=1}^{m(k)}\lambda_{k,j}F_{k-j}+\sigma, \\ \frac{a}{(t_k+1+A)^\alpha}, & otherwise \end{cases} \tag{3.23}$$

where

- $m(k) = \min\{k, m\}$, $m \in \mathbb{N}$, $\sigma > 0$, $\theta \in (0, 1)$, $b \geq a > 0$, $A \geq 0$, $0.5 < \alpha \leq 1$,
- $\lambda_{k,j} \geq \lambda \geq 0, \quad j = 1, \ldots, m(k)$ such that $\sum_{j=1}^{m(k)} \lambda_{k,j} = 1$,
- $s_k = s_{k-1} + I\left\{F_k < \sum_{j=1}^{m(k)} \lambda_{k,j} F_{k-j} - \sigma\right\}$, for $k = 1, 2, \ldots$, and $s_0 = 0$,
- $t_k = t_{k-1} + I\left\{\sum_{j=1}^{m(k)} \lambda_{k,j} F_{k-j} - \sigma \leq F_k \leq \sum_{j=1}^{m(k)} \lambda_{k,j} F_{k-j} + \sigma\right\}$, for $k = 1, 2, \ldots$, and $t_0 = 0$.

Adaptive step sizes defined by the scheme (3.23) differ from the steps generated by the mean-sigma scheme (3.3) in the expression $\sum_{j=1}^{m(k)} \lambda_{k,j} F_{k-j}$ which allows previous function values to be taken with different weights at each iterate. In this way, the step size scheme (3.23) can use more effectively the information about the optimization process stored in previous function values. Another advantage is that the bigger step sizes can be taken when a sufficient decrease in the objective function is observed.

The step sizes generated by (3.23) have the same properties as the steps generated by (3.3). The SA algorithm (2.9) with the step sizes generated by (3.23) is called CC-Adaptive SA algorithm. In the rest of the thesis, we focus on the Mean-Sigma SA algorithm, but the same conclusions can be drawn for the CC-Adaptive SA Algorithm.

3.3 Min-Max Stochastic Approximation

In this section, the second adaptive step size scheme is presented.

3.3.1 Step Size Scheme

The mean-sigma scheme (3.3) estimates the optimal function value at each iterate by forming the interval J_k using the previously observed noisy function values. In order to enhance the interval estimate, an approach that

Algorithm 5: CC-Adaptive SA Algorithm

Step 0. Choose $x_0 \in \mathbb{R}$, $\sigma > 0$, $m \in \mathbb{N}$, $\theta \in (0,1)$, $b \geq a > 0$, $A \geq 0$, $0.5 < \alpha \leq 1$ and $\lambda > 0$. Set $k = 0$.

Step 1. Choose d_k such that (2.8) holds and $\lambda_{k,j} \geq \lambda > 0,\ j = 1, \ldots, m(k)$ such that $\sum_{j=1}^{m(k)} \lambda_{k,j} = 1$.

Step 2. Calculate F_k and select a_k according to the criterion (3.23).

Step 3. Calculate $x_{k+1} = x_k + a_k d_k$.

Step 4. If some termination criterion is satisfied then stop. Else, set $k = k + 1$ and go to Step 1.

has a direct insight into whether the objective function is improving is suggested. We propose using the minimum and the maximum of $m(k)$ previous noisy function values instead of the shifted mean. Therefore, in each iterate the following interval is constructed

$$\tilde{J}_k = (\min_{1 \leq j \leq m(k)} F_{k-j}, \max_{1 \leq j \leq m(k)} F_{k-j}).$$

This approach allows a new step only if there is relatively strong statistical evidence of the improvement of the objective function. The formal rule of the new step size scheme is

$$a_k = \begin{cases} a\theta^{s_k}, & F_k < \min_{1 \leq j \leq m(k)} F_{k-j} \\ 0, & F_k > \max_{1 \leq j \leq m(k)} F_{k-j}, \\ \frac{a}{(t_k+1+A)^\alpha}, & otherwise \end{cases} \tag{3.24}$$

where $m(k) = \min\{k, m\}$, and

- $m \in \mathbb{N}$, $\theta \in (0,1)$, $a > 0$, $A \geq 0$, $0.5 < \alpha \leq 1$,
- $s_k = s_{k-1} + I\left\{F_k < \min_{1 \leq j \leq m(k)} F_{k-j}\right\}$ for $k = 1, 2, \ldots$ and $s_0 = 0$,

- $t_k = t_{k-1} + I\left\{\min_{1\le j\le m(k)} F_{k-j} \le F_k \le \max_{1\le j\le m(k)} F_{k-j}\right\}$ for $k = 1, 2, \ldots$, and $t_0 = 0$.

The scheme (3.24) is called min-max step size scheme. SA algorithm (2.9) with the steps generated by the min-max adaptive step size scheme (3.24) are called Min-Max SA algorithm.

Algorithm 6: Min-Max SA Algorithm

Step 0. Choose $x_0 \in \mathbb{R}^n$, $m \in \mathbb{N}$, $\theta \in (0,1)$, $a > 0$, $A \ge 0$ and $0.5 < \alpha \le 1$. Set $k = 0$.

Step 1. Choose d_k such that (2.8) holds.

Step 2. Calculate F_k and select a_k according to the criterion (3.24).

Step 3. Calculate $x_{k+1} = x_k + a_k d_k$.

Step 4. If some termination criterion is satisfied then stop. Else, set $k = k + 1$ and go to Step 1.

According the min-max scheme (3.24), if F_k is larger than the maximum of $m(k)$ previously observed function values, the step $a_k = 0$ is taken in the next iterate. If F_k is smaller than the minimum of $m(k)$ previously observed function values, we suggest step size $a_k = a\theta^{s_k}$ in the next iterate. Otherwise, if F_k is inside $\tilde{J}_k$, we propose a backup step size similar to the step size (2.7), substituting k with t_k which counts the mentioned events.

Remark 3.3.1 *Note that bounds of the interval $\tilde{J}_k$, events $\left\{F_k < \min_{1\le j\le m(k)} F_{k-j}\right\}$ and $\left\{F_k > \max_{1\le j\le m(k)} F_{k-j}\right\}$ are modifications of the lower and upper records statistics $\{F_k < \min_{1\le j\le k-1} F_j\}$ and $\{F_k > \max_{1\le j\le k-1} F_j\}$, respectively. The difference is that bounds of the interval $\tilde{J}_k$ consist only of $m(k)$ previous random variables F_{k-j}. The record statistics arise in many areas such as climatology, sports, medicine, traffic, industry and they are very popular among researches, [39]. The*

records theory is developed only when $F_j, j = 1, \ldots, k$ are i.i.d. random variables. In this case, both, upper and lower, record statistics are mutually independent and infinitely many of them occur. These results do not hold for the bounds of $\tilde{J}_k$ because the bounds are not i.i.d. Theory of non i.i.d. records is only developed for the special case when there is a certain linear trend among the variables.

3.3.2 Properties of the Adaptive Step Size Sequence

We will show that under assumption A11, the step size sequence $\{a_k\}$, defined by the min-max scheme (3.24), satisfies the conditions (2.4) almost surely.

Rewriting (3.24), the step size a_k has the following discrete distribution

$$a_k : \begin{pmatrix} 0 & \frac{a}{(t_k+1+A)^\alpha} & a\theta^{s_k} \\ p_k^1 & p_k^2 & p_k^3 \end{pmatrix},$$

where

$$p_k^1 = P(a_k = 0) = P(F_k > \max_{1\le j\le m(k)} F_{k-j}),$$

$$p_k^2 = P(a_k = \frac{a}{(t_k+1+A)^\alpha}) = P(\min_{1\le j\le m(k)} F_{k-j} \le F_k \le \max_{1\le j\le m(k)} F_{k-j})$$

and

$$p_k^3 = P(a_k = a\theta^{s_k}) = P(F_k < \min_{1\le j\le m(k)} F_{k-j}).$$

Since the step sizes generated by (3.24) are discrete random variables, our first step is to determine the distribution of the step sizes and the frequency of the events $\{F_k > \max_{1\le j\le m(k)} F_{k-j}\}$, $\{F_k < \min_{1\le j\le m(k)} F_{k-j}\}$ and $\{\min_{1\le j\le m(k)} F_{k-j} \le F_k \le \max_{1\le j\le m(k)} F_{k-j}\}$. Note that we only need infinitely many events $\{\min_{1\le j\le m(k)} F_{k-j} \le F_k \le \max_{1\le j\le m(k)} F_{k-j}\}$ to satisfy (2.4).

Lemma 3.3.1 *If the noise terms ξ_k are i.i.d. continuous random variables and $f_k = f_{k-j},\ j = 1, \ldots, m(k)$, then the following inequalities hold*

$$P(F_k > \max_{1\le j\le m(k)} F_{k-j}) = \frac{1}{m(k)+1},$$

$$P(F_k < \min_{1\le j\le m(k)} F_{k-j}) = \frac{1}{m(k)+1},$$

$$P(\min_{1\le j\le m(k)} F_{k-j} \le F_k \le \max_{1\le j\le m(k)} F_{k-j}) = \frac{m(k)-1}{m(k)+1}.$$

Proof. Let us denote by $\Phi(x)$ the cumulative distribution function (cdf) of any random variable ξ_k. If we denote by $\Phi_j^k(x)$ the cdf of the random variable F_{k-j}, then from $F_{k-j} = f_{k-j} + \xi_{k-j}$, we have that

$$\begin{aligned}\Phi_j^k(x) &= P(F_{k-j} \le x) = P(f_{k-j} + \xi_{k-j} \le x) = P(\xi_{k-j} \le x - f_{k-j}) \\ &= \Phi(x - f_{k-j}). \end{aligned} \tag{3.25}$$

Denote by $\Phi_{(m(k))}^k(x)$ the cdf of the random variable $\max_{1\le j\le m(k)} F_{k-j}$. The i.i.d. property of the noise terms implies that $F_{k-j},\ j = 1, ..., m(k)$ are also independent continuous random variables, so the equality (3.25) and the assumption $f_k = f_{k-j},\ j = 1, \ldots, m(k)$ imply that

$$\begin{aligned}\Phi_{(m(k))}^k(x) &= P(\max_{1\le j\le m(k)} F_{k-j} \le x) = P(F_{k-1} \le x, \cdots, F_{k-m(k)} \le x) \\ &= P(F_{k-1} \le x) \cdots P(F_{k-m(k)} \le x) = \Phi_1^k(x) \cdots \Phi_{m(k)}^k(x) \\ &= \Phi(x - f_{k-1}) \cdots \Phi(x - f_{k-m(k)}) = (\Phi(x - f_k))^{m(k)}. \end{aligned} \tag{3.26}$$

For any two independent continuous random variables X and Y with cdfs $\Phi_X(x)$ and $\Phi_Y(x)$ respectively, the probability of the event $\{X > Y\}$ can be expressed as

$$P(X > Y) = \int_{-\infty}^{+\infty} \Phi_Y(x)\Phi'_X(x)dx. \tag{3.27}$$

So, (3.25)-(3.27) and the independence of the random variables F_k and $\max_{1\leq j\leq m(k)} F_{k-j}$ imply that

$$\begin{aligned} P(F_k > \max_{1\leq j\leq m(k)} F_{k-j}) &= \int_{-\infty}^{+\infty} (\Phi(x-f_k))^{m(k)} \Phi'(x-f_k)dx \\ &= \int_0^1 y^{m(k)} dy = \frac{1}{m(k)+1}, \end{aligned}$$

since $\Phi(x)$ is a cdf and $\lim_{x\to-\infty}\Phi(x) = 0$ and $\lim_{x\to+\infty}\Phi(x) = 1$. Similarly, it can be derived that

$$P(F_k < \min_{1\leq j\leq m(k)} F_{k-j}) = \frac{1}{m(k)+1}$$

and finally,

$$P(\min_{1\leq j\leq m(k)} F_{k-j} \leq F_k \leq \max_{1\leq j\leq m(k)} F_{k-j}) = 1 - \frac{2}{m(k)+1} = \frac{m(k)-1}{m(k)+1},$$

which completes the proof. ∎

Remark 3.3.2 *If the noise terms are i.i.d. continuous random variables and if there are $m(k)$ consecutive zero steps $a_{k-1} = a_{k-2} = \ldots = a_{k-m(k)} = 0$, then $x_k = x_{k-1} = x_{k-2} \ldots = x_{k-m(k)}$, so $f_k = f_{k-j}$ for $j = 1, \ldots, m(k)$. Therefore, Lemma 3.3.1 holds.*

Remark 3.3.2 helps us to recognize the importance of having $m(k)$ consecutive zero steps, i.e., the importance of the event $A_k = \{a_{k-1} = a_{k-2} = \ldots = a_{k-m(k)} = 0\}$. Our next step is to investigate the probability of having $m(k)$ consecutive zero steps.

Lemma 3.3.2 *Assume that A11 holds. Let the step size sequence $\{a_k\}$ be defined by the min-max step size scheme (3.24). Then, for $k = 1, 2, \ldots$ and $m \in \mathbb{N}$, the following inequality holds*

$$P(A_k) > 0, \tag{3.28}$$

where $A_k = \{a_{k-1} = a_{k-2} = \ldots = a_{k-m(k)} = 0\}$.

Proof. Assume the contrary that there exists $k \in \mathbb{N}$ such that

$$P(A_k) = 0.$$

It follows

$$\begin{aligned}
0 &= P(A_k) \\
&= P(\bigcap_{i=1}^{m(k)} \left\{ F_{k-i} > \max_{1\le j\le m(k)} F_{k-i-j} \right\}) \\
&= P(F_{k-1} > \max_{1\le j\le m(k)} F_{k-1-j}, \ldots, F_{k-m(k)} > \max_{1\le j\le m(k)} F_{k-m(k)-j}) \\
&= P(F_{k-1} > F_{k-2} > \ldots > F_{k-m(k)} > \max_{1\le j\le m(k)} F_{k-m(k)-j}) \\
&\geq P\left(F_{k-1} > F_{k-2} > \ldots > F_{k-m(k)} > \ldots > F_{k-2m(k)}\right).
\end{aligned}$$

Therefore, we have

$$P\left(F_{k-1} > F_{k-2} > \ldots > F_{k-m(k)} > \ldots > F_{k-2m(k)}\right) = 0.$$

Similarly, like in the proof of Lemma 3.2.1, let $B^k_{\frac{\delta}{2}}$, denote the event

$$B^k_{\frac{\delta}{2}} = \left\{ f_{k-j} \text{ is in } \frac{\delta}{2} - \text{neighbourhood of the } f^*,\ j = 1, \ldots, 2m(k) \right\}.$$

We chose $\delta > 0$ such that

$$P(B^k_{\frac{\delta}{2}}) > 0. \tag{3.29}$$

Now,

$$\begin{aligned}
0 &= P\left(F_{k-1} > F_{k-2} > \ldots > F_{k-m(k)} > \ldots > F_{k-2m(k)}\right) \\
&\geq P\left(F_{k-1} > F_{k-2} > \ldots > F_{k-2m(k)} | B^k_{\frac{\delta}{2}}\right) P(B^k_{\frac{\delta}{2}}).
\end{aligned} \tag{3.30}$$

So, from (3.29) and (3.30) we obtain

$$P\left(F_{k-1} > F_{k-2} > \ldots > F_{k-m(k)} > \ldots > F_{k-2m(k)} | B^k_{\frac{\delta}{2}}\right) = 0. \qquad (3.31)$$

However, if $f_{k-j},\ j = 1, 2, \ldots, 2m(k)$ are in a $\frac{\delta}{2}-$neighbourhood of the optimal value f^*, then we have

$$|f_{k-j} - f_{k-i}| \leq |f_{k-j} - f^*| + |f^* - f_{k-i}| < \frac{\delta}{2} + \frac{\delta}{2} = \delta,$$

for all $j, i = 1, 2, \ldots, 2m(k)$ and

$$f_{k-i} - \delta < f_{k-j} < f_{k-i} + \delta.$$

Under the realization of the event $B^k_{\frac{\delta}{2}}$, the inequalities

$$\xi_{k-j} > \xi_{k-j-1} + \delta,\ j = 1, 2, \ldots, 2m(k) - 1$$

imply that for $j = 1, 2, \ldots, 2m(k) - 1$

$$\begin{aligned} F_{k-j} &= f_{k-j} + \xi_{k-j} > f_{k-j} + \xi_{k-j-1} + \delta \\ &> f_{k-j-1} + \xi_{k-j-1} = F_{k-j-1}. \end{aligned}$$

So,

$$\begin{aligned} &P\left(F_{k-j} > F_{k-j-1},\ j = 1, 2, \ldots, 2m(k) - 1 | B^k_{\frac{\delta}{2}}\right) \geq \\ &P\left(\xi_{k-j} > \xi_{k-j-1} + \delta,\ j = 1, 2, \ldots, 2m(k) - 1 | B^k_{\frac{\delta}{2}}\right). \qquad (3.32) \end{aligned}$$

Now, (3.31) and (3.32) imply that

$$P\left(\xi_{k-j} > \xi_{k-j-1} + \delta,\ j = 1, 2, \ldots, 2m(k) - 1 | B^k_{\frac{\delta}{2}}\right) = 0. \qquad (3.33)$$

Since the conditional probability in (3.33) is independent of the condition, we can rewrite relation (3.33) as

$$P\left(\xi_{k-j} > \xi_{k-j-1} + \delta,\ j = 1, 2, \ldots, 2m(k) - 1\right) = 0. \qquad (3.34)$$

Note that

$$\begin{aligned} I(\delta) &= P\left(\xi_{k-j} > \xi_{k-j-1} + \delta,\ j = 1, 2, \ldots, 2m(k) - 1\right) \\ &= P\left(\xi_{k-1} > \xi_{k-2} + \delta > \xi_{k-3} + 2\delta > \ldots > \xi_{k-2m(k)} + (2m(k) - 1)\delta\right) \\ &= \int_{-\infty}^{\infty} p(x_{k-1})dx_{k-1} \int_{-\infty}^{x_{k-1}-\delta} p(x_{k-2})dx_{k-2} \ \cdots \\ &\quad \int_{-\infty}^{x_{k-2m(k)+1}-(2m(k)-1)\delta} p(x_{k-2m(k)})dx_{k-2m(k)} > 0 \end{aligned} \tag{3.35}$$

almost surely for all $\delta > 0$ since $p(x) > 0$ a.s. by A11. Moreover, $I(\delta)$ is a decreasing function, with

$$\lim_{\delta \to 0} I(\delta) = \frac{1}{(2m(k))!} \quad \text{and} \quad \lim_{\delta \to +\infty} I(\delta) = 0.$$

The relation (3.35) is in contradiction to (3.34). Therefore, (3.28) holds for all k. ∎

Now, when we know that $m(k)$ consecutive zero steps occur with nonzero probability, we can show that there is nonzero probability of occurring each of the steps $a_k = a\theta_k^s$, $a_k = 0$ and $a_k = \frac{a}{(t_k+1+A)^\alpha}$ at each iterate k.

Lemma 3.3.3 *Assume that A11 holds. Let the step size sequence* $\{a_k\}$ *be defined by the min-max step size scheme (3.24). Then, for all* $k = 1, 2, \ldots$

$$P(a_k = a\theta^{s_k}) > 0,\ P(a_k = 0) > 0 \ \textit{and}\ P(a_k = \frac{a}{(t_k + 1 + A)^\alpha}) > 0.$$

Proof. From Remark 3.3.2 and Lemma 3.3.1, it follows

$$P(a_k = a\theta^{s_k}) \geq P(a_k = a\theta^{s_k}|A_k) \cdot P(A_k) = \frac{1}{m(k) + 1} \cdot P(A_k) > 0,$$

$$P(a_k = 0) \geq P(a_k = 0|A_k) \cdot P(A_k) = \frac{1}{m(k) + 1} \cdot P(A_k) > 0$$

and

$$\begin{aligned} P(a_k = \frac{a}{(t_k+1+A)^\alpha}) &\geq P(a_k = \frac{a}{(t_k+1+A)^\alpha}|A_k) \cdot P(A_k) \\ &= \frac{m(k)-1}{m(k)+1} \cdot P(A_k) > 0. \end{aligned}$$

Note that the conditional probabilities $P(\cdot|A_k)$ are well defined because of Lemma 3.2.2. ∎

Lemma 3.3.3 ensures that infinitely many nonzero steps occur almost surely.

Lemma 3.3.4 *Assume that A11 holds. Let the step size sequence $\{a_k\}$ be defined by the min-max step size scheme (3.24). Then, there are infinitely many steps $a_k = a\theta^{s_k}$ and infinitely many steps $a_k = \frac{a}{(t_k+1+A)^\alpha}$ almost surely.*

Proof. The proof is analogous to the proof of Lemma 3.2.3. ∎

Remark 3.3.3 *Analogous conclusion as in Remark 3.3.2 holds. Almost surely infinitely many consecutive steps $a_k = 0$ cannot occur, since almost surely there are infinitely many nonzero steps.*

Lemma 3.3.4 ensures that the step size sequence $\{a_k\}$ satisfies the conditions (2.4).

Theorem 3.3.1 *Assume that A11 holds. Then, the step size sequence $\{a_k\}$ defined by the min-max step size scheme (3.24) satisfies the conditions (2.4).*

Proof. The proof is analogous to the proof of Theorem 3.2.1. ∎

3.4 Convergence Analysis

In this section, we establish the convergence of the proposed algorithms. The case of negative gradient and the case of arbitrarily descent direction are discussed separately.

The SA convergence theorems, Theorem 2.2.1 and Theorem 2.3.1, assume deterministic step sizes $\{a_k\}$ that satisfy the conditions (2.4). In order to use these results when the step sizes a_k are stochastic, we need to assume the following. The steps a_k are $\mathcal{F}_k$-measurable, where $\mathcal{F}_k$ is the σ-algebra generated by $x_0, x_1, x_2, \dots x_k$, and $\{x_k\}$ is a sequence generated by the corresponding algorithm. This assumption is similar to the assumption in [49]. Moreover, the step size conditions (2.4) are satisfied almost surely in this case. Under these additional assumptions, the SA convergence theorems, Theorem 2.2.1 and Theorem 2.3.1, also hold when the step sizes a_k are stochastic.

Theorem 3.4.1 *Assume that A2-A5 and A11 hold. Let $\{x_k\}$ be a sequence generated by Mean-Sigma SA, Min-Max SA or CC-Adaptive SA algorithm. Then, the sequence $\{x_k\}$ converges to x^* a.s. for an arbitrary initial approximation x_0.*

Corollary 3.4.1 *Assume that A1-A3 and A11 hold. Let $\{x_k\}$ be a sequence generated by Mean-Sigma, Min-Max or CC-Adaptive SA algorithm with $d_k = -G_k$. Then, the sequence $\{x_k\}$ converges to x^* a.s. for an arbitrary initial approximation x_0.*

Theorem 3.4.1 and Corollary 3.4.1 also hold for the SA algorithms with the generalized scheme (3.23).

3.5 Quasi-Newton Stochastic Approximation

Let us consider the SA algorithm (2.9). Assume that a quasi-Newton direction $d_k = -B_k^{-1}G_k$ is chosen as the search direction. The quasi-Newton

directions might yield inaccurate and unstable information in noisy environment. One possible remedy is to impose the following condition on B_k

$$\text{for all } k \geq 1,\ B_k \text{ depends only on } (k-1)\text{th sample set.} \tag{3.36}$$

This is already successfully tested in [30, 52, 69]. Note that if the condition (3.36) holds, because of the zero-mean assumption A2, we have

$$\begin{aligned} E[d_k|\mathcal{F}_k] &= E[-B_k^{-1}G_k|\mathcal{F}_k] = -B_k^{-1}E[G_k|\mathcal{F}_k] = -B_k^{-1}E[g_k+\varepsilon_k|\mathcal{F}_k] \\ &= -B_k^{-1}g_k. \end{aligned}$$

Assumption A4 can be rewritten as

A4' for all k there exists $c_1' > 0$ such that

$$(x_k - x^*)^T B_k^{-1} g_k \geq c_1' ||x_k - x^*|| \text{ a.s.}$$

Another condition on B_k that we impose is the following

$$\text{there exist } \mu_1, \mu_2 > 0 \text{ such that for all } k \geq 1,\ \mu_1 E \preceq B_k^{-1} \preceq \mu_2 E. \tag{3.37}$$

Here, E denotes the $n \times n$ identity matrix, and notation $M \preceq N$ means that the matrix $N - M$ is positive semidefinite. If the condition (3.37) is satisfied, then we have

$$||d_k|| = ||-B_k^{-1}G_k|| \leq \mu_2 ||G_k||,$$

so assumption A5 holds.

Now, we state the convergence theorem for a descent direction method with a quasi-Newton direction and SA step sizes, that follows from previous discussion and Theorem 2.3.1.

Theorem 3.5.1 *Assume that A2, A3 and A4' hold. Let $\{x_k\}$ be a sequence generated by (2.9), with the step size sequence $\{a_k\}$ satisfying the conditions (2.7). A quasi-Newton direction $d_k = -B_k^{-1}G_k$ is chosen with B_k that satisfies conditions (3.36) and (3.37). Then the sequence $\{x_k\}$ converges to x^* a.s. for an arbitrary initial approximation x_0.*

Corollary 3.5.1 *Assume that A2, A3, A4' and A11 hold. Let* $\{x_k\}$ *be a sequence generated Mean-Sigma SA, Min-Max SA or CC-Adaptive SA algorithm where* $d_k = -B_k^{-1}G_k$ *is a quasi-Newton direction. Assume that* B_k *satisfies the conditions (3.36) and (3.37). Then the sequence* $\{x_k\}$ *converges to* x^* *a.s. for an arbitrary initial approximation* x_0.

Convergence of Mean-Sigma, Min-Max or CC-Adaptive SA algorithms with a quasi-Newton direction can be also established under slightly different assumptions. We will remove assumption A4', and we impose similar assumption to assumption A3.

A3' There exists a constant $c' > 0$ such that

$$E(||G_k||^2|\mathcal{F}_k) \leq c' \text{ a.s. for all } k.$$

Assumption A3' is also used in [8] for minimization problems arising in supervised machine learning, where the objective function is strongly convex. To establish a convergence after these changes, an assumption on Lipschitz continuity of the gradient $g(x)$ is needed (assumption A6).

Now, we can formulate the second convergence theorem for a descent direction method with a quasi-Newton direction and SA step sizes.

Theorem 3.5.2 *Assume that A2, A3' and A6 hold. Let* $\{x_k\}$ *be a sequence generated by (2.9), where* $d_k = -B_k^{-1}G_k$ *is a quasi-Newton direction, and* B_k *satisfies the conditions (3.36) and (3.37). Then, for an arbitrary initial approximation* x_0,

$$\lim_{k\to\infty} ||g_k|| = 0 \text{ a.s.}$$

Proof. For each k and some $t \in (0,1)$, since $x_{k+1} = x_k + a_k d_k$, we have

$$\begin{aligned} f(x_{k+1}) &= f(x_k) + a_k g(x_k + t a_k d_k)^T d_k \\ &= f(x_k) + a_k g_k^T d_k + a_k (g(x_k + t a_k d_k) - g_k)^T d_k \\ &\leq f(x_k) + a_k g_k^T d_k + a_k ||g(x_k + t a_k d_k) - g_k|| \cdot ||d_k||. \end{aligned}$$

Assumption A6 and $t \in (0,1)$ imply that

$$f(x_{k+1}) \leq f(x_k) + a_k g_k^T d_k + a_k^2 L||d_k||^2.$$

For $d_k = -B_k^{-1}G_k$, because of the condition (3.37), we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - a_k g_k^T B_k^{-1} G_k + a_k^2 L||-B_k^{-1}G_k||^2 \\ &\leq f(x_k) - a_k g_k^T B_k^{-1} G_k + a_k^2 L\mu_2^2 ||G_k||^2. \end{aligned}$$

It holds because of (3.36) and (3.37) and assumptions A2 and A3'. Now, taking the conditional expectation with respect to $\mathcal{F}_k$, we have

$$\begin{aligned} E[f(x_{k+1})|\mathcal{F}_k] &\leq f(x_k) - a_k g_k^T E[B_k^{-1}G_k|\mathcal{F}_k] + a_k^2 L\mu_2^2 E[||G_k||^2|\mathcal{F}_k] \\ &\leq f(x_k) - a_k g_k^T B_k^{-1} g_k + a_k^2 L\mu_2^2 E[||G_k||^2|\mathcal{F}_k] \\ &\leq f(x_k) - a_k \mu_1 ||g_k||^2 + a_k^2 L\mu_2^2 c'. \end{aligned}$$

Substracting $f^* = f(x^*)$ from the both sides we obtain

$$E[f(x_{k+1}) - f^*|\mathcal{F}_k] \leq f(x_k) - f^* - a_k\mu_1||g_k||^2 + a_k^2 L\mu_2^2 c',$$

where $U_k = f(x_k) - f^*$ are nonnegative random variables. Using that $\sum_k a_k^2 < \infty$, from the Theorem of Robbins and Siegmund, we have that U_k converges a.s. to a random variable U and $\sum_k a_k\mu_1||g_k||^2 < \infty$ a.s. Because of $\sum_k a_k = \infty$, we have that $\lim_{k\to\infty}||g_k|| = 0$ a.s. which completes the proof. ∎

Finally, the second convergence result for algorithms with the proposed step size schemes follows.

Corollary 3.5.2 *Assume that A2, A3', A6 and A11 hold. Let $\{x_k\}$ be a sequence generated by Mean-Sigma SA, Min-Max SA or CC-Adaptive SA algorithm, where $d_k = -B_k^{-1}G_k$ is a quasi-Newton direction. Assume that B_k satisfies the conditions (3.36) and (3.37). Then, for an arbitrary initial approximation x_0,*

$$\lim_{k\to\infty} ||g_k|| = 0 \text{ a.s.}$$

Chapter 4

Numerical Implementation

> In theory, theory and practice are the same. In practice, they are not.
>
> Yogi Berra

The sensitivity analysis of mean-sigma (3.3) and min-max (3.24) schemes with respect to the parameter θ for different levels of noise is conducted. Mean-Sigma SA and Min-Max SA algorithms are tested using different search directions. These algorithms are compared to classical SA algorithms with deterministic steps (2.7) and Switching SA algorithm with the scheme (2.12). The generalized step size scheme (3.23) is also tested. The CC-Adaptive SA algorithm is compared to Mean-Sigma SA algorithm and classical SA algorithm in the application to the linear regression models.

4.1 Testing Procedure

Numerical implementation is carried out on a collection of 20 test problems selected from [38] and [44]. The test functions and the problem dimensions

n are given in Table 4.1, while the detail description of all problems is listed in Appendix. All test problems have the form of nonlinear least squares

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

No	Problem	n
1	The Gaussian function	3
2	The Box 3-dimensional function	3
3	The variably dimensioned function	4
4	The Watson function	4
5	The Penalty Function 1	10
6	The Penalty Function 2	4
7	The Trigonometric Function	10
8	The Beale Function	2
9	The Chebyquad Function	10
10	The Gregory and Karney Tridiagonal Matrix Function	4
11	The Hilbert Matrix Function	4
12	The De Jong Function 1	3
13	The Branin RCOS Function	2
14	The Colville Polynomial	4
15	The Powell 3D Function	3
16	The Himmelblau function	2
17	The Fletcher-Powell helical valley function	3
18	The Biggs EXP6 function	6
19	Strictly Convex 1	10
20	Strictly Convex 2	10

Table 4.1: Test problems
Test problemi

The problems are transformed into noisy ones by adding the normal

distributed noise

$$\xi \sim \mathcal{N}(0, s^2) \quad \text{and} \quad \varepsilon \sim \mathcal{N}(0, s^2 E_{n\times n}),$$

to the function and gradient, where s denotes the noise level and $E_{n\times n}$ is the identity matrix. We have tested two different noise levels, $s = 0.4, 1$. The objective function and the gradient value at the current iterate x_k are calculated using sample average approximation with the sample size 3. For each test problem and each algorithm, $N = 50$ independent runs starting from the same initial point are conducted. The final iterate x_{end}, the final function value F_{end} and the final gradient value G_{end} are used as exit parameters. The algorithms stop if the gradient value is small enough, $||G_k|| \leq c$, where $c = \min\{\sqrt{n}s, 1\}$ or the maximal number of $200n$ function evaluations is reached, with each gradient evaluation counted as n function evaluations. The runs are classified into three categories:

1. convergent runs
2. partially convergent runs
3. divergent runs.

The run is convergent if the method stops due to the gradient tolerance, $||G_{end}|| \leq c$. The number of convergent runs is denoted by $Nconv$. If $||G_{end}|| > 200\sqrt{n}$, the run is divergent. The number of divergent runs is denoted by $Ndiv$. Finally, the run that stopped due to exhausting the maximal number of allowed function evaluations is partially convergent. Their number is denoted by $Npar$. In these cases the maximal number of function evaluation is not large enough to achieve convergence with the given gradient tolerance, but the function values are nevertheless decreased so the algorithm makes some progress.

Algorithms are tested using both, the gradient and descent direction. The BFGS direction $d_k = -B_k^{-1}G_k$, is used as the descent direction. The update formula is

$$B_{k+1} = B_k - \frac{B_k\delta_k\delta_k^T B_k}{\delta_k^T B_k \delta_k} + \frac{\Delta_k\Delta_k^T}{\Delta_k\delta_k}, \tag{4.1}$$

where

$$\delta_k = x_{k+1} - x_k \quad \text{and} \quad \Delta_k = G(x_{k+1}, \varepsilon_k) - G(x_k, \varepsilon_k).$$

The gradient difference Δ_k was calculated using the same sample set, at x_k and x_{k+1}, according to the theoretical analysis developed in the previous chapter.

The specification of the parameters for calculations of the step sizes is the following. The values of parameters a, A and α are given in Table 4.2. For the parameter σ in the step size scheme (3.3), we use the noise level s which, as we explained earlier, is closely related to the variance of the sampling distribution of the estimator F_k of the optimal value f^*. Additional results where these values differ are available in [59]. The most suitable value for parameter m is derived empirically. We have used $m = 10$. Performance of the algorithms for different values of m is available in [34, 59].

Consecutive zero steps that can occur during the implementation of both proposed schemes (3.3) and (3.24) may lead to no progress of the algorithm. As an additional implementation issue, we limit the number of consecutive zero steps. The following correction is applied. If the number of consecutive zero steps is greater than some predetermined number m_{corr}, in the next iterate the step size $a_k = \frac{a}{(t_k+1+A)^\alpha}$ is used. The $m_{corr} = m + 1$ is used as the correction value.

4.2 Sensitivity Analysis

In this subsection, we analyze Mean-Sigma SA and Min-Max SA algorithms with respect to the parameter θ for different levels of noise. A Mean-Squared Error (MSE) of the objective function estimator is used as a sensitivity measure. MSE is given by

$$MSE(f) = \sum_{i:||G^{(i)}||\leq c} (y^{(i)} - f^*)^2 / N_{conv},$$

where$y^{(i)}$ is last approximate of the optimal function value, $i = 1, 2, \ldots, 50$ and f^* is the optimal function value.

Problem	a	A	α
1	1	1	0.75
2	1	100	0.501
3	0.1	1	0.75
4	0.1	1	0.75
5	0.1	1	0.75
6	0.1	100	0.501
7	1	100	0.501
8	1	100	0.501
9	0.1	100	0.75
10	0.5	1	0.501
11	0.5	1	0.501
12	0.1	100	0.75
13	0.5	1	0.501
14	1	100	0.501
15	0.1	100	0.75
16	0.5	1	0.501
17	1	0	0.602
18	1	0	0.602
19	0.5	100	0.501
20	0.1	100	0.75

Table 4.2: Initialization of the parameters a, A and α
Vrednosti parametara a, A i α

The following abbreviations are used:

- MSGD - Mean-Sigma SA algorithm with $d_k = -G_k$
- MSDD - Mean-Sigma SA algorithm with $d_k = -B_k^{-1}G_k$
- MMGD - Min-Max SA algorithm with $d_k = -G_k$
- MMDD - Min-Max SA algorithm with $d_k = -B_k^{-1}G_k$.

The comparative results of MSGD and MSDD are listed in Table 4.3 and Table 4.4. The results of MMGD and MMDD are presented in Table 4.5 and Table 4.6. We report results for two values of the parameter $\theta = 0.75, 0.999$.

prb	σ	MSGD		MSDD	
		$\theta = 0.75$	$\theta = 0.999$	$\theta = 0.75$	$\theta = 0.999$
1	0.4	5.00E-05	3.92E-04	7.50E-05	2.88E-04
	1	3.04E-03	3.92E-04	9.88E-04	2.88E-04
2	0.4	9.80E-05	6.72E-02	6.88E-04	8.33E-06
	1	1.10E-02	5.26E-03	1.68E+01	1.39E-03
3	0.4	fail	fail	9.68E-03	2.00E-06
	1	fail	fail	3.56E-02	1.16E-02
4	0.4	6.05E-03	fail	1.12E+01	1.06E+01
	1	1.36E-03	1.88E-01	fail	3.15E+00
5	0.4	fail	fail	fail	fail
	1	fail	fail	fail	fail
6	0.4	9.68E-04	3.92E-04	5.20E-02	3.09E-02
	1	1.15E-03	1.25E-03	1.66E+00	1.76E-01
7	0.4	2.00E-04	1.28E-04	2.00E-04	1.28E-04
	1	fail	fail	fail	fail
8	0.4	fail	fail	1.77E-01	1.41E-01
	1	fail	fail	9.41E-01	7.71E-01
9	0.4	8.63E-07	5.11E-05	6.43E-04	4.21E-07
	1	fail	fail	fail	fail
10	0.4	1.53E-01	8.57E-02	1.58E+00	1.87E+00
	1	1.77E-02	1.24E-01	2.66E-01	3.65E-01

Table 4.3: Mean-Sigma: MSE(f) for Problems 1-10
Mean-Sigma: MSE(f) za probleme 1-10

According to the results, performances of the algorithms are sensitive with respect to the parameter θ, regardless of the chosen scheme, direction and noise. Choosing a larger θ decreases MSE in almost all cases for smaller level of noise $s = 0.4$, regardless of the chosen direction. When $s = 1$, taking

prb	σ	MSGD		MSDD	
		$\theta = 0.75$	$\theta = 0.999$	$\theta = 0.75$	$\theta = 0.999$
11	0.4	7.22E-04	9.80E-05	1.38E-01	9.50E-03
	1	3.04E-03	5.00E-03	1.46E-01	fail
12	0.4	2.05E-03	7.20E-05	8.02E+00	1.89E-01
	1	4.42E-03	3.87E-03	2.46E+03	5.87E+01
13	0.4	8.93E-08	1.53E-01	3.58E-01	6.52E-01
	1	8.17E-04	9.21E-03	6.26E+00	1.64E+00
14	0.4	fail	fail	fail	fail
	1	fail	fail	fail	fail
15	0.4	5.83E-03	2.05E-03	1.77E-02	7.74E-03
	1	6.50E-03	fail	2.03E-02	5.00E-03
16	0.4	fail	fail	2.76E-01	2.61E-01
	1	fail	2.28E-01	4.64E+00	2.50E-01
17	0.4	fail	fail	fail	6.78E-03
	1	fail	fail	fail	1.42E+00
18	0.4	2.65E-03	2.37E-03	1.97E-03	7.68E-03
	1	3.44E-03	5.27E-02	4.47E-01	2.75E-01
19	0.4	1.46E-03	2.00E-04	1.88E-02	5.45E-04
	1	8.33E-04	fail	1.19E+00	fail
20	0.4	1.46E-03	6.48E-04	2.74E-02	1.46E-03
	1	fail	3.72E-01	3.05E+00	2.92E-01

Table 4.4: Mean-Sigma: MSE(f) for Problems 11-20
Mean-Sigma: MSE(f) za probleme 11-20

the larger θ does not produce always such clear pattern in reduction of MSE as for the smaller noise. Therefore, when there exists a strong influence of the noise, it may be useful to take a smaller θ. The smaller θ will produce larger steps at the beginning of the process.

Regardless of the chosen direction and the level of noise, taking a larger θ is superior in the number of convergent runs. Moreover, regardless of the noise level and chosen θ, it may be concluded that algorithms with BFGS

prb	σ	MMGD		MMDD	
		$\theta = 0.75$	$\theta = 0.999$	$\theta = 0.75$	$\theta = 0.999$
1	0.4	7.20E-05	5.00E-05	2.42E-04	2.47E-04
	1	1.46E-03	2.59E-03	8.16E-04	2.47E-04
2	0.4	3.38E-04	1.80E-05	1.79E+01	8.33E-06
	1	9.80E-05	1.28E-04	1.30E-01	7.37E-04
3	0.4	fail	fail	fail	5.94E-04
	1	fail	fail	fail	2.88E-02
4	0.4	1.69E-03	7.75E-04	fail	fail
	1	2.46E-03	3.31E-03	fail	fail
5	0.4	fail	fail	fail	fail
	1	fail	fail	fail	fail
6	0.4	2.90E-04	1.80E-05	6.88E-03	3.11E-03
	1	8.03E-04	2.88E-04	2.70E-02	2.23E-01
7	0.4	3.20E-05	1.28E-04	2.32E-04	1.28E-04
	1	fail	fail	fail	fail
8	0.4	2.63E-04	fail	7.78E+00	9.62E-01
	1	2.59E-03	fail	2.94E+01	7.22E-01
9	0.4	3.78E-05	3.10E-05	3.26E-03	5.83E-04
	1	fail	fail	6.97E-03	6.97E-03
10	0.4	2.45E-01	1.29E-01	6.57E-01	5.32E-01
	1	2.44E-01	1.28E-01	3.57E-01	3.75E+00

Table 4.5: Min-Max: MSE(f) for Problems 1-10
Min-Max: MSE(f) za probleme 1-10

direction (MSDD and MMDD) have higher number of convergent runs than gradient algorithms (MSGD and MMGD).

In most of the cases, MMGD yields smaller MSE than MSGD. This finding holds regardless of the chosen m, [34, 59]. Therefore, Extreme Value Statistics may be more suitable criteria for the step size selection when the direction is negative gradient. On the other hand, it seems that MSDD behaves superior than MMDD, especially for larger level of noise.

prb	σ	MMGD		MMDD	
		$\theta = 0.75$	$\theta = 0.999$	$\theta = 0.75$	$\theta = 0.999$
11	0.4	5.71E-03	5.78E-04	2.24E-01	1.05E-01
	1	1.04E-02	1.06E-03	3.69E+00	8.88E-02
12	0.4	fail	4.50E-04	3.87E+01	9.88E-01
	1	fail	1.57E-03	2.01E+02	1.52E+01
13	0.4	9.78E-06	1.24E-06	1.93E-03	3.45E-01
	1	4.59E-05	6.10E-03	1.88E+01	9.34E-01
14	0.4	fail	fail	fail	fail
	1	fail	fail	fail	fail
15	0.4	3.06E-04	2.00E-03	1.48E-02	1.61E-02
	1	1.25E-03	1.11E-02	2.30E-02	1.67E-03
16	0.4	6.91E-03	fail	fail	1.44E-02
	1	5.53E-03	fail	fail	fail
17	0.4	fail	fail	1.98E-01	4.56E+00
	1	fail	fail	8.74E+01	8.44E-01
18	0.4	fail	2.65E-03	2.08E+01	3.24E-03
	1	2.59E-03	5.02E-03	5.69E-02	1.16E-01
19	0.4	3.38E-04	5.12E-04	3.44E-03	2.00E-04
	1	3.20E-03	7.84E-02	fail	1.27E+01
20	0.4	3.43E-01	8.82E-04	5.15E-02	2.37E-03
	1	fail	2.12E-01	2.62E+01	fail

Table 4.6: Min-Max: MSE(f) for Problems 11-20
Min-Max: MSE(f) za probleme 11-20

4.3 Comparison of the Algorithms

The performance of Mean-Sigma SA and Min-Max SA algorithms is compared to SA algorithms with the classical steps (2.7) and the switching step size scheme (2.12). The following abbreviations are used

- SAGD - Gradient SA algorithm (2.3) with the step sizes (2.7)

- SADD - Descent direction SA algorithm (2.9) with $d_k = -B_k^{-1}G_k$ and the step sizes (2.7).

- XDGD - Switching SA algorithm

The number of function evaluations in successful and partially successful runs is chosen as the performance measures

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

where $Ncon_{ij}$ is the number of successful runs for ith Algorithm to solve problem j, $Npar_{ij}$ is the number of partially successful runs for ith Algorithm to solve problem j, $fcalc_{ij}^r$ is the number of function evaluations needed for ith Algorithm to solve problem j in rth run and n_j is the dimension of problem j, $i = 1, \cdots, 7,\ j = 1, \cdots, 20,\ r = 1, \cdots, 50$. We used $\theta = 0.999$.

Overviews of the successful, partially successful and unsuccessful runs for the noise levels $s = 0.4$ and $s = 1$ are given in Figure 4.1 and Figure 4.2, respectively. The results demonstrate that Mean-Sigma SA and Min-Max SA algorithms have the smallest number of divergent runs and the highest number of convergent runs regardless of the chosen direction and the noise level. Algorithms MSDD and MMDD are significantly better than the corresponding SADD algorithm for both noise levels. Mean-Sigma SA and Min-Max SA algorithms are competitive with the Switching SA algorithm which confirms that taking noisy function values as a criterion for adjusting steps can improve the optimization process.

Figure 4.3 and Figure 4.4 show performance profiles for $s = 0.4$ and $s = 1$, respectively. For both levels of noise, Mean-Sigma SA and Min-Max SA algorithms outperform all other tested algorithms regardless of the chosen direction and noise level.

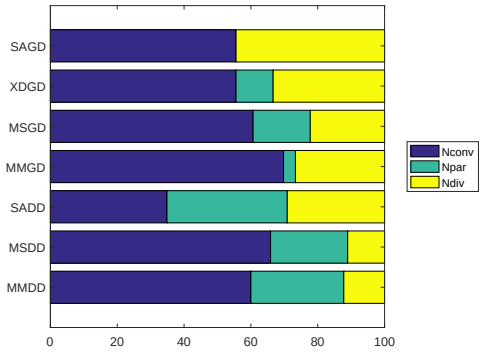

Figure 4.1: Percentages of successful, partially successful and divergent runs, $s = 0.4$
Procenti uspešnih, delimično uspešnih i divergentnih postupaka, $s = 0.4$

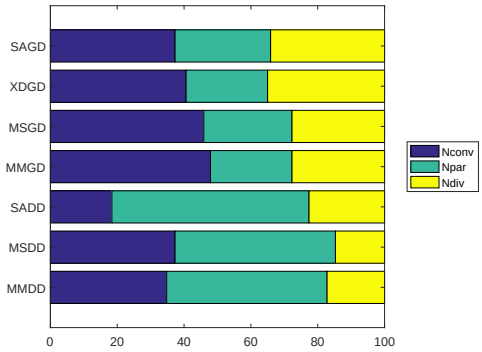

Figure 4.2: Percentages of successful, partially successful and divergent runs, $s = 1$
Procenti uspešnih, delimično uspešnih i divergentnih postupaka, $s = 1$

4.4 Application to Regression Models

In this section we consider a linear regression model given in the matrix form

$$y = X\beta + \epsilon, \tag{4.2}$$

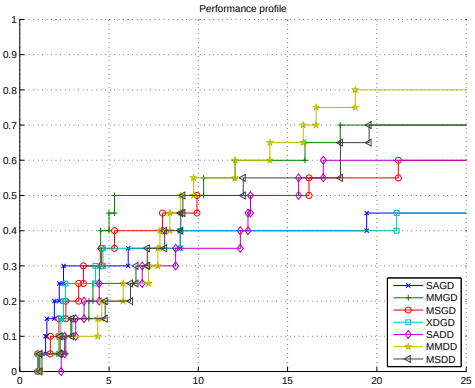

Figure 4.3: Performance profile, $s = 0.4$
Profil učinka, $s = 0.4$

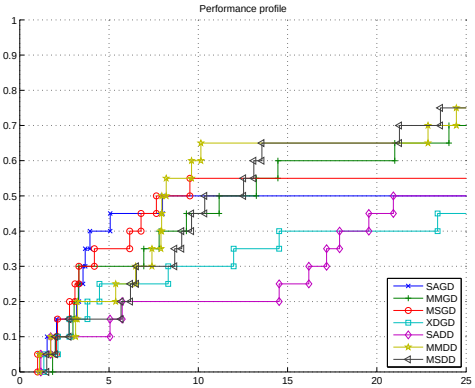

Figure 4.4: Performance profile, $s = 1$
Profil učinka, $s = 1$

where

- $y = (y_1, y_2, ..., y_n)^T$ is the vector of dependent variables,

- $X = [x_{ij}]_{n\times p}$ is the matrix of independent variables,
- $\beta = (\beta_1, \beta_2, ..., \beta_p)^T$ is the vector of associated regression coefficients,
- $\epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_n)^T$ is the vector of i.i.d. random errors with $E(\epsilon_i) = 0$ and $D(\epsilon_i) = s^2$.

The most commonly used method for estimating the unknown parameters $\beta_1, \beta_2, ..., \beta_p$ is the Ordinary Least-Square (OLS) method, where the residual square error

$$RSS = \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2$$

is minimized. In other words, the parameter estimates are obtained by solving the unconstrained OLS optimization problem

$$\hat{\beta}^{ols} = \underset{\beta\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2. \tag{4.3}$$

The estimators obtained by the OLS method are unbiased and consistent. However, they often have low bias, but large variance. To overcome this deficiency of the OLS method and improve the estimates, introduction of an additional information via the process of regularization is suggested. Tibshirani introduced the Least Absolute Shrinkage and Selection Operator (LASSO) regularization method, [61]. LASSO regularization is a process of adding constraints in the form of L_1-norm of the parameter vector β. The associated unconstrained optimization problem is given by

$$\hat{\beta}^{lasso} = \underset{\beta\in\mathbb{R}^1}{\arg\min} \Big\{ \sum_{i=1}^{n} \big(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\big)^2 + \mu \sum_{j=1}^{p} |\beta_j| \Big\}. \tag{4.4}$$

Due to the nature of the L_1 penalty, the LASSO method automatically does the selection of independent variables. In practice, the value of μ, as the level of regularization, is predefined, or it is chosen from some candidate set

using selection methods as Cross-Validation, Bayesian information criterion or Akaike information criterion.

SA algorithm (2.3) with the steps (2.7), Mean-Sigma SA algorithm and CC-Adaptive SA algorithm are applied for solving the problem (4.4) in order to find the estimates of the parameter vector β in the regression model (4.2). The direction $d_k = -G_k$ is taken. The abbreviation CCGD is used for CC-Adaptive SA algorithm which uses negative gradient direction, while the abbreviations for the other two algorithms are the same as in the previous sections (SAGD and MSGD).

The application is illustrated on the following example.

Example 1 *[61] In this example we are looking for the estimate of the parameter β in $Y = X\beta + \epsilon$, where the true value of β is*

$$\beta = (3, 1.5, 0, 0, 2, 0, 0, 0)^T.$$

We simulated $N = 50$ data sets of $n = 100$ observations, where the random errors ϵ_i, $i = 1, 2, ..., n$ are i.i.d. with normal distributions

$$\epsilon_i \sim \mathcal{N}(0, s^2), i = 1, 2, ..., n,$$

with $s = 3$. The column vectors $X_i, i = 1, 2, ..., p$ of the matrix X of independent variables are chosen to have n-dimensional normal distributions

$$X_j \sim \mathcal{N}(0, C), j = 1, 2, ..., p,$$

where the covariance matrix $C = [c_{ij}]$ is such that $c_{ij} = \rho^{|i-j|}, i, j = 1, 2, ..., p$, with $\rho = 0.5$, [62]. The K-fold cross-validation with $K = 5$ is used to estimate the regularization level μ in (4.4), [14]. As a candidate set for the regularization parameter μ, the set $\{0, 0.01, 0.1, 1, 10, 100\}$ is considered.

The values of parameters used in the step sizes are $a = 0.001$, $A = 0, 10, 100$, $\alpha = 0.602$, $m = 10$, $\theta = 0.99$ and $b = 1$. Results that we present are obtained with $m = 10$ and $\theta = 0.99$. The coefficients $\lambda_{k,j}$ are chosen as follows

$$\lambda_{k,1} = \begin{cases} 1, & F_k > \sum_{j=1}^{m(k)} \tilde{\lambda}_{k,j} F_{k-j} \\ \tilde{\lambda}_{k,1}, & \text{otherwise} \end{cases},$$

and

$$\lambda_{k,j} = \begin{cases} 0, & F_k > \sum_{j=1}^{m(k)} \tilde{\lambda}_{k,j} F_{k-j} \\ \tilde{\lambda}_{k,j}, & \text{otherwise} \end{cases}, \ j = 2, ..., m(k),$$

where

$$\tilde{\lambda}_{k,j} = \lambda, \ j \neq \tilde{j} \text{ and } \tilde{\lambda}_{k,\tilde{j}} = 1 - (m(k) - 1)\lambda,$$

$\lambda = 0.01$ and $\tilde{j}$ is such that $F_{k-\tilde{j}} = \max_{1 \leq j \leq m(k)} F_{k-j}$. The gradient of the objective function in (4.4) is approximated by the finite differences with step $h = 10^{-5}$.

Comparison of the algorithms is based on the evaluation of Mean Square Error (MSE) and Median Square Error (MedianSE) defined by

$$MSE = \frac{1}{N} \sum_{k=1}^{N} (\hat{\beta}^k - \beta)^T C (\hat{\beta}^k - \beta)$$

and

$$MedianSE = Median\{(\hat{\beta}^k - \beta)^T C (\hat{\beta}^k - \beta), k = 1, 2, ..., N\},$$

respectively, where $\hat{\beta}^k$ is the kth estimate of the parameter β.

MSE and MedianSE for different values of the parameter A and different initial iterates β_0 in the optimization processes are given in Table 4.7, Table 4.8 and Table 4.9.

MSGD and CCGD have been equally successful with little difference in MSEs or MedianSEs, almost always in favour of (3.3), except for $A = 10$ and initial point $\beta_0 = (0, 0, 0, 0, 0, 0, 0, 0)^T$, when result in bigger MSE and MedianSE.

MSGD and CCGD have better performance when the optimization process starts far from the solution. SAGD is very sensitive with respect to the choice of the parameter A, which is not the case for the algorithm with the new step size schemes.

	MSE	MedianSE
$\beta_0 = (0,0,0,0,0,0,0,0)^T$		
SAGD	0.67713284	0.65167476
MSGD	0.73254119	0.65512802
CCGD	0.73114389	0.64433737
$\beta_0 = (10,10,10,10,10,10,10,10)^T$		
SAGD	0.81613584	0.73665411
MSGD	0.71485402	0.66087552
CCGD	0.72263938	0.66571446

Table 4.7: MSE and MedianSE, $A = 0$
MSE i MedijanaSE, $A = 0$

	MSE	MedianSE
$\beta_0 = (0,0,0,0,0,0,0,0)^T$		
SAGD	0.72407352	0.70555047
MSGD	0.74199922	0.67134859
CCGD	0.73055080	0.65525906
$\beta_0 = (10,10,10,10,10,10,10,10)^T$		
SAGD	1.05999336	0.99529997
MSGD	0.71411317	0.66605187
CCGD	0.72268785	0.66862433

Table 4.8: MSE and MedianSE, $A = 10$
MSE i MedijanaSE, $A = 10$

	MSE	MedianSE
$\beta_0 = (0,0,0,0,0,0,0,0)^T$		
SAGD	1.13762219	1.17352551
MSGD	0.71541790	0.64763942
CCGD	0.72503869	0.65551539
$\beta_0 = (10,10,10,10,10,10,10,10)^T$		
SAGD	2.21081217	2.15215925
MSGD	0.70413645	0.63140689
CCGD	0.70862254	0.63606241

Table 4.9: MSE and MedianSE, $A = 100$
MSE i MedijanaSE, $A = 100$

Future Work

> Science is always wrong, it never solves a problem without creating ten more.
>
> George Bernard Shaw

In the thesis we have proposed two adaptive step size schemes for SA algorithms. According to the schemes, the step sizes selection is based on the previously observed noisy function values. Larger steps are used if values of the objective function are decreased sufficiently. Under a non restrictive assumption of i.i.d. continuous random noise with a positive pdf, the generated step size sequences have the desired SA step size property. The almost sure convergence of SA algorithms with the proposed adaptive step size schemes is established. Numerical results verify better performance of the SA algorithms with new adaptive step sizes compared to the existing algorithms with adaptive step sizes. In the future, it will be challenging to analyze convergence in a more general case of state dependent noise and with no restrictions to its pdf, since we have obtained good numerical results in these cases too. It will also be interesting to introduce variability in constants m and σ.

Appendix

The list of test problems is the following.
Problem 1. [38] Gaussian function; $n = 3$, $r = 15$

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

where $y_1 = y_{15} = 0.0009$, $y_2 = y_{14} = 0.004$, $y_3 = y_{13} = 0.0175$, $y_4 = y_{12} = 0.0540$, $y_5 = y_{11} = 0.1295$, $y_6 = y_{10} = 0.2420$, $y_7 = y_9 = 0.3521$ and $y_8 = 0.3989$, $x_0 = (0.4, 1, 0)$, $x^* -$ unknown, $f^* = 1.12793 \cdot 10^{-8}$;

Problem 2. [38] Box three-dimensional function; $n = 3$, $r = 10$

$$f_i(x) = \exp[-t_i x_1] - \exp[-t_i x_2] - x_3(\exp[-t_i] - \exp[-10t_i]),$$

where $t_i = \frac{i}{10}$, $i = 1, ..., m$, $x_0 = (0, 10, 5)$, $x^* = (1, 10, 1)$ or $(10, 1, -1)$ or $x_1 = x_2$ and , $x_3 = 0$, $f^* = 0$;

Problem 3. [38] The variably dimensioned function; $n = 4$, $r = 8$

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

where $2 \leq i \leq n$, and

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

$$f_{2n}(x) = (\sum_{j=1}^{n}(n - j + 1)x_j^2) - 1, y_i = \exp(\frac{i}{10}) + \exp(\frac{i-1}{10}),$$

$x_0 = (0.75, 0.5, 0.25, 0)$, $x^* -$ unknown, $f^* = 9.37629 \cdot 10^{-6}$;

Problem 4. [38] The Watson function; $n = 6$, $r = 13$

$$f_i(x) = x_3 e^{-t_i x_1} - x_4 e^{-t_i x_2} + x_6 e^{-t_i x_5} - y_i,$$

$$t_i = 0.1i,\ y_i = e^{-t_i} - 5e^{-10t_i} + 3e^{-4t_i},$$

$x_0 = (10, 10, 1, 1, 10, 1)$, $x^* = (1, 10, 1, 5, 4, 3)$, $f^* = 0$;

Problem 5. [38] Penalty function I; $n = 10$, $r = 11$

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

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

Problem 6. [38] Penalty function II; $n = 4$, $r = 8$

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

$2 \le i \le n$, and

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

and

$$f_{2n}(x) = (\sum_{j=1}^{n}(n - j + 1)x_j^2) - 1,\ y_i = \exp(\frac{i}{10}) + \exp(\frac{i-1}{10}),$$

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

Problem 7. [38] Trigonometric function; $n = 10$, $r = 10$, $i = 1, \ldots, n$

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

$x_0 = (1, 0, ..., 1, 0), x^* -$ unknown, $f^* = 0$;

Problem 8. [38] Beale function; $n = 2$, $r = 3$, $i = 1, \ldots, n$

$$f_i(x) = y_i - x_1(1 - x_2^i),$$

$y_1 = 1.5$, $y_2 = 2.25$, $y_3 = 2.625$ $x_0 = (1, 1)$, $x^* = (3, 0.5)$, $f^* = 0$;

Problem 9. [38] Chebyquad function; $n = 10$, $r = 10$

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

T_i is the i-th Chebyshev polynomial shifted to the interval $[0, 1]$,

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

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

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

Problem 10. [38] The Gregory and Karney Tridiagonal Matrix Function; $n = 4$

$$f(x) = x'Ax - 2x_1,$$

where A is the $(-1, 2, -1)$ tridiagonal matrix, except that $A(1, 1) = 1$, $x_0 = (0, 0, 0, 0)$, $x^* = (n, n - 1, ..., 2, 1)$, $f^* = -n$;

Problem 11. [38] The Hilbert Matrix Function; $n = 4$

$$f(x) = x^T Ax,$$

where A is $n \times n$ Hilbert matrix, i.e., $a_{ij} = \frac{1}{i+j-1}$ for $i, j = 1, \cdots, n$, $x_0 = (1, 1, 1, 1)$, $x^* = (0, 0, 0, 0)$, $f^* = 0$;

Problem 12. [38] The De Jong Function 1; $n = 3$

$$f(x) = \sum_{i=1}^{n} x_i^2,$$

$x_0 = (-5.12, 0, 5.12)$, $x^* = (0, 0, 0)$, $f^* = 0$;

Problem 13. [38] The Branin RCOS Function; $n = 2$

$$f(x) = (x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x + 10,$$

$x_0 = (-1, 1)$;

Problem 14. [38] The Colville Polynomial; $n = 4$;

$$\begin{aligned} f(x) &= 100(x_2 - x_1)^2 + (1 - x_1)^2 + 90(x_4 - x_3)^2 \\ &+ (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1), \end{aligned}$$

$x_0 = (1/2, 1, -1/2, -1)$, $x^* = (-\pi, 12.275)$, $x^* = (\pi, 2.275)$, $x^* = (9.42478, 2.475)$, $f^* = 0.397887$;

Problem 15. [38] The Powell 3D Function; $n = 3$;

$$f(x) = 3 - (\frac{1}{1 + (x_1 - x_2)^2}) - \sin(\pi x_2 x_3/2) - \exp[-(\frac{x_1 + x_2}{x_2}) - 2]^2,$$

$(0, 1, 2)$, $x^* = (0, 0, 0)$;

Problem 16. [38]The Himmelblau function; $n = 2$;

$$f(x) = (x_1^2 + x - 11)^2 + (x_1 + x_2^2 - 7)^2,$$

$x_0 = (-1.3, 2.7),\ x^* = (3, 2),\ x^* = (3, 2),\ x^* = (-.2805118, 3.131312),$
$x^* = (-3.779310, -3.283186),\ x^* = (3.584428, -1.848126),\ f^* = 0;$

Problem 17. [44] The Fletcher-Powell helical valley function; $n = 3$

$$f(x) = 100(x_3 - 10\theta)^2 + 100(\sqrt{x_1^2 + x_2^2} - 1)^2 + x_3^2,$$

$x_0 = (-1, 0, 0), \quad x^* = (1, 0, 0), \quad f^* = 10;$

Problem 18. [44] The Biggs EXP6 function; $n = 6, m = 13,\ i = 1, \ldots, n$

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

$x_0 = (10, 10, 1, 1, 10, 1), \quad x^* = (1, 10, 1, 5, 4, 3), \quad f^* = 0;$

Problem 19. [44] The Strictly Convex 1; $n = 10$

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

$x_0 = (1/n, ..., i/n, ..., 1), \quad x^* = (0, ..., 0), \quad f^* = 10;$

Problem 20. [44] The Strictly Convex 2; $n = 10$

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

$x_0 = (1, ..., 1), \quad x^* = (0, ..., 0), \quad f^* = 5.5;$

Bibliography

[1] S. Andradottir, *A Review of Simulation Optimization Techniques,* Proceedings of the 30th conference on Winter simulation, (1998), 151-158.

[2] A. Antoniou, W. S. Lu, *Practical Optimization: Algorithms and Engineering Applications,* Springer, 2007.

[3] A. Benveniste, M. Metivier, P. Priouret, *Adaptive Algorithms and Stochastic Approximations,* Springer, New York, 1990.

[4] D. P. Bertsekas, J. N. Tsitsiklis, *Gradient Convergence in Gradient Methods with Errors*, SIAM. J. Optim. 10(3) (2000), 627-642.

[5] J. R. Blum, *Multidimensional Stochastic Approximation Methods,* Ann. Math. Stat. 25 (1954), 737-744.

[6] 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), 977-995.

[7] R. H. Byrd, G. M. Chin, J. Nocedal, Y. Wu, *Sample Size Selection in Optimization Methods for Machine Learning*, Math. Program. 134(1) (2012), 127-155.

[8] R. H. Byrd, S. L. Hansen, J. Nocedal, Y. Singer, *A Stochastic Quasi-Newton Method for Large Scale Optimization,* SIAM J. Optim. 26(2) (2016), 1008-1031.

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

[10] A. R. Conn, K. Scheinberg, L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Book Series on Optimization, SIAM, Philadelphia, 2009.

[11] B. Delyon, A. Juditsky, *Accelerated Stochastic Approximation,* SIAM J. Optim. 3(4) (1993), 868-881.

[12] J. Dennis, R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* SIAM, 1996.

[13] R. Durrett, *Probability: Theory and Examples,* Second edition, Duxbury Press, Belmont, CA, 1995.

[14] B. Efron, *The Estimation of Prediction Error: Covariance Penalties and Crossvalidation,* J. Amer. Statis. Assoc. 99 (2004), 619-642.

[15] M. A. Epelman , *Overview of Algorithms for Unconstrained Optimization,* Lecture notes, University of Michigan, 2012.

[16] V. Fabian, *On Asymptotic Normality in Stochastic Optimization,* Ann. Math. Stat. 39 (1968), 1327-1332.

[17] V. Fabian, *Stochastic Approximation,* J. S. Rustigi ed, Academic Press, New York, 439-470.

[18] H. T. Fang, H. F. Chen, *Almost Surely Convergent Global Optimization Algorithm Using Noise-Corrupted Observations,* J. Optim. Theor. Appl. 104(2) (2000), 343-376.

[19] A. Gut, *An Intermediate Course in Probability,* Springer, 2009.

[20] I. Griva, S. G. Nash, A. Sofer, *Linear and Nonlinear Optimization,* SIAM, 2009.

[21] C. Kao, S. P. Chen, *A Stochastic Quasi-Newton Method for Simulation Response Optimization,* Eur. J. Oper. Res. 173 (2006), 30-46.

[22] C. Kao, W. T. Song, S. Chen, *A Modified Quasi-Newton Method for Optimization in Simulation,* Int. Trans. O.R. 4(3) (1997), 223-233.

[23] J. Kiefer, J. Wolfowitz, *Stochastic Estimation of the Modulus of a Regression Function*, Ann. Math. Stat. 23 (1952), 462-466.

[24] H. Kesten, *Accelerated Stochastic Approximation,* Ann. Math. Stat. 29 (1958), 41-59.

[25] H. J. Kushner, D. S. Clark, *Stochastic Approximation for Constrained and Unconstrained Systems*, Springer, 1978.

[26] H. J. Kushner, G. G. Yin, *Stochastic Approximation Algorithm and Applications*, Springer, New York, 1997.

[27] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, 1999.

[28] K. Knight, *Mathematical Statistics*, Chapman & Hall/CRC, Boca Raton, Florida, 2000.

[29] N. Krejić, Z. Lužanin, I. Stojkovska, *A Gradient Method for Unconstrained Optimization in Noisy Environment,* Appl. Numer. Math. 70 (2013), 1-21.

[30] N. Krejić, Z. Lužanin, I. Stojkovska, Z. Ovcin, *Descent Direction Method with Line Search for Unconstrained Optimization in Noisy Environment,* Optim. Methods Softw. 30(6) (2015), 1164-1184.

[31] N. Krejić, Z. Lužanin, F. Nikolovski, I. Stojkovska, *A Nonmonotone Line Search Method for Noisy Minimization,* Optim. Lett. 9(7) (2015), 1371-1391.

[32] N. Krejić, N. Krklec Jerinkić, *Stochastic Gradient Methods for Unconstrained Optimization,* Pesq. Oper. 34 (3) (2014), 373-393.

[33] M. Kresoja, M. Dimovski, I. Stojkovska, Z. Lužanin, *Stochastic Approximation with Adaptive Step Sizes for Optimization in Noisy Environment and its Application in Regression Models,* Matematički bilten 40(4) (2016), 62-79.

[34] M. Kresoja, Z. Lužanin, I. Stojkovska, *Stochastic Approximation with Adaptive Step Sizes,* Numer. Algor. (2017) DOI: 10.1007/s11075-017-0290-4

[35] N. Krklec Jerinkić, *Line Search Methods with Variable Sample Size*, PhD Thesis, University of Novi Sad, 2014.

[36] B. Li, Y. Ong, M. Le, C. Goh, *Memetic Gradient Search*, Proc. of IEEE CEC (2008), 2894-2901.

[37] D. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, Springer, 2008.

[38] J. J. Moré, B. S. Garbow, K. E. Hillstrom, *Testing Unconstrained Optimization Software,* ACM Trans. Math. Software 7(1) (1981), 17-41.

[39] V. B. Nevzorov, *Records: Mathematical Theory,* American Mathematical Societ, 2001.

[40] J. Nocedal, J. Wright, *Numerical Optimization,* Springer, 1999.

[41] A. Nobel, T. Adams, *Estimating a Function from Ergodic Samples with Additive Noise,* IEEE Transactions on Information Theory 47 (2001), 2895- 2902.

[42] F. Yousefian, A. Nedic, U. V. Shanbhag, *On Stochastic Gradient and Subgradient Methods with Adaptive Steplength Sequences,* Automatica J. IFAC 48(1) (2012), 56-67.

[43] S. M. Ross, *Introduction to Probability Models,* Elsevier, 2007.

[44] M. Raydan, *The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem,* SIAM J. Optim. 7 (1) (1997), 26-33.

[45] H. Robbins, S. Monro, *A Stochastic Approximation Method,* Ann. Math. Stat. 22 (1951), 400-407.

[46] H. Robbins, D. Siegmund, *A Convergence Theorem for Nonnegative Almost Supermatringales and Some Applications,* Optimizing Methods in Statistics, Academic Press, New York (1971), 233-257.

[47] H. Rupert, *Stochastic Approximation*, Handbook of Sequential Analysis, Marcek Dekker, New York (1991), 503-529.

[48] O. Ryan, G. Dahl, K. Morken, *Nonlinear Optimization,* Lecture notes, University of Oslo, 2013.

[49] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Chapter 6. Stochastic Approximation Methods, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[50] J. Sascks, *Asymptotic Distribution of Stochastic Approximation Procedures,* Ann. Math. Stat. 29 (1958), 889-892.

[51] A. Shapiro, D. Dentcheva, A. Ruszzynsky, *Lectures on Stochastic Programming: Modeling and Theory,* SIAM, 2009.

[52] N. N. Schraudolph, J. Yu, S. Gnter, *A Stochastic Quasi-Newton Method for Online Convex Optimization,* Proceedings of 11th International Conference on Artificial Intelligence and Statistics, San Juan, Puerto Rico (2007), 433-440.

[53] K. Sirlantzis, J. D. Lamb, W. B. Liu, *Novel Algorithms for Noisy Minimization Problems with Applications to Neural Networks Training,* J. Optim. Theor. Appl. 129 (2) (2006), 325-340.

[54] J. Solomon, *Numerical Algorithms,* AK Peters/CRC Press, 2015.

[55] J. C. Spall, *Adaptive Stochastic Approximation by the Simultaneous Perturbation Method,* IEEE Trans. Automat. Contr. 45(10) (2000), 1839-1853.

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

[57] J. C. Spall, *Feedback and Weighting Mechanism for Improving Jacobian Estimates in the Adaptive Simultaneous Perturbation Algorithm*, IEEE Trans. Automat. Contr. 54 (6) (2009), 1216-1229.

[58] I. Stojkovska, *Modifications of Newton's Method for Stochastic Optimization Problems,* PhD Thesis, University of Skopje, 2011.

[59] Z. Lužanin, I. Stojkovska, M. Kresoja, *Descent Direction Stochastic Approximation Algorithm with Adaptive Step Sizes,* Tech. Rep., 2016.

[60] R. K. Sundaram, *First Course in Optimization Theory,* Cambridge University Press, UK, 1996.

[61] R. Tibshirani, *Regression Shrinkage and Selection via the Lasso,* J. R. Stat. Soc. Series B 58 (1996), 267-288.

[62] R. Tibshirani, *The Lasso Method for Variable Selection in the Cox Model,* Stat. Med. 16 (1997), 385-395.

[63] X. Yue, *Improved Simultaneous Perturbation Stochastic Approximation and its Application in Reinforcement Learning*, International Conference on Computer Science and Software Engineering 1 (2008), 329-339.

[64] Y. Wardi, *Stochastic Algorithms with Armijo Stepsizes for Minimization of Functions,* J. Optim. Theor. Appl. 64(2) (1990), 399-417.

[65] Z. Xu, *A Combined Direction Stochastic Approximation Algorithm*, Optim. Lett. 4(1) (2010), 117-129.

[66] Z. Xu, Y. H. Dai, *A Stochastic Approximation Frame Algorithm with Adaptive Directions*, Numer. Math. Theor. Meth. Appl. 1(4) (2008), 460-474.

[67] Z. Xu, X. Xu, *A New Hybrid Stochastic Approximation Algorithm*, Optim. Lett. 7(3) (2013), 593-606.

[68] Z. Xu, Y. H. Dai, *New Stochastic Approximation Algorithms with Adaptive Step Sizes*, Optim. Lett. 6 (2012), 1831-1846.

[69] X. Wang, S. Ma, W. Liu, *Stochastic Quasi-Newton Methods for Nonconvex Stochastic Optimization*, arXiv:1412.1196 [math.OC], 2014.

[70] H. White, *Some Asymptotic Results for Learning in Single Hidden Layer Feedforward Neural Network*, J. Amer. Stat. Assoc. 84 (1989), 1003-1013.

[71] H. Zhang, W. Hager, *A Nonmonotone Line Search Technique and its Application to Unconstrained Optimization,* SIAM J. Optim. 14(4) (2004), 1043-1056.

Biography

I was born on 9th May 1988 in Kikinda. I received a Bachelor degree in Financial Mathematics (2010) and a Master degree in Applied Mathematics (2011) from Department of Mathematics and Informatics, Faculty of Sciences at University of Novi Sad. Since 2012, I have been PhD student at the same Department in the field of Numerical mathematics. By June 2014, I passed all the exams with the GPA 10.00. I am a junior researcher at the projects "Numerical methods - simulation and application" and "Approximation of integral and differential operators and applications", both supported by Serbian Ministry of Education, Science and Technological Development. I am giving tutorials in Numerical Analysis, Econometrics, Mathematical Models in Economy, Financial Mathematics, Distributed Optimization and Applications and Modelling Seminar at Department of Mathematics and Informatics. I also give tutorials and teach LLL courses at University Centre of Applied Statistics. My research interests revolve around numerical optimization, data mining and statistical/mathematical modelling problems arising in industry, medicine and social sciences.

Novi Sad, February 2017 Milena Kresoja

UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCE
KEY WORDS DOCUMENTATION

Accession number:
ANO
Identification number:
INO
Document type: Monograph type
DT
Type of record: Printed text
TR
Contents code: PhD dissertation
CC
Author: Milena Kresoja
AU
Mentor: Prof. Dr. Zorana Lužanin
MN

Title: Modifications of Stochastic Approximation Algorithm Based on Adaptive Step Sizes
TI

Language of text: English
LT
Language of abstract: English/Serbian
LA

Country of publication: Republic of Serbia
CP
Locality of publication: Vojvodina
LP
Publication year: 2017
PY
Publisher: Author's reprint
PU
Publication place: Novi Sad, Faculty of Sciences, Trg Dositeja Obradovića 4
PP
Physical description: 4/111/71/9/0/4/1
(chapters/pages/literature/tables/pictures/graphics/appendices)
PD
Scientific field: Mathematics
SF
Scientific discipline: Numerical mathematics
SD
Subject / Key words: Nonlinear optimization, stochastic optimization, noisy function, stochastic approximation, adaptive step sizes, gradient direction, descent direction.
SKW

UC:
Holding data: Library of the Department of Mathematics and Informatics, Novi Sad
HD
Note:
N
Abstract: The problem under consideration is an unconstrained minimization problem in noisy environment. The common approach for solving the problem is Stochastic Approximation (SA) algorithm. We propose a class of adaptive step size schemes for the SA algorithm. The step size selection in the proposed schemes is based on the objective function

values. At each iterate, interval estimates of the optimal function value are constructed using the fixed number of previously observed function values. If the observed function value in the current iterate is larger than the upper bound of the interval, we reject the current iterate. If the observed function value in the current iterate is smaller than the lower bound of the interval, we suggest a larger step size in the next iterate. Otherwise, if the function value lies in the interval, we propose a small safe step size in the next iterate. In this manner, a faster progress of the algorithm is ensured when it is expected that larger steps will improve the performance of the algorithm. We propose two main schemes which differ in the intervals that we construct at each iterate. In the first scheme, we construct a symmetrical interval that can be viewed as a confidence-like interval for the optimal function value. The bounds of the interval are shifted means of the fixed number of previously observed function values. The generalization of this scheme using a convex combination instead of the mean is also presented. In the second scheme, we use the minimum and the maximum of previous noisy function values as the lower and upper bounds of the interval, respectively. The step size sequences generated by the proposed schemes satisfy the step size convergence conditions for the SA algorithm almost surely. Performance of SA algorithms with the new step size schemes is tested on a set of standard test problems. Numerical results support theoretical expectations and verify efficiency of the algorithms in comparison to other relevant modifications of SA algorithms. Application of the algorithms in LASSO regression models is also considered. The algorithms are applied for estimation of the regression parameters where the objective function contains L_1 penalty.
AB

Accepted by Scientific Board on: 16.7.2015.
ASB
Defended:
DE

Thesis defend board:
President: Nataša Krejić, PhD, Full Professor, Faculty of Sciences, University of Novi Sad

Member: Zorana Lužanin, PhD, Full Professor, Faculty of Sciences, University of Novi Sad, advisor

Member: Sanja Rapajić, PhD, Associate Professor, Faculty of Sciences, University of Novi Sad

Member: Irena Stojkovska, PhD, Associate Professor, Faculty of Natural Sciences and Mathematics, Ss. Cyril and Methodius University in Skopje

Member: Zoran Ovcin, PhD, Assistant Professor, Faculty of Technical Sciences, University of Novi Sad

DB

UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj:
RBR
Identifikacioni broj:
IBR
Tip dokumentacije: Monografska dokumentacija
TD
Tip zapisa: Tekstualni štampani materijal
TZ
Vrsta rada: Doktorska disertacija
VR
Autor: Milena Kresoja
AU
Mentor: Prof. dr Zorana Lužanin
MN

Naslov rada: Modifikacije algoritma stohastičke aproksimacije zasnovane na prilagođenim dužinama koraka
NR

Jezik publikacije: engleski
JP
Jezik izvoda: engleski/srpski
JI

Zemlja publikovanja: Republika Srbija
ZP
Uže geografsko područje: Vojvodina
UGP
Godina: 2017.
GO
Izdavač: Autorski reprint
IZ
Mesto i adresa: Novi Sad, Prirodno-matematički fakultet, Trg Dositeja Obradovića 4
MA
Fizički opis rada: 4/111/71/9/0/4/1
(broj poglavlja/strana/lit. citata/tabela/slika/grafika/priloga)
FO
Naučna oblast: Matematika
NO
Naučna disciplina: Numerička matematika
ND
Predmetna odrednica/Ključne reči: Nelinearna optimizacija, stohastička optimizatija, funkcija sa stohastičkim šumom, stohastička aproksimacija, prilagođene dužine koraka, gradijentni pravac, opadajući pravac.
PO

UDK:
Čuva se: u biblioteci Departmana za matematiku i informatiku, Novi Sad
ČU
Važna napomena:
VN
Izvod: Predmet istraživanja doktorske disertacije su numerički postupci za rešavanje problema stohastičke optimizacije. Najpoznatiji numerički postupak za rešavanje pomenutog problema je algoritam stohastičke

aproksimacije (SA). U disertaciji se predlaže nova klasa šema za prilagođavanje dužina koraka u svakoj iteraciji. Odabir dužina koraka u predloženim šemama se zasniva na vrednostima funkcije cilja. U svakoj iteraciji formira se intervalna ocena optimalne vrednosti funkcije cilja koristeći samo registrovane vrednosti funkcije cilja iz fiksnog broja prethodnih iteracija. Ukoliko je vrednost funkcije cilja u trenutnoj iteraciji veća od gornje granice intervala, iteracija se odbacuje. Korak dužine 0 se koristi u narednoj iteraciji. Ako je trenutna vrednost funkcije cilja manja od donje granice intervala, predlaže se duži korak u narednoj iteraciji. Ukoliko vrednost funkcije leži u intervalu, u narednoj iteraciji se koristi korak dobijen harmonijskim pravilom. Na ovaj način se obezbeđuje brži progres algoritma i izbegavaju mali koraci posebno kada se povećava broj iteracija. Šeme izbegavaju korake proporcionalne sa $1/k$ kada se očekuje da će duži koraci poboljšati proces optimizacije. Predložene šeme se razlikuju u intervalima koji se formiraju u svakoj iteraciji. U prvoj predloženoj šemi se formira veštački interval poverenja za ocenu optimalne vrednosti funkcije cilja u svakoj iteraciji. Granice tog intervala se uzimaju za kriterijume dovoljnog smanjenja ili rasta funkcije cilja. Predlaže se i uopštenje ove šeme tako što se umesto srednje vrednosti koristi konveksna kombinacija prethodnih vrednosti funkcije cilja. U drugoj šemi, kriterijum po kom se prilagođavaju dužine koraka su minimum i maksimum prethodnih registrovanih vrednosti funkcije cilja. Nizovi koji se formiranju predloženim šemama zadovoljavaju uslove potrebne za konvergenciju SA algoritma skoro sigurno. SA algoritmi sa novim šemama za prilagođavanje dužina koraka su testirani na standardnim test problemima i upoređeni sa SA algoritmom i njegovim postojećim modifikacijama. Rezultati pokazuju napredak u odnosu na klasičan algoritam stohastičke aproksimacije sa determinističkim nizom dužine koraka kao i postojećim adaptivnim algoritmima. Takođe se razmatra primena novih algoritama na LASSO regresijske modele. Algoritmi su primenjeni za ocenjivanje parametara modela.

IZ

Datum prihvatanja teme od strane NN Veća: 16.07.2015.
DP
Datum odbrane:
DO
Članovi komisije:
Predsednik: dr Nataša Krejić, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu

Član: dr Zorana Lužanin, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu, mentor

Član: dr Sanja Rapajić, vanredni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu

Član: dr Irena Stojkovska, vanredni profesor, Prirodno-matematički fakultet, Univerzitet u Skoplju

Član: dr Zoran Ovcin, docent, Fakultet tehničkih nauka, Univerzitet u Novom Sadu
KO