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

Nataša Krklec Jerinkić

Line search methods with variable sample size

- PhD thesis -

Novi Sad, 2013

.

Introduction

The problem of finding optimal solution is a well known problem which takes place in various areas of life. Therefore, the optimization is recognized and developed as a special part of science for many years. It takes place in many different fields such as economy, engineering, social sciences etc. Roughly speaking, scientific approach of finding optimal solution often involves two phases. The first one consists of building a model and defining the objective function. The next phase is to find the decision variable which provides the optimal value of the objective function. However, building a model is not that easy task. If we include large number of factors, the problem may be very hard or even impossible to solve. On the other hand, excluding too many factors can result in poor approximation of the real problem.

Obtaining a good model that has relatively modest number of variables is a problem itself. Development of the probability theory somewhat facilitates this difficulty. Random variables are often used to collect all the remaining factors and that way the model becomes more complete. Moreover, the problems that involve some future outcomes are the subject of many research efforts. Since the future outcomes are usually not deterministic, random variables are used to describe the uncertainty. That way, stochastic optimization problems are developed. In general, they can be viewed as optimization problems where the objective function is a random variable. However, finding the optimal solution that covers all the scenarios for the future outcomes is often impossible. Therefore, the common approach is to try to find the optimal solution at least for the expected outcome. That way we obtain the problems where the objective function is in the form of mathematical expectation. Moreover, if we assume that there are no constraints on the decision variables, we obtain the problems that are considered within this thesis.

Mathematical expectation with respect to random variable yields a deterministic value. Therefore, the problems that we consider are in fact deterministic optimization problems. However, finding the analytical form of the objective function can be very difficult or even impossible. This is the reason why the sample average is often used to approximate the objective function. Under some mild conditions, this can bring us close enough to the original objective function. Moreover, if we assume that the sample is generated at the beginning of the optimization process, we can consider this sample average function as the deterministic one and therefore the deterministic optimization methods are applicable.

In order to obtain a reasonably good approximation of the objective function, we have to use a relatively large sample size. Since the evaluation of the function under expectation is usually very expensive, the number of these evaluations is a common way of measuring the cost of an algorithm and applying some deterministic method on the sample average function from the start can be very costly. Therefore, methods that vary the sample size throughout the optimization process are developed. Roughly speaking, they can be divided into two classes. The methods from the first class are dealing with to determining the optimal dynamics of increasing the sample size, while the methods from the second class allow decrease of the sample size at some iterations.

The main goal of this thesis is to develop the class of methods that can decrease the cost of an algorithm by decreasing the number of function evaluations. The idea is to decrease the sample size whenever it seems to be reasonable - roughly speaking, we do not want to impose a large precision, i.e. to use a large sample size when we are far away from the solution that we are searching for. The detailed description of the new method is presented in Chapter 4 together with the convergence analysis.

Another important characteristic of the methods that are proposed

here is the line search technique which is used for obtaining the subsequent iterates. The idea is to find a suitable direction and to search along it until we obtain a sufficient decrease in the function value. The sufficient decrease is determined throughout a line search rule. In Chapter 4, that rule is supposed to be monotone, i.e. we are imposing a strict decrease of the function value. In order to decrease the cost of the algorithm even more and to enlarge the set of suitable search directions, we use nonmonotone line search rules in Chapter 5. Within that chapter, these rules are modified to fit the variable sample size framework. Moreover, the convergence analysis is presented and the convergence rate is also discussed.

In Chapter 6, numerical results are presented. The test problems are various - some of them are academic and some of them are real world problems. The academic problems are here to give us more insight into the behavior of the algorithms. On the other hand, data that comes from the real world problems are here to test the real applicability of the proposed algorithms. In the first part of that chapter, the focus is on the variable sample size techniques. Different implementations of the proposed algorithm are compared to each other and to the other sample schemes as well. The second part is mostly devoted to the comparison of the various line search rules combined with different search directions in the variable sample size framework. The overall numerical results show that using the variable sample size can improve the performance of the algorithms significantly, especially when a nonmonotone line search rules are used.

The following chapter provides the background material for the subsequent chapters. In Chapter 2, basics of the nonlinear optimization are presented and the focus is on the line search, while Chapter 3 deals with the relevant stochastic optimization methods. These chapters provide a review of the relevant known results, while the rest of the thesis represent the original contribution.

Acknowledgement

I would like to take this opportunity to express my deep gratitude to my closest family, especially my parents and my husband, for providing me the continuous support throughout all these years. The person who is equally important for my PhD education and who I am especially grateful to is my advisor, professor Nataša Krejić, who encouraged me to cope with the scientific work and unselfishly shared her knowledge with me.

Novi Sad, June 6, 2013 Nataša Krklec Jerinkić

.

Contents

List of Figures

List of Tables

Chapter 1

Overview of the background material

1.1 Functional analysis and linear algebra

We start this section by introducing the basic notation that is used within this thesis. $\mathbb{N}$ represents the set of positive integers, $\mathbb{R}$ denotes the set of real numbers, while $\mathbb{R}^n$ stands for n-dimensional space of real numbers and $\mathbb{R}^{n\times m}$ represents the space of real valued matrices with n rows and m columns. Vector $x \in \mathbb{R}^n$ is considered as a column vector and it will be represented by $x = (x_1, x_2, \ldots, x_n)^T$. The norm $\|x\|$ will represent the Euclidean norm $\|x\|_2$, i.e.

$$\|x\|^2 = \sum_{i=1}^{n} x_i^2$$

and the scalar product is

$$x^T y = \sum_{i=1}^{n} x_i y_i.$$

In general, we denote by $x \geq 0$ the vectors whose components are nonnegative and the space of such vectors by $\mathbb{R}^n_+$.

Since we are working only with the real number spaces, we can define a compact set as a subset of $\mathbb{R}^n$ which is closed and bounded.

Definition 1 *The set X is bounded if there exists a positive constant M such that for every $x \in X$ $\|x\| \leq M$.*

Neighborhood of a point x, i.e. any open subset of $\mathbb{R}^n$ that contains x, is denoted by $\mathcal{O}(x)$. Next, we give the definition of convex combination and convex set.

Definition 2 *A convex combination of vectors $v_1, v_2, \ldots, v_k$ is given by $\sum_{i=1}^{k} \alpha_i v_i$ where $\alpha_1, \alpha_2, \ldots, \alpha_k$ are nonnegative real numbers such that $\sum_{i=1}^{k} \alpha_i = 1$.*

Definition 3 *Set K is a convex set if every convex combination of its elements remains in K.*

We define the distance between two vectors x and y by $d(x, y) = \|x - y\|$. Moreover, the distance of a vector x from a set B is $d(x, B) = inf_{y \in B} d(x, y)$. Finally, we define the distance between two sets as follows.

Definition 4 *Distance between sets A and B is defined by $Dev(A, B) = sup_{x \in A} d(x, B)$*

Now, consider the space of squared matrices $\mathbb{R}^{n\times n}$. The element in the ith row and jth column of the matrix A is denoted by $A_{i,j}$. The identity matrix is denoted by I. Notation $A = 0$ means that every component of the matrix is zero. The determinant of the matrix A is denoted by $|A|$. The inverse of A will be denoted by A^{-1} if it exists and in that case we say that A is nonsingular. If the matrix is positive definite, then we know that it is nonsingular. We state the definition of positive definite and positive semidefinite matrix. We say that vector $x = 0$ if every component of that vector is zero.

Definition 5 *Matrix $A \in \mathbb{R}^{n\times n}$ is positive semidefinite if for every $x \in \mathbb{R}^n$ we have that $x^T Ax \geq 0$. Matrix $A \in \mathbb{R}^{n\times n}$ is positive definite if for every $x \in \mathbb{R}^n$, $x \neq 0$ the inequality is strict, that is $x^T Ax > 0$.*

We denote the Frobenius norm by $\|\cdot\|_F$, i.e.

$$\|A\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}^2.$$

The weighted Frobenius norm is given by

$$\|A\|_W = \|W^{\frac{1}{2}} A W^{\frac{1}{2}}\|_F$$

where $W \in \mathbb{R}^{n\times n}$. Next, we state the Sherman-Morrison-Woodbury formula.

Theorem 1.1.1 *Suppose that $a, b \in \mathbb{R}^n$ and $B = A + ab^T$ where $A \in \mathbb{R}^{n\times n}$ is nonsingular. Then, if B is nonsingular, its inverse is given by*

$$B^{-1} = A^{-1} - \frac{A^{-1}ab^T A^{-1}}{1 + b^T A^{-1} a}.$$

Within this thesis, we work with real valued functions. In general, we consider functions $f : D \to \mathbb{R}^m$ where $D \subseteq \mathbb{R}^n$. The set of functions which are continuous on D is denoted by $C(D)$. The set of functions that have continuous first derivatives on D is denoted by $C^1(D)$. Functions that belong to that set are often referred to as continuously-differentiable or smooth functions. $C^2(D)$ represents the set of functions that have continuous second derivatives and so on.

Lipschitz continuous function are very important for the analysis in this thesis and therefore we give the following definition.

Definition 6 *Function $f : D \to \mathbb{R}^m$, $D \subseteq \mathbb{R}^n$ is Lipschitz continuous on the set $D \subseteq D$ if there exists a constant $L \geq 0$ such that for every $x, y \in D$*

$$\|f(x) - f(y)\| \leq L\|x - y\|.$$

The first derivative of the function $f(x) = (f_1(x), \ldots, f_m(x))^T$ is often referred to as the Jacobian and it is denoted by $J(x)$. Its components are

$$(J(x))_{i,j} = \frac{\partial f_i(x)}{\partial x_j}.$$

We state an important property of the Jacobian throughout Mean Value Theorem.

Theorem 1.1.2 *Suppose that the function $f : D \to \mathbb{R}^m$, $D \subseteq \mathbb{R}^n$ is continuously-differentiable on the set D. Then, for every $x, y \in D$ there exists $t \in (0, 1)$ such that*

$$f(y) - f(x) = J(x + t(y - x))(y - x).$$

Moreover,

$$f(y) - f(x) = \int_0^1 J(x + t(y - x))(y - x)dt.$$

We are especially interested in the case where $m = 1$ that is when $f : D \to \mathbb{R}$. In that case, we denote the first derivative of the function f by ∇f and call it the gradient. The gradient is assumed to be a column vector,

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)^T.$$

Moreover, we denote the second derivative by $\nabla^2 f$. The second derivative is often called the Hessian. Its elements are

$$\left(\nabla^2 f(x)\right)_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}.$$

The following result holds for this special case when it comes to derivatives.

Theorem 1.1.3 *Suppose that $f \in C^1(D)$, $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$. Then, for every $x, y \in D$ there exists $t \in (0,1)$ such that*

$$f(y) - f(x) = \nabla^T f(x + t(y - x))(y - x).$$

Moreover,

$$f(y) - f(x) = \int_0^1 \nabla^T f(x + t(y - x))(y - x) dt.$$

If the function is twice continuously-differentiable, then we can apply the second order Taylor's series to obtain the following result.

Theorem 1.1.4 *If $f \in C^2(D)$, $f : D \to \mathbb{R}$, then for every $x, y \in D$ there exists $t \in (0,1)$ such that*

$$f(y) = f(x) + \nabla^T f(x)(y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x + t(y - x))(y - x).$$

Next, we provide the definition of the directional derivative.

Definition 7 *The directional derivative of the function $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$ at the point x in the direction d is given by*

$$\lim_{h \to 0} \frac{f(x + hd) - f(x)}{h}.$$

If the gradient exists, then the directional derivative is of the following form.

Theorem 1.1.5 *Suppose that $f \in C^1(D)$, $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$. Then the directional derivative of the function f at the point x in the direction d is given by $\nabla^T f(x)d$.*

The class of convex function is a very important class which is going to be considered within this thesis. Therefore, we give the definition of convex function.

Definition 8 *Function $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$ is convex if for every $x, y \in D$ and every $\alpha \in [0, 1]$*

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha) f(y).$$

The function is strongly convex if for every $x, y \in D$ and every $\alpha \in (0, 1)$

$$f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha) f(y).$$

We also give the following characterizations of convex functions.

Theorem 1.1.6 *Suppose that $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$ and $f \in C^1(D)$. Then the function f is convex if and only if for every $x, y \in D$*

$$f(x) \geq f(y) + \nabla^T f(y)(x - y).$$

Furthermore, the function is strongly convex if and only if there exists a positive constant γ such that for every $x, y \in D$

$$f(x) \geq f(y) + \nabla^T f(y)(x - y) + \frac{1}{2\gamma}\|x - y\|^2.$$

Theorem 1.1.7 *Suppose that $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$ and $f \in C^2(D)$. Then the function f is convex if and only if the Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for every $x \in D$. The function is strongly convex if and only if the Hessian $\nabla^2 f(x)$ is positive definite for every $x \in D$.*

Within this thesis, we are particularly interested in conditions that yield convergence. However, almost equally important is the rate of convergence. Therefore, we state the following definition.

Definition 9 *Suppose that the sequence $\{x_k\}_{k\in\mathbb{N}}$ converges to x^*. The convergence is Q-linear if there is a constant $\rho \in (0,1)$ such that for all k sufficiently large*

$$\|x_{k+1} - x^*\| \leq \rho \|x_k - x^*\|.$$

The convergence is Q-superlinear if

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

The convergence is Q-quadratic if there exists a positive constant M such that for all k sufficiently large

$$\|x_{k+1} - x^*\| \leq M \|x_k - x^*\|^2.$$

The convergence is R-linear if for all k sufficiently large

$$\|x_k - x^*\| \leq a_k$$

where $\{a_k\}_{k\in\mathbb{N}}$ is a sequence which converges to zero Q-linearly.

We conclude this subsection by stating Taylor's expansion in the case where $n = m = 1$. In that case, the derivative of order k is denoted by $f^{(k)}$. Especially, the first and the second derivative are usually denoted by f' and f'', respectively.

Theorem 1.1.8 *Suppose that the function $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}$ is k times continuously-differentiable, i.e. $f \in C^k(D)$. Then for every $x, y \in D$ there exists $\theta \in [0,1]$ such that*

$$f(y) = f(x) + \sum_{j=1}^{k-1} \frac{f^{(j)}(x)}{j!}(y-x)^j + \frac{f^{(k)}(x+\theta(y-x))}{k!}(y-x)^k.$$

1.2 Probability theory

In this thesis, we deal only with real valued random variables. The set of all possible outcomes is denoted by Ω. Then any subset of Ω is called an event. In order to define a random variable, we need to state the definition of a σ-field. We denote the partitive set of Ω by $\mathcal{P}(\Omega)$, while the complementary set of the set A is $\bar{A} = \Omega \setminus A$.

Definition 10 *Suppose that $\mathcal{F} \subseteq \mathcal{P}(\Omega)$. Then $\mathcal{F}$ is a σ-field on Ω if the following conditions are satisfied:*

- $\Omega \in \mathcal{F}$,
- *if* $A \in \mathcal{F}$, *then* $\bar{A} \in \mathcal{F}$,
- *if* $\{A_k\}_{k\in\mathbb{N}} \subseteq \mathcal{F}$, *then* $\bigcup_{k=1}^{\infty} A_k \in \mathcal{F}$.

Now, we can define the probability function. The empty set if denoted by $\varnothing$.

Definition 11 *The function $P : \mathcal{F} \to [0,1]$ is called the probability function on a space $(\Omega, \mathcal{F})$ if it satisfies the following conditions:*

- $P(\Omega) = 1$,

- *if* $\{A_k\}_{k\in\mathbb{N}} \subseteq \mathcal{F}$ *and* $A_i \bigcap A_j = \varnothing$ *for* $i \neq j$ *then*

$$P\left(\bigcup_{k=1}^{\infty} A_k\right) = \sum_{k=1}^{\infty} P\left(A_k\right).$$

It can be shown that this definition yields $P(\varnothing) = 0$. This furthermore implies that the second condition of the definition also holds for any finite number of events. In general, we have that
$P(A_1 \cup \ldots \cup A_k)$ is given by

$$\sum_{i=1}^{k} P(A_i) - \sum_{i<j} P(A_i \cap A_j) + \ldots + (-1)^{k+1} P(A_k | A_1 \cap \ldots \cap A_k).$$

One of the properties that is often used is

$$P(\bar{A}) = 1 - P(A).$$

If $P(B) > 0$, then we can define the conditional probability by

$$P(A|B) = \frac{P(A \bigcap B)}{P(B)}.$$

Moreover, the following holds

$$P(A_1 \cap \ldots \cap A_k) = P(A_1) P(A_2|A_1) \cdots P(A_k|A_1 \cap \ldots \cap A_{k-1}).$$

We also state an important definition of independent events.

Definition 12 *The sequence of events* $A_1, A_2, \ldots$ *from* $\mathcal{F}$ *is independent if for every finite sequence of indices* $k_1 < \ldots < k_s$ *the following equality holds*

$$P(A_{k_1} \cap \ldots \cap A_{k_s}) = P(A_{k_1}) \cdots P(A_{k_s})$$

The space $(\Omega, \mathcal{F}, P)$ is called the probability space. Next, we define Borel's σ-field.

Definition 13 *Borel's σ-field $\mathcal{B}$ in topological space $(\mathbb{R}, \tau)$ is the smallest σ-field that contains τ.*

Now, we can define random variable.

Definition 14 *Mapping $X : \Omega \to \mathbb{R}$ is a random variable on the space $(\Omega, \mathcal{F}, P)$ if $X^{-1}(S) \in \mathcal{F}$ for every $S \in \mathcal{B}$.*

The cumulative distribution function $F_X : \mathbb{R} \to [0, 1]$ for the random variable X is given by

$$F_X(x) = P\left(X < x\right).$$

Furthermore, we define the quantile z_α as follows.

$$z_\alpha = \inf_{F_X(t) \geq \alpha} t.$$

In many cases it can be viewed as the number which satisfies

$$\alpha = F_X(z_\alpha).$$

Random variables can be discrete or continuous. A discrete random variable may take only countable many different values. For example, indicator function is of that kind. We denote it by I_A, i.e.

$$I_A(\omega) = \begin{cases} 1, & \omega \in A \\ 0, & \omega \in \bar{A} \end{cases}.$$

Random variable X is continuous if there exists a nonnegative function φ_X such that for every $S \in \mathcal{B}$

$$P(S) = \int_S \varphi_X(x) dx.$$

In that case, we call φ_X the probability density function or just density. One of the most important random variable from this class is the one with the normal distribution $\mathcal{N}$. If X is normally distributed random variable with the mean m and the variance σ^2, i.e. if $X : \mathcal{N}(m, \sigma^2)$, then its density function is

$$\varphi_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-m)^2}{2\sigma^2}}.$$

Especially important case is when $m = 0$ and $\sigma^2 = 1$. Then we say that X has the standard normal distribution. Moreover, if $X : \mathcal{N}(0, 1)$ we can define the Laplace function Φ which satisfies

$$\Phi(x) = F_X(x) - 0.5$$

for every $x \geq 0$ and $\Phi(x) = -\Phi(-x)$ for $x < 0$.

We also state the cumulative distribution function for the Gumbel distribution with the location parameter μ and the scale parameter β

$$F_X(x) = e^{-e^{-(x-\mu)/\beta}}.$$

We say that $\mathbf{X} = (X_1, \ldots, X_n) : \Omega \to \mathbb{R}^n$ is a random vector if every component X_i is a random variable. The cumulative distribution function for the random vector is defined by

$$F_{\mathbf{X}}(x_1, \ldots, x_n) = P(X_1 < x_1 \cap \ldots \cap X_n < x_n).$$

Random vector $\mathbf{X}$ is continuous if and only if there exists density function $\varphi_{\mathbf{X}} \geq 0$ such that for every $S \in \mathcal{B}(\mathbb{R}^n)$

$$P(\mathbf{X} \in S) = \int \ldots \int_S \varphi_{\mathbf{X}}(x_1, \ldots, x_n) dx_1 \ldots dx_n$$

Multidimensional normal distribution $\mathcal{N}(m, V)$ is given by the density function

$$\varphi_{\mathbf{X}}(x) = \frac{1}{\sqrt{(2\pi)^n |V|}} e^{-\frac{1}{2}(x-m)^T V^{-1}(x-m)}.$$

where $m = (m_1, \ldots, m_n)$ and V is a matrix which is usually called the covariance matrix. We state an important result regarding this distribution.

Theorem 1.2.1 *If a random vector has multidimensional normal distribution, then every component of that vector has one-dimensional normal distribution.*

If we are dealing with more that one random variable, an important question is whether they are independent.

Definition 15 *Random variables $X_1, X_2, \ldots$ are independent if the events $X_1^{-1}(S_1), X_2^{-1}(S_2) \ldots$are independent for all $S_i \in \mathcal{B}(\mathbb{R})$, $i = 1, 2, \ldots$.*

Suppose that (X, Y) is a random vector. If the components of that vector are discrete, then X and Y are independent if and only if for every x_i and every y_j

$$P(X = x_i \cap Y = y_j) = P(X = x_i)P(Y = y_j).$$

On the other hand, if they are continuous, X and Y are independent if and only if for every $(x, y) \in \mathbb{R}^2$

$$\varphi_{(X,Y)}(x, y) = \varphi_X(x)\varphi_Y(y).$$

If random variables are independent and they have the same distribution we say that they are i.i.d. (independent and identically distributed). Suppose that $Z_1, \ldots, Z_n$ are i.i.d. with standard normal distribution $\mathcal{N}(0, 1)$. Then we say that the random variable

$$\chi_n^2 = Z_1^2 + \ldots + Z_n^2$$

has the chi-squared distribution with n degrees of freedom. The density function is given by

$$\varphi_{\chi_n^2}(x) = \begin{cases} \frac{1}{2^{n/2}\Gamma(n/2)} x^{n/2-1} e^{-x/2}, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

where Γ is the gamma function defined by

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt.$$

Moreover, if we define

$$t_n = \frac{Z}{\sqrt{\frac{\chi_n^2}{n}}}$$

where $Z : \mathcal{N}(0,1)$, we obtain Student's t-distribution with n degrees of freedom. The relevant density function is

$$\varphi_{t_n}(x) = \frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})\sqrt{n\pi}}\left(1+\frac{x^2}{n}\right)^{-\frac{k+1}{2}}.$$

Moreover, it can be shown that for every $x \in \mathbb{R}$

$$\lim_{n\to\infty}\varphi_{t_n}(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$$

and therefore Student's t-distribution is often approximated with the standard normal distribution.

Let us consider some numerical characteristics of random variables. We define the mathematical expectation as follows.

Definition 16 *If X is a discrete random variable, then the mathematical expectation $E(X)$ exists if and only if $\sum_{k=1}^\infty |x_k|P(X=x_k) < \infty$ where $x_1, x_2, \ldots$ are the values that X may take and it is given by*

$$E(X) = \sum_{k=1}^\infty x_k P(X = x_k).$$

If X is continuous, the mathematical expectation exists if $\int_{-\infty}^\infty |x|\varphi_X(x)dx < \infty$ and it is defined by

$$E(X) = \int_{-\infty}^\infty x\varphi_X(x)dx.$$

Now, we state some characteristics of the mathematical expectation. We say that an event happens almost surely if it happens with probability 1.

Theorem 1.2.2 *Let $X_1, X_2, \ldots, X_n$ be random variables that poses the mathematical expectations and $c \in \mathbb{R}$. Then the following holds.*

- $|E(X_k)| \leq E(|X_k|)$.
- $E(c) = 0$.
- $E(cX_k) = cE(X_k)$.
- $E(X_1 + \ldots + X_n) = E(X_1) + \ldots + E(X_n)$.
- *If $X_k \geq 0$ almost surely, then $E(X_k) \geq 0$.*
- *If $X_1, X_2, \ldots, X_n$ are independent, then*

$$E\left(\prod_{k=1}^{n} X_k\right) = \prod_{k=1}^{n} E(X_k).$$

- *If $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ is a random vector, then*

$$E(\mathbf{X}) = \left(E(X_1), \ldots, E(X_n)\right).$$

- *If $\mathbf{X}$ is continuous and $\mathbf{x}$ represents n dimensional vector, then*

$$E(\mathbf{X}) = \int_{\mathbb{R}^n} \mathbf{x}\varphi_{\mathbf{X}}(\mathbf{x})d\mathbf{x}.$$

Before we state one more important feature of mathematical expectation, we need to define Borel's function.

Definition 17 *$f : \mathbb{R}^n \to \mathbb{R}^m$ is a Borel's function if for every S from Borel's σ field $\mathcal{B}(\mathbb{R}^m)$ the inverse $f^{-1}(S)$ belongs to Borel's σ field $\mathcal{B}(\mathbb{R}^n)$.*

Theorem 1.2.3 *Let $f : \mathbb{R} \to \mathbb{R}$ be a Borel's function. Then, if X is discrete the mathematical expectation of $f(X)$ is*

$$E(f(X)) = \sum_{k=1}^{\infty} f(x_k)P(X = x_k)$$

and if X is continuous

$$E(f(X)) = \int_{-\infty}^{\infty} f(x)\varphi_X(x)dx.$$

The variance is also a very important feature of random variables. We denote it by $D(X)$, but it is also common to use the notation $Var(X)$ or $\sigma^2(X)$. Before we define the variance, we give the definitions of the moments and the central moments.

Definition 18 *Let X be a random variable and $k \in \mathbb{N}$. Then the moment of order k of X is given by $E(X^k)$, while the central moment of order k is $E\left((X - E(X))^k\right)$.*

Definition 19 *The variance of a random variable X is the second order central moment of that random variable, i.e.*

$$D(X) = E\left((X - E(X))^2\right).$$

However, the variance if more often calculated by using the following formula which can easily be obtained from the definition

$$D(X) = E(X^2) - E^2(X).$$

The standard deviation is usually denoted by $\sigma(X)$ and it is equal to $\sqrt{D(X)}$.

Theorem 1.2.4 *Let $X_1, X_2, \ldots, X_n$ be random variables with the variances $D(X_1), D(X_2), \ldots, D(X_n)$ and $c \in \mathbb{R}$. Then the following holds.*

- $D(X_k) \geq 0$.
- $D(X_k) = 0$ *if and only if* X_k *is a constant almost surely.*
- $D(cX_k) = c^2 D(X_k)$.
- $D(X_k + c) = D(X_k)$.
- *If* $X_1, X_2, \ldots, X_n$ *are independent, then*

$$D\left(\sum_{k=1}^{n} X_k\right) = \sum_{k=1}^{n} D(X_k).$$

- *If* $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ *is a random vector, then*

$$D(\mathbf{X}) = \left(D(X_1), \ldots, D(X_n)\right).$$

Now, let X be a random variable. The random variable of the following form

$$X^* = \frac{X - E(X)}{\sqrt{D(X)}}$$

is called the standardized random variable. Moreover, we obtain from the characteristics of the mathematical expectation and the variance that $E(X^*) = 0$ and $D(X^*) = 1$. Especially, for a normally distributed random variable $X : \mathcal{N}(m, \sigma^2)$ we have that $E(X) = m$, $D(X) = \sigma^2$ and $X^* : \mathcal{N}(0, 1)$.

Finally, we define the covariance and the correlation.

Definition 20 *Let X and Y be some random variables. The covariance of these variables is given by*

$$cov(X,Y) = E\left((X - E(X))(Y - E(Y))\right).$$

The covariance is often calculated as

$$cov(X,Y) = E(XY) - E(X)E(Y).$$

Moreover, the following equality holds

$$D(X - Y) = D(X) + D(Y) - 2cov(X,Y).$$

Definition 21 *The correlation between random variables X and Y is given by*

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sqrt{D(X)D(Y)}}.$$

Now, we define four basic types of convergence concerning random variables.

Definition 22 *A sequence of random variables $X_1, X_2, \ldots$ converges in probability towards random variable X if for every $\varepsilon > 0$*

$$\lim_{k\to\infty} P(|X_k - X| \geq \varepsilon) = 0.$$

Definition 23 *A sequence of random variables $X_1, X_2, \ldots$ converges almost surely towards random variable X if*

$$P(\lim_{k\to\infty} X_k = X) = 1.$$

Definition 24 *A sequence of random variables $X_1, X_2, \ldots$ converges in mean square towards random variable X if the following conditions hold*

- $E(X_k^2) < \infty$ *for every* $k \in \mathbb{N}$
- $\lim_{k\to\infty} E((X_k - X)^2) = 0$.

Definition 25 *A sequence of random variables* $X_1, X_2, \ldots$ *converges in distribution towards random variable* X *if, for every* $x \in \mathbb{R} \cup \{-\infty, \infty\}$ *such that* $F_X(x)$ *is continuous, the following holds*

$$\lim_{k\to\infty} F_{X_k}(x) = F_X(x).$$

Convergence in mean square implies convergence in probability. Also, almost sure convergence implies convergence in probability. Convergence in probability furthermore implies convergence in distribution. Moreover, if a sequence of random variables converges to a constant, then convergence in distribution implies convergence in probability.

Let us consider a sequence of independent random variables $X_1, X_2, \ldots$ and define

$$Y_n = \frac{1}{n}\sum_{k=1}^{n} X_k, \quad S_n = Y_n - E(Y_n).$$

We state the conditions under which the sequence $\{S_n\}_{n\in\mathbb{N}}$ converges to zero. Convergence in probability is stated in the so called weak laws of large numbers, while the strong laws of large numbers consider almost sure convergence.

Theorem 1.2.5 *If there exists a constant* C *such that* $D(X_k) \leq C$ *for every* $k \in \mathbb{N}$*, then the sequence* $\{S_n\}_{n\in\mathbb{N}}$ *converges in probability towards zero.*

Theorem 1.2.6 *If the random variables* $X_1, X_2, \ldots$ *have a same distribution and a finite mathematical expectation* $E(X_k) = a$*, then the sequence* $\{S_n\}_{n\in\mathbb{N}}$ *converges in probability towards zero, or equivalently the sequence* $\{Y_n\}_{n\in\mathbb{N}}$ *converges in probability towards* a*.*

Theorem 1.2.7 *If the random variables $X_1, X_2, \ldots$ have a same distribution and finite variance, then the sequence $\{S_n\}_{n\in\mathbb{N}}$ converges almost surely towards zero.*

If we denote the mathematical expectation in the previous theorem by a, then we obtain that the sequence $\{Y_n\}_{n\in\mathbb{N}}$ converges to a almost surely. Finally, we state the Central limit theorem.

Theorem 1.2.8 *If the random variables $X_1, X_2, \ldots$ have a same distribution and a finite variance, then for every x*

$$\lim_{n\to\infty} P\left(\frac{S_n}{\sqrt{D(Y_n)}} < x\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt.$$

Chapter 2

Nonlinear optimization

Within this chapter we are going to set the deterministic framework for algorithms described in chapters 5 and 6. In order to do that, we will provide some basics for unconstrained continuous optimization. Mainly, there are two basic approaches used to solve the nonlinear optimization problems - the line search and the trust region. The latter ones are not going to be described in detail since our algorithms relay on the line search only. Special class within the line search framework is represented by nonmonotone line search methods. They are especially useful in stochastic environment as we will see in chapter 6. Therefore, we will present various nonmonotone techniques at the end of this chapter.

2.1 Unconstrained optimization

Let us begin by introducing the optimization problem in a frequently used general form. Consider a real valued function $f : D \to \mathbb{R}$ where D is a subset of finite dimensional space $\mathbb{R}^n$. This function may represent some quantitative measure of the state of the system under consider-

ation. It is called the objective function since it is the value that we are trying to control. In optimization problems, controlling this function means finding a value where it attains minimum or maximum. However, finding a maximum of function f is an equivalent problem to finding a minimum of $-f$. Therefore, without loss of generality, we will consider only the minimization problems. The argument of function f will usually be denoted by x. It can be considered as a tool for controlling the value of the system output. For example, if we are dealing with finance and we want to maximize the profit, the objective function f to be minimized would be the loss (the negative profit) and x would be the vector with components representing the share of wealth (capital) invested in each of n different financial assets. This example may seem more suitable for illustrating the stochastic optimization problem since the outcome of the system is highly uncertain. However, if we decide to maximize the expected profit, the problem could formally be considered as a deterministic one. Moreover, it is convenient for introducing the concept of constrained optimization because most naturally imposed conditions on vector x are nonnegativity of the components that have to sum up to 1. In that case, the set where we are locking for a potential solution is $D = \{x \in \mathbb{R}^n | \quad x \geq 0, \quad \sum_{i=1}^{n} x_i = 1\}$.

In general, the optimization problem can be stated as

$$\min_{x \in D} f(x). \tag{2.1}$$

The set D is often called the feasible set and it is usually represented in the form

$$D = \{x \in \mathbb{R}^n | \quad g_i(x) \leq 0, i = 1, \ldots, s, \quad h_i(x) = 0, i = 1, \ldots, m\}, \tag{2.2}$$

where $g_1, \ldots, g_s, h_1, \ldots, h_m$ are real valued functions that represent inequality and equality constraints. Usually, these functions, together

with the objective function f are supposed to be at least continuous. In opposite to continuous optimization, discrete optimization deals with the feasible set which contains only countable many points. However, these kind of problems will not be considered here.

An important special case of problem (2.1) is when $D = \mathbb{R}^n$. This is exactly the formulation of unconstrained optimization problems. In order to make this problem solvable, the objective function f has to be bounded from below. Constrained problems can be converted to unconstrained using a penalty function. For example, we can incorporate the constraint functions in the objective function which yields the new merit function to be minimized. One way of doing that is

$$\Phi(x;\mu) = f(x) + \mu \sum_{i=1}^{m} |h_i(x)| + \mu \sum_{i=1}^{s} [g_i(x)]^+,$$

where μ is called the penalty parameter and $[z]^+ = \max\{z, 0\}$. Penalty parameter is a positive constant. Increasing this parameter means that we are more rigorous when constraints are violated. This particular penalty function has the property of being exact which means that for penalty parameter which is large enough, any local solution of constrained optimization problem is a local minimizer of penalty function. However, the function presented above is not that convenient because it does not have the property of being differentiable. But there are penalty functions that posses that property. Suppose that we have only the equality constraints and that $h_1, \ldots, h_m$ and f are smooth. Then we can form the smooth penalty function

$$\Phi(x;\mu) = f(x) + \mu \|h(x)\|^2,$$

where $h(x) = (h_1(x), \ldots, h_m(x))^T$. If there are inequality constraints also, we can introduce a slack variable $y \in \mathbb{R}^s_+$ and form the new set of equality constraints $g(x) + y = 0$ where $g(x) = (g_1(x), \ldots, g_s(x))^T$.

Then we obtain the problem of minimizing the function

$$\Phi(x, y; \mu) = f(x) + \mu \sum_{i=1}^{m} h_i^2(x) + \mu \sum_{i=1}^{s} (g_i(x) + y_i)^2,$$

subject to $y \geq 0$. Although this is not exactly the unconstrained optimization problem, it is close enough to an unconstrained problem since nonnegativity constraints can easily be incorporated in almost any algorithm for unconstrained optimization. There are also merit functions that are both smooth and exact. For example, Fletcher's augmented Lagrangian can have that property but it also requires higher derivative information which makes the practical implementation more expensive. For further references on this topic see Nocedal, Wright [46].

Now, let us formally state the unconstrained optimization problem that we will consider in the sequel

$$\min_{x \in \mathbb{R}^n} f(x). \tag{2.3}$$

Function f is nonlinear in general and it is assumed to be continuously-differentiable and bounded from below. As one can see, the goal of optimization is to find the argument that provides the lowest value of function f, i.e. we are looking for

$$x^* = argmin f(x),\ x \in \mathbb{R}^n.$$

Unfortunately, this is often too much to ask for. Global solution - a point x^* which satisfies $f(x^*) \leq f(x)$ for every $x \in \mathbb{R}^n$, frequently remains unreachable even if it exists. A local solution is the one that we settle for in most cases. By definition, x^* is a local solution (minimizer) if there exists some neighborhood $\mathcal{O}(x^*)$ such that for every $x \in \mathcal{O}(x^*)$, $f(x^*) \leq f(x)$. If the previous inequality is strict, then we say that x^* is a strict local minimizer. Analogous definition stands for

a strict global minimizer. However, if the function f is smooth there are more practical ways of characterizing the solution. We will state the theorem that provides the first-order necessary conditions for a local solution.

Theorem 2.1.1 *[46] If x^* is a local minimizer of function f and f is continuously differentiable on $\mathcal{O}(x^*)$, then $\nabla f(x^*) = 0$.*

The point x^* that satisfies condition $\nabla f(x^*) = 0$ is called a stationary point of function f. Therefore, every local minimizer must be a stationary point. If the objective function is two times continuously differentiable, then we can state the second-order necessary conditions for a local minimizer.

Theorem 2.1.2 *[46] If x^* is a local minimizer of function f and $f \in C^2(\mathcal{O}(x^*))$ then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.*

Previous two theorems give us conditions for local solutions. If $\nabla f(x^*) \neq 0$ or if there is some vector h such that $h^T \nabla^2 f(x^*) h < 0$, then we know that x^* is not a local minimizer. Next theorem provides the second-order sufficient conditions. It means that if these conditions are satisfied, we can say that the point under consideration is at least local solution. Moreover, it is a strict local minimizer. However, the following conditions require the positive definitness of the Hessian $\nabla^2 f(x^*)$.

Theorem 2.1.3 *[46] Suppose that $f \in C^2(\mathcal{O}(x^*))$. Moreover, suppose that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Then x^* is a strict local minimizer of function f.*

The important special case of optimization problem (2.3) is when the objective function is convex. Then any local minimizer is in fact a global minimizer and therefore every stationary point of the function

f is a solution that we are searching for. Furthermore, this will be an assumption that is necessary for proving the convergence rate result of our algorithm.

2.2 Line search methods

In order to clarify the concept of the line search methods, we have to take in consideration iterative way of solving the optimization problems. That means that we want to construct the sequence of points that will (hopefully) converge towards a solution of the problem (2.3). These points are called iterates and the sequence of iterates is usually denoted by $\{x_k\}_{k\in\mathbb{N}}$. If we want to construct this sequence, we have to provide the starting point - the initial iterate x_0. Sometimes we are able to localize the solution, i.e. to find a subset of $\mathbb{R}^n$ which contains it. Then the starting point is chosen from that subset. This localization plays an important role in optimization because there are iterative methods that are only locally convergent. This means that the sequence of iterates will converge in the right direction (towards a solution) only if the starting point is close enough to a minimizer. Unfortunately, the region around the solution where we should start is usually given by some theoretical means and it is hardly detectable in practice.

When we choose the starting point, we need a rule which gives us the following iterate. Suppose that we are at the iteration k, i.e. at the iterate x_k and we need to find a step s_k such that

$$x_{k+1} = x_k + s_k.$$

Two important features of s_k are the direction and the length of that step. Both of them need to be determined in some way. The question is what will be determined first. The choice yields two different concepts that were already mentioned. If we choose to put the boundary on

the length of s_k and then find the best possible direction, we are using the trust region method. More precisely, the idea is to make a model function m_k which approximates the behavior of the objective function f in a region around x_k. The model function is usually quadratic in a form of

$$m_k(s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T B_k s,$$

where B_k is some approximation of the Hessian $\nabla^2 f(x_k)$. The region around x_k is called the trust region because it is the region in which we believe that the model function is a good representation of the objective function f. It is given by the so called trust region radius usually denoted by Δ_k. Therefore, the problem that we are solving in the iteration k (at least approximately) is

$$\min_s m_k(s) \quad \text{subject to} \quad \|s\| \leq \Delta_k.$$

This is the concept that has been the subject of interest for many researchers. Comprehensive material on this topic can be found in Conn et al. [13].

While the trust region method puts the step length first, the line search starts iteration by choosing a direction that points to the next iterate. Let us denote this direction by p_k. After that, we are trying to find the optimal length of p_k, i.e. we search along this direction for the point that has the lowest function value. Therefore, the following problem is solved exactly or approximately

$$\min_{\alpha > 0} f(x_k + \alpha p_k). \tag{2.4}$$

The positive number α is often called the step size. The step size that (approximately) solves the problem (2.4) at iteration k is denoted by α_k. The next iteration is then defined as

$$x_{k+1} = x_k + \alpha_k p_k.$$

Now, the question is how to choose the search direction and how to solve the problem (2.4). First, we will see what kind of a search direction is desirable and what are the most common choices for p_k.

2.2.1 Search directions

Recall that the objective function is assumed to be smooth. Therefore the Taylor series yields

$$f(x_k + \alpha p_k) = f(x_k) + \alpha p_k^T \nabla f(x_k) + O(\alpha^2).$$

Our primary goal at every iteration is to obtain a point that is better than the current one. If we look at the previous equality, we can conclude that negativity of $p_k^T \nabla f(x_k)$ implies the existence of a small enough step size such that $f(x_k + \alpha_k p_k) < f(x_k)$. Therefore, the condition that the search direction should satisfy is

$$p_k^T \nabla f(x_k) < 0. \tag{2.5}$$

Direction that satisfies previous inequality is called descent search direction for the function f at the iteration k.

One of the choices for a descent search direction is the negative gradient. The method that uses $p_k = -\nabla f(x_k)$ is called the steepest descent method. Notice that the only case when negative gradient does not satisfy the condition (2.5) is when $\nabla f(x_k) = 0$, i.e. when x_k is a stationary point of function f. This method is appealing since it is cheep in the sense that it does not require any second order information. Therefore, it is widely applicable. However, this method can be very slow - the convergence rate is at most linear. Moreover, it is sensitive to poor scaling. Poor scaling happens when the objective function is much more sensitive in some components of the argument than the others.

Another important method is the Newton method. In some sense it is opposite to the steepest descent method. While the steepest descent is cheep and slow, the Newton method is expensive and fast. Assume that the objective function is in $C^2(\mathbb{R}^n)$ and consider the following model function at the iteration k

$$m_k(p) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p.$$

This model function is an approximation of $f(x_k + p)$ and therefore our goal is to minimize it. If we assume that the Hessian $\nabla^2 f(x_k)$ is positive definite, then the unique minimizer of the function m_k is

$$p_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k).$$

This direction is called the Newton direction and it is descent if the Hessian is positive definite. If we are in a neighborhood of a strict local minimizer where $\nabla^2 f(x^*)$ is sufficiently positive definite, the Hessian matrices will also be positive definite and the Newton method will perform very well yielding the potential to achieve quadratic local convergence. The problem arises if the Hessian at x^* is nearly singular or if we are far away from a solution. Then, the method can be unstable or even undefined and the modifications that make the Hessian matrices sufficiently positive definite are needed. The modifications can be, for instance, adding a multiple of the identity matrix or applying the modified Cholesky factorization [46]. The idea is to obtain a positive definite approximation of the Hessian matrix which can be written in the form $B_k = \nabla^2 f(x_k) + E_k$, where E_k is the correction matrix. After that, we define the search direction as

$$p_k = -B_k^{-1} \nabla f(x_k). \tag{2.6}$$

For this kind of methods, convergence result of the form $\lim_{k\to\infty} \nabla f(x_k) = 0$ can be obtained under the assumption of uniformly bounded conditional numbers $\|B_k\| \|B_k^{-1}\|$. The rate of convergence depends on the $\nabla^2 f(x^*)$. If the Hessian is sufficiently positive

definite, the correction matrix will eventually became $E_k = 0$ and the method transforms to pure Newton's method which yields the quadratic convergence. On the other hand, the rate is no more than linear if the Hessian is nearly singular.

Although the Newton method has many nice properties, it can be too expensive since it requires the computation of the second derivatives at every iteration. To avoid this, quasi-Newton methods are developed. The idea is to construct a matrix that approximates the Hessian matrix by updating the previous approximation and using the first order information. The rate of convergence is no more than superlinear, but the cost is significantly smaller then in Newton's method.

Quasi-Newton method provides the search direction in the form of (2.6) where B_k is an approximation of the Hessian. Define the discrepancy between the gradients in two neighboring iterations by

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

This difference together with the difference between two iterates

$$s_k = x_{k+1} - x_k$$

is used to obtain the approximation of the second order derivative. Now, the question is how do we choose B_k or more precisely, how do we update the current approximation to obtain B_{k+1}? The main condition that B_{k+1} should satisfy is the secant equation

$$B_{k+1}s_k = y_k. \tag{2.7}$$

This comes from approximating $B_{k+1} \approx \nabla^2 f(x_k + t_k s_k)$ in Taylor's expansion

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_k + t_k s_k)s_k$$

where $t_k \in (0, 1)$. Another way of viewing this condition is to construct the model function

$$m_{k+1}(s) = f(x_{k+1}) + (\nabla f(x_{k+1}))^T s + \frac{1}{2}s^T B_{k+1}s$$

that approximates $f(x_{k+1}+s)$ and to require the match of the gradient at points x_k and x_{k+1}, i.e. to demand $\nabla m_{k+1}(0) = \nabla f(x_{k+1})$ and $\nabla m_{k+1}(-s_k) = \nabla f(x_k)$. The first condition is already satisfied while the second one yields the secant equation (2.7). However, the secant equation does not provide an unique solution for B_{k+1}. Therefore, other conditions are imposed such as low rank of $B_{k+1} - B_k$ and the symmetry of B_{k+1}. Therefore, we can set B_{k+1} to be the solution of the problem

$$\min \|B - B_k\| \quad \text{subject to} \quad B^T = B, \quad Bs_k = y_k. \tag{2.8}$$

Different norms provide different updating formulas. If we use the weighted Frobenius norm [46], we obtain the DFP (Davidon-Fletcher-Powell) formula

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

where I is the identity matrix. Since we have to solve the system of linear equations to obtain the search direction, it is sometimes more effective to work with the inverse Hessian approximations $H_k \approx (\nabla^2 f(x_k))^{-1}$. Sherman-Morrison-Woodbury formula provides the updating formula that correspond to the DFP update

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

Since this is an approximation of the inverse Hessian, i.e. $H_k = B_k^{-1}$, the secant equation becomes

$$H_{k+1} y_k = s_k. \tag{2.9}$$

The other approach is to use the BFGS (Broyden-Fletcher-Goldfarb-Shanno) formula

$$H_{k+1} = (I - \frac{1}{y_k^T s_k} s_k y_k^T) H_k (I - \frac{1}{y_k^T s_k} y_k s_k^T) + \frac{1}{y_k^T s_k} s_k s_k^T. \tag{2.10}$$

We obtain this formula by solving the problem

$$\min \|H - H_k\| \quad \text{subject to} \quad H^T = H, \quad s_k = H y_k \tag{2.11}$$

with the weighted Frobenius norm just like in (2.8).

In order to obtain a descent search direction, we need H_k (and therefore B_k) to be positive definite matrix. This is possible only if the curvature condition is satisfied, i.e. if

$$s_k^T y_k > 0.$$

It can be shown that if $H_k > 0$ and the previous inequality holds, then the subsequent BFGS approximation H_{k+1} will also be positive definite. The same holds for B_k. Therefore, we can start with a positive definite initial approximation and use the updating formula only if the curvature condition holds. Else, we can skip the updating and put $H_{k+1} = H_k$. The initial approximation is often defined as $H_0 = I$. There are some other possibilities, of course, but this one is not so bad because BFGS approximation tends to correct itself in just a few iterations if the correct line search is applied. It is also considered as more successful in practice than DFP [46].

BFGS and DFP are rank-2 updating formulas, i.e. the difference $B_{k+1} - B_k$ is a rank-2 matrix. The method that represents rank-1 updating formulas is the SR1 (Symmetric-rank-1) method defined by

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \tag{2.12}$$

It produces a sequence of symmetric matrices that satisfy the secant equation. Unlike the previously stated BFGS updating, this method does not guaranty positive definiteness of the approximation matrix and therefore there is no guaranty for descent search direction of the form (2.6). Moreover, it does not have the superlinear convergence

result as BFGS has. However, SR1 approximation of the (inverse) Hessian is often better in practice than BFGS approximation and good numerical results made this method very popular [46].

Notice that the SR1 update is not well defined if the denominator $(s_k - H_k y_k)^T y_k$ is zero. This problem can be solved by leaving the approximation unchanged if, for example, the following holds

$$|(s_k - H_k y_k)^T y_k| < 10^{-8} \|s_k - H_k y_k\| \|y_k\|.$$

Such update provides a sequence of good approximations because the previous inequality usually does not happen very often. One more nice property of this method is that it finds the solution in at most n iterations when the objective function is strongly convex and the line search is appropriate. The main difference between SR1 and the other two methods is that the search direction obtained by the SR1 might be nondescent. Therefore, nonmonotone line search is more appropriate for this kind of search directions as we will see in a sequel.

In Chapter 6, we l present some numerical results that use slightly modified versions of the quasi-Newton search directions. The methods that are also considered here are the so called spectral gradient methods. They are constructed by Barzilai and Borwein [2] and therefore they are often referred as BB methods. In [2] the global convergence for convex quadratic objective function is considered without any line search. The spectral gradient idea is developed for several optimization problems by Birgin et al. (see [9] for example). Again, the central issue is the secant equation. More precisely, we want to find a diagonal matrix of the special form

$$D_k = \gamma_k I, \quad \gamma_k \in \mathbb{R}$$

that best fits the secant equation (2.9). This matrix will be considered as an approximation of the inverse Hessian $(\nabla^2 f(x_k))^{-1}$ and the search direction will be parallel to negative gradients direction, i.e.

$$p_k = -\gamma_k I \nabla f(x_k) = -\gamma_k \nabla f(x_k).$$

The quotient γ_k that contains the second order information is obtained as the solution of the problem

$$\min_{\gamma\in\mathbb{R}} \|\gamma y_{k-1} - s_{k-1}\|^2$$

where s_{k-1} and y_{k-1} are defined as above. This problem can be solved analytically and the solution is given by

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}. \tag{2.13}$$

The other possibility is to put

$$\gamma_k = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}. \tag{2.14}$$

This quotient is obtained by observing the secant equation (2.7) and solving the problem

$$\min_{\gamma\in\mathbb{R}} \|y_{k-1} - \gamma s_{k-1}\|^2.$$

Since the solution of the previous problem yields an approximation of the Hessian and not its inverse, the search direction is $p_k = -\gamma_k \nabla f(x_k)$ with γ_k given by (2.14). Either way, this is the method that incorporates the information of the second order while the computation is very easy. It was originally constructed to accelerate the steepest descent method that sometimes tend to converge in zigzag fashion if the Hessian is nearly singular at the solution (Forsythe [25]). Notice that if the curvature condition $s_{k-1}^T y_{k-1} > 0$ does not hold, then γ_k can be negative and the search direction is not the descent one. However, if we use the safeguard (Tavakoli, Zhang [63])

$$\bar{\gamma}_k = \min\{\gamma_{max}, \max\{\gamma_k, \gamma_{min}\}\}$$

and set $p_k = -\bar{\gamma}_k \nabla f(x_k)$, then we are sure that the direction is descent and numerical stability can be controlled. The parameters are proposed to be $0 < \gamma_{min} << 1 << \gamma_{max} < \infty$.

The spectral gradient method is proposed to be combined with nonmonotone line search which is not that strict on decreasing the function value. The main reason for this proposal is that the monotone line search may reduce the spectral gradient method to the ordinary steepest descent and destroy its good performance regarding the rate of convergence. For example, Raydan [53] used the nonmonotone technique from Grippo et al. [30] to obtain the globally convergent BB method. Using the safeguard rule which prohibits nondescent search directions, he proved the convergence for the class of continuously-differentiable objective functions with bounded level sets. Nonmonotone techniques are presented in the following section in more details.

Another approach for obtaining the search direction that is widely known is given by the conjugate gradient method that was originally constructed for solving systems of linear equations. However, it will not be the subject of our research and we will not make any more comments on it. For more details, see [46] for instance.

At all of the previously stated methods, an approximation of a gradient can be used. There are many examples where the true gradient is not known or it is very hard to calculate it. In that case, the function evaluations at different points are used in order to obtain the gradient approximation. For example, interpolation techniques can be applied where the focus is on the optimal choice of the interpolation points. On the other hand, there are finite difference methods. For example, if we use the centered finite difference estimator, then the ith component of the gradient approximation is

$$(\nabla f(x))_i \approx \frac{f(x + he_i) - f(x - he_i)}{2h}$$

where e_i represents the ith column of the identity matrix. If the ap-

proximation is used in a framework of an iterative method, then the parameter h can be substituted by the sequence of parameters $\{h_k\}_{k\in\mathbb{N}}$ which usually tends to zero. That way, more and more accurate approximation if obtained. For more insight in derivative-free optimization in general, one can see Conn et al. [16] for example.

2.2.2 Step size

After reviewing the methods for obtaining the search direction, we l describe the basics for finding the step length. Of course, the best thing would be if we solve (2.4) exactly and take the full advantage of direction p_k. However, this can be too hard and it can take too much time. More importantly, it is not necessary to solve this problem exactly, but it is desirable to find an approximate solution that decreases the value of the objective function. Therefore, we are searching for α_k such that $f(x_k + \alpha_k p_k) < f(x_k)$. However, requiring arbitrary small decrease is not enough. In order to ensure the convergence, we impose the sufficient decrease condition

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \eta\alpha_k(\nabla f(x_k))^T p_k, \tag{2.15}$$

where η is some constant that belongs to the interval $(0, 1)$, usually set to $\eta = 10^{-4}$. This condition is often called the Armijo condition.

In order to obtain a reasonable reduction in the objective function, we need to ensure that the step length is not too short. This can be done by imposing the curvature condition

$$(\nabla f(x_k + \alpha_k p_k))^T p_k \geq c\,(\nabla f(x_k))^T\, p_k \tag{2.16}$$

where c is some constant that satisfies $0 < \eta < c < 1$. This condition, together with (2.15) makes the Wolfe conditions. Let us define the function $\Phi(\alpha) = f(x_k + \alpha p_k)$. Previous condition is then equivalent to $\Phi'(\alpha_k) \geq \Phi'(0)$. This means that increasing the step size would

probably not be beneficial for decreasing the objective function value. However, there is no guaranty that α_k is the local minimum of function $\Phi(\alpha)$. If we want to obtain the step size that is at least in a broad neighborhood of the stationary point of function Φ, we can impose the strong Wolfe conditions. They consist of the Armijo condition (2.15) and

$$|(\nabla f(x_k + \alpha_k p_k))^T p_k| \leq c|(\nabla f(x_k))^T p_k|$$

instead of (2.16). Now, we will state the important result that gives the conditions for the existence of the step size that satisfies the (strong) Wolfe conditions.

Lemma 2.2.1 *[46] Suppose that the function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and let p_k be a descent direction for function f at point x_k. Also, suppose that f is bounded below on $\{x_k + \alpha p_k | \alpha > 0\}$. Then if $0 < \eta < c < 1$, there exist intervals of step lengths satisfying the (strong) Wolfe conditions.*

Another alternative for the Wolfe conditions are the Goldstein conditions given by

$$f(x_k)+(1-c)\alpha_k(\nabla f(x_k))^T p_k \leq f(x_k+\alpha_k p_k) \leq f(x_k)+c\alpha_k(\nabla f(x_k))^T p_k$$

where c is a positive constant smaller than 0.5. Bad side of imposing these conditions is that they may exclude all the minimizers of function Φ. Moreover, there are indications that the Goldstein conditions are not well suited for quasi-Newton methods [46].

Notice that if we want to check whether the (strong) Wolfe conditions are satisfied, we have to evaluate the gradient at every candidate point. There are many situations when evaluating the derivatives is much more expensive than evaluating the objective function. In that sense, less expensive would be the technique which is called backtracking. If our goal is to find a step size that satisfies the Armijo condition,

backtracking would start with some initial value α_k^0 and check if (2.15) holds for $\alpha_k = \alpha_k^0$. If it holds, we have found the suitable step size. If not, we decrease the initial step size by multiplying it with some constant $\beta \in (0, 1)$ and check the same condition with $\alpha_k = \beta\alpha_k^0$. If the Armijo condition still does not hold, we repeat the procedure of decreasing the step size until a suitable step length is found. If the conditions of Lemma 2.2.1 are satisfied, we will find the step length that satisfies (2.15) after a finite number of trials. Therefore, under the standard assumptions on the objective function f and the search direction p_k the backtracking technique is well defined.

Sometimes interpolation is used in order to enhance the backtracking approach. The idea is to use the points that we have obtained to approximate the function $\Phi(\alpha)$ with a polynomial function $\Phi_q(\alpha)$. This approximating function is usually quadratic or cubic and therefore easy to work with, i.e. we can find its exact minimum which is further used to approximate the solution of problem (2.4). For example, we can require the match of these two functions and their derivatives at $\alpha = 0$ as well as the match of the functions at the last one or two points that did not satisfy the Armijo condition.

Interpolation can also be used for obtaining the initial step size at every iteration. For example, we can use the data that we have already obtained, construct the quadratic function of α and set the initial guess α_k^0 to be the minimizer of that quadratic function. Another popular approach is to require the match between the first-order change at the current iteration and the previous one, i.e. to impose $\alpha_k^0(\nabla f(x_k))^T p_k = \alpha_{k-1}(\nabla f(x_{k-1}))^T p_{k-1}$. After obtaining the starting step size, we can continue with the standard backtracking. This interpolation approach for finding the starting point is suitable for the steepest descent method for instance. However, if the Newton-like method is used, we should always start with $\alpha_k^0 = 1$ because the full step $s_k = p_k$ has some nice properties such as positive influence on the convergence rate. See [46].

We will conclude this section by stating the famous Zoutendijk's result regarding the global convergence of the line search methods. It will be stated for the Wolfe conditions, but similar results can be obtained for the strong Wolfe or the Goldstein conditions. This theorem reveals the importance of the angle between the search direction p_k and the negative gradient direction. Let us denote this angle by θ_k, i.e. we define

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

Then, the following holds.

Theorem 2.2.1 *[46] 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 ∇f is Lipschitz continuous on $\mathcal{N}$ and that p_k is a descent search direction. Also, suppose that f is bounded below on $\mathbb{R}^n$ and that the step size α_k satisfies the Wolfe conditions. Then*

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

This result implies that $\lim_{k\to\infty} \cos^2\theta_k \|\nabla f(x_k)\|^2 = 0$ and therefore, if we have the sequence of search directions $\{p_k\}_{k\in\mathbb{N}}$ that are close enough to the negative gradient, or more precisely, if there exists a positive constant δ such that $\cos\theta_k \geq \delta$ for every k sufficiently large, then we have $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$. In other words, we obtain the global convergence. This is obviously true for the negative gradient search direction where $\cos\theta_k = 1$ for all k. On the other hand, if we consider the Newton-like method with $p_k = -H_k \nabla f(x_k)$ and $H_k > 0$, then $\cos\theta_k$ will be bounded from below if the conditional number of the matrix H_k is uniformly bounded from above. More precisely, if $\|H_k\|\|H_k^{-1}\| \leq M$ for some positive constant M and every k, then one

can show that $\cos\theta_k \geq \frac{1}{M}$ and obtain the global convergence result under the stated assumptions.

Until now, we have seen what are the main targets concerning the search directions and the step sizes. The Armijo condition has been playing the most important role in imposing the sufficient decrease in the objective function. In the following section, we will review the line search strategies that do not require that strong descent. Moreover, they even allow an increase in function value at some iterations in order to obtain faster convergence in practice and to increase the chances of finding the global solution.

2.3 Nonmonotone strategy

2.3.1 Descent search directions

There are various reasons that have lead researchers to introduce the nonmonotone strategies. The first line search technique for unconstrained optimization is proposed by Grippo et al. [30]. They considered Newton's method and notice that imposing the standard Armijo condition on the step size can severely slow down the convergence, especially if the objective function has narrow curved valleys. If the iterate of the algorithm comes in such kind of valley, it remains trapped and algorithm starts to crawl. This happens because, in that case, the Armijo rule is satisfied only for small step sizes and therefore the full step is not accepted. On the other hand, it is known that the full step is highly desirable when we use the quasi-Newton or Newton methods because it brings the potential for superlinear or even quadratic convergence. Therefore, line search rules that give more chances for full step to be accepted are developed. The same basic idea has been proposed earlier in Chamberlain et al. [11] but for the constrained optimization problems. The first attempt for unconstrained optimization

[30] has been to search for the step size that satisfies

$$f(x_k + \alpha_k p_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \eta \alpha_k (\nabla f(x_k))^T p_k \tag{2.17}$$

where $m(0) = 0, \quad 0 \leq m(k) \leq \min\{m(k-1)+1, M\}$ for $k \geq 1$, $\eta \in (0,1)$ and M is a nonnegative integer. In other words, we want to find the point where function value is sufficiently smaller than the maximum of the previous M (or less at the beginning) function values. This strategy can be viewed as a generalization of the standard Armijo rule. The important fact about this first nonmonotone line search is that it requires the descent search direction. More precisely, the authors assume that search directions satisfy the following conditions for some positive constants c_1 and c_2

$$(\nabla f(x_k))^T p_k \leq -c_1 \|\nabla f(x_k)\|^2 \tag{2.18}$$

$$\|p_k\| \leq c_2 \|\nabla f(x_k)\|. \tag{2.19}$$

Under the assumption of bounded level set and twice continuously-differentiable objective function, they have proved that every accumulation point of the algorithm is a stationary point of f. However, they modify the Newton step every time the Hessian is singular by using the negative gradient direction in that iterations. Moreover, they suggest that the standard Armijo rule should be used at the beginning of optimization process leaving the nonmonotone strategy for the remaining part. Their numerical study shows that using $M = 10$ provides some significant savings in number of function evaluations when compared to $M = 0$. It indicates the advantage of the nonmonotone rule over the standard Armijo since the number of function evaluations is often the important criterion for evaluating the algorithms.

In [31], Grippo et al. relax the line search even more. Roughly speaking, they allow some directions to be automatically accepted and they are checking whether the sufficient decrease is made only

occasionally. This modification improved the performance of their algorithm. This is confirmed by Toint [64] who tested two algorithms of Grippo et al. and compared them to the standard line search algorithm on the CUTE collection Bongratz et al. [10]. Toint also proposes the new modification of this nonmonotone strategy which made some savings in the number of function evaluations for some of the tested problems. The advantage of nonmonotone strategies is clear, especially for modified algorithm [31]. It is visible in the number of function evaluations as well as in CPU time since the modified algorithm performs the best in most of the tested problems. It is also addressed that using the monotone strategy at the beginning of the process appears to be beneficial.

In [17] Dai provides some basic analysis of the nonmonotone scheme (2.17). He proposes that one should try to put through the full step by applying the nonmonotone rule and if it does not work, standard Armijo should be applied. Moreover, if the objective function is not strongly nonlinear one should prefer monotone scheme. He considers the descent search direction and gives the necessary conditions for the global convergence. Under the standard assumptions such as Lipschitz continuity of the gradient and boundedness of the objective function from below, he proves that the sequence $\{\max_{1\leq i\leq M} f(x_{Mk+i})\}_{k\in\mathbb{N}}$ is strictly monotonically decreasing. Furthermore, under the additional assumptions (2.18) and (2.19) on the search directions, he proves that every accumulation point of the algorithm is stationary. A weaker result can be obtained by imposing a weaker assumption instead of (2.19). More precisely, if we assume the existence of some positive constants β and γ such that for every k

$$\|p_k\|^2 \leq \beta + \gamma k \tag{2.20}$$

we can obtain that $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$.

Dai also proves R-linear rate of convergence when the algorithm is applied on a continuously-differentiable objective function which is

uniformly convex. Under the assumptions (2.18) and (2.19), he proves the existence of the constants $c_4 > 0$ and $c_5 \in (0, 1)$ such that for every k

$$f(x_k) - f(x^*) \leq c_4 c_5^k (f(x_1) - f(x^*))$$

where x^* is a strict global minimizer.

It is noticed that the performance of the algorithms based on the line search rule (2.17) is very dependent on the choice of M which is considered as one of the drawbacks. Another nonmonotone line search method is proposed in Zhang, Hager [71] where it is pointed out that the likelihood of finding global optimum is increased by using nonmonotone rules. The authors conclude that their method provides some savings in number of function and gradient evaluations compared to monotone technique. Moreover, the savings are noted compared to the nonmonotone line search (2.17) as well. Instead of using the maximum of previous function values, it is suggested in [71] that a convex combination of previously computed function values should be used. The algorithm is constructed to find a step size α_k that satisfies the condition

$$f(x_k + \alpha_k p_k) \leq C_k + \eta \alpha_k (\nabla f(x_k))^T p_k, \tag{2.21}$$

where C_k is defined recursively. More precisely, $C_0 = f(x_0)$ and

$$C_{k+1} = \frac{\eta_k Q_k}{Q_{k+1}} C_k + \frac{1}{Q_{k+1}} f(x_{k+1}), \tag{2.22}$$

where $Q_0 = 1$ and

$$Q_{k+1} = \eta_k Q_k + 1 \tag{2.23}$$

with $\eta_k \in [\eta_{min}, \eta_{max}]$ and $0 \leq \eta_{min} \leq \eta_{max} \leq 1$. Parameter η_k determines the level of monotonicity. If we put $\eta_k = 1$ for every k, algorithm treats all previous function values equally, i.e.

$$C_k = \frac{1}{k+1} \sum_{i=0}^{k} f(x_i), \tag{2.24}$$

while $\eta_k = 0$ yields standard Armijo rule. The authors say that the best numerical results are obtained if we let η_k be close to 1 far from the solution and closer to 0 when we achieve a neighborhood of the minimizer. However, they report only the results for $\eta_k = 0.85$ since it provides satisfactory performance. The convergence analysis is conducted for the descent search directions and it is shown that $C_k \geq f(x_k)$ which makes the line search rule well defined. Under the similar assumptions as in [17], the global convergence is proved. However, the result depends on η_{max}. In general, $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$ but if $\eta_{max} < 1$ then the stronger result holds, i.e. every accumulation point is stationary for function f. Also, the R-linear convergence for strongly convex functions is obtained.

2.3.2 General search directions

Notice that all the above stated line search rules require descent search directions in order to be well defined. However, this requirement is not always satisfied. There are many applications where derivatives of the objective function are not available. Moreover, there are also quasi-Newton methods that do not guaranty descent search directions. These methods have very nice local properties but making them globally convergent has been a challenge. In order to overcome this difficulty and to obtain globally and superlinearly convergent algorithm, Li and Fukushima [41] introduced a new line search. They consider the problem of solving the system of nonlinear equations $F(x) = 0$, $F : \mathbb{R}^n \to \mathbb{R}^n$ which is equivalent to the problem of minimizing $f(x) = \|F(x)\|$ or $f(x) = \|F(x)\|^2$. The line search rule is of the form

$$\|F(x_k + \alpha_k p_k)\| \leq \|F(x_k)\| - \sigma_1 \|\alpha_k p_k\|^2 + \varepsilon_k \|F(x_k)\| \qquad (2.25)$$

where σ_1 is some positive constant and $\{\varepsilon_k\}_{k\in\mathbb{N}}$ is a sequence of positive numbers which satisfies the following condition

$$\sum_{k=0}^{\infty} \varepsilon_k < \infty. \tag{2.26}$$

Notice that under the standard assumptions about F, positivity of ε_k yields acceptance of any kind of direction providing that the step size is small enough. This kind of sequence is used, for example in Birgin et al. [7] where inexact quasi-Newton methods are considered and therefore the search direction is not descent in general.

Probably the earliest derivative-free line search was introduced by Griewank in [29], but some difficulties were discovered concerning the line search. Therefore, the (2.25) is considered as one of the first well defined derivative-free line search rules (Cheng, Li [12]). The work that combines the ideas from [30], [41] and Lucidi, Sciandrone [42] is presented by Diniz-Ehrhardt et al. in [23]. The idea was to construct a method that accepts nondescent search directions, tolerates the nonmonotone behavior and explores several search directions simultaneously. More precisely, before reducing the step size, all directions from a finite set are checked for satisfying the following line search rule

$$f(x_k + \alpha p_k) \leq \max\{f(x_k), \ldots, f(x_{\max\{k-M+1,0\}})\} + \varepsilon_k - \alpha^2\beta_k \tag{2.27}$$

where β_k belongs to the sequence that satisfies the following assumption.

P 1 *$\{\beta_k\}_{k\in\mathbb{N}}$ is a bounded sequence of positive numbers with the property*

$$\lim_{k\in K} \beta_k = 0 \Rightarrow \lim_{k\in K} \nabla f(x_k) = 0,$$

for every infinite subset of indices $K \subseteq \mathbb{N}$.

What are the choices for this sequence? For example, it can be defined as $\beta_k = \min\{\delta, \|\nabla f(x_k)\|^\tau\}$ where δ and τ are some positive constants. Actually, the choice $\beta_k = \delta$ is also valid. Moreover, some approximation of the gradient's norm which tends to be exact can also be used. For instance, we can use $\beta_k = \| \left(g_1^k, \ldots, g_n^k\right)^T \|$ where

$$g_i^k = \frac{f(x_k + h_k e^i) - f(x_k - h_k e^i)}{2h_k}$$

if we ensure that $\lim_{k\to\infty} h_k = 0$.

In order to demonstrate the power of the line search (2.27), the authors even used random search directions. Their numerical results suggest that some amount of random search directions can be beneficial, especially for the large dimension problems. Namely, increasing the percentage of random directions yielded the increase of number of iterations but also of the number of successfully solved problems. The sequence (2.26) was defined as $\varepsilon_k = |f(x_0)|/k^{1.1}$. They also consider the algorithm that uses the SR1 directions which can be nondescent. Convergence analysis rely exclusively on the line search technique. First, assuming that the objective function is bounded from below, it is proved that there exists a subsequence of iterations K such that $\lim_{k\in K} \alpha_k^2 \beta_k = 0$. Furthermore, if (x^*, p^*) is a limit point of $\{(x_k, p_k)\}_{k\in K}$ then it satisfies the inequality $(\nabla f(x^*))^T p^* \geq 0$. However, for proving the existence of an accumulation point which is stationary for f, descent search directions are required.

Line search (2.21) has also been modified in order to accept nondescent directions which makes it applicable to derivative-free optimization problems. The modified line search proposed in [12] is of the form

$$f(x_k + \alpha_k p_k) \leq C_k + \epsilon_k - \gamma \alpha_k^2 f(x_k) \tag{2.28}$$

where $\gamma \in (0, 1)$ and C_k is defined as in [71] with the slight modification

concerning ϵ_k

$$C_{k+1} = \frac{\eta_k Q_k}{Q_{k+1}}(C_k + \epsilon_k) + \frac{1}{Q_{k+1}} f(x_{k+1}). \tag{2.29}$$

Here, $f(x_k)$ plays the role of β_k from (2.27) since the problem under consideration is solving the system of nonlinear equations $F(x) = 0$ and the objective function is defined as $f(x) = \frac{1}{2}\|F(x)\|^2$. This line search is combined with the spectral residual method proposed by La Cruz et al. [40] and it yielded promising results which support the idea that spectral methods should be combined with nonmonotone line search. Convergence results again distinguish the case where η_{max} is smaller than 1. In that case, it is proved that every limit point x^* of sequence of iterates satisfies $(F(x^*))^T J(x^*)F(x^*) = 0$ where $J(x)$ is the Jacobian of $F(x)$.

Chapter 3

Stochastic optimization

3.1 Stochastic in optimization

Let us begin this section by distinguishing two main types of stochastic optimization. The key difference between them can be expressed through the role of noise in optimization. In the first case, the noise is inevitable. It appears as a random variable in the objective function or within constraints. This can be a consequence of uncertainty or errors in measuring output or input data. For example, uncertainty is present if we are observing a system whose performance is going to be known in future, but it depends on many factors that can not be considered while making the model. Therefore, they are considered as random variables. The measuring errors can also be considered as random variables. For instance, if we are trying to optimize the temperature of some gas, objective function - the temperature is obtained only with finite number of decimals and rounding errors appear. They can be considered as noise in measuring.

These were examples where randomness is present whether we like it or not. On the other hand, stochastic optimization can represent

the algorithms where some noise is intentionally introduced. This is usually done by using random points or random search directions like in direct search methods [62]. They are especially convenient when there is lack of information about the derivatives and the objective function itself. Even if that is not the case, random directions are used to explore the regions where standard directions would not enter. This can speedup the convergence or increase the likelihood of finding global optimizer. Moreover, random vectors are used in approximations of derivatives within simultaneous perturbation techniques that will be described latter. The main idea is to decrease the number of function evaluations when the dimension of the problem is large.

Finally, let us point out that noisy data does not have to mean that the problem that we are observing contains explicit noise. The typical example is a well known problem of finding maximum likelihood estimators. This is the problem of parameter estimation. Suppose that the type of distribution of a random variable is known, but it is not fully determined because we do not know the values of parameters of that distribution. For example, suppose that the variable is normally distributed, but the mean and the variance are unknown. Furthermore, suppose that we have some realization of a random sample from that distribution which is i.i.d. and let us denote that realization by $\xi_1, \ldots, \xi_N$. Then, we are searching for the parameters that maximize the likelihood of that particular sample realization. If we denote the relevant probability distribution function by $f_p(x)$ the problem becomes $\max_x \prod_{i=1}^{N} f_p(\xi_i)$ or equivalently

$$\min_x - \sum_{i=1}^{N} \ln f_p(\xi_i) \tag{3.1}$$

where x represents the vector of parameters to be estimated. For example, if the underlying distribution is $\mathcal{N}(\mu, \sigma^2)$, then the problem is $\min_{\mu,\sigma} - \sum_{i=1}^{N} \ln(\frac{1}{\sqrt{(2\pi\sigma^2)}} e^{-\frac{(\xi_i-\mu)^2}{2\sigma^2}})$. The key issue here is that formally,

the objective function is random since it depends on random sample. However, once the sample realization is known, we can consider the objective function as deterministic. Another important example of "vanishing noise" is regression problem. Although the data represents random variables, we obtain only the input-output pairs $(a_i, y_i), i = 1, \ldots, N$ and the problem is in the least squares form

$$\min_x \sum_{i=1}^{N} (g(x, a_i) - y_i)^2, \tag{3.2}$$

where g is a model function. Specifically, in the case of linear regression it is in the form of $g(x, a) = a^T x$. In general, many data fitting problems are in a form of least squares

$$\min_x \sum_{i=1}^{N} f_i^2(x). \tag{3.3}$$

For further references one can consult Friedlander, Schmidt [26] for instance.

We will consider the stochastic optimization problems with underlying randomness rather than the stochastic algorithms with intentionally imposed noise. The considered problems have various forms. First, we can consider objective function which is random, i.e. $\min_x F(x, \xi)$ where ξ represents the noise and x is the decision variable. For example, this problem can appear when we have to make the decision now but the full information about the problem parameters is going to be known at some future moment. The lack of information in that case is introduced by the random vector ξ. In the investment world ξ may represent the return of different financial assets and the decision variable can represent the portion of wealth to be invested in each one of those assets. If, for example, the distribution of ξ is discrete and known, one can make different problems by

observing different scenarios and maybe try to solve the one which is most probable. However, this approach can result in great amount of risk. The most common way of dealing with these kind of problems is to introduce the mathematical expectation, i.e. to try to minimize the mean of the random function

$$\min_x E(F(x,\xi)). \tag{3.4}$$

Although this way the noise is technically removed, these kind of problems are hard to solve. Even if the distribution of the random vector ξ is known the expectation can be hard to evaluate, i.e. to find its analytical form.

Of course, all the problems that we mentioned so far can be a part of constrained or unconstrained optimization. Moreover, the constraints can also be defined by random variables. For example, if $F(x,\xi)$ represents the loss of a portfolio then one can try to minimize the loss but at the same time to make sure that the loss will not exceed certain value. Furthermore, one can choose to put the constraint $D(x,\xi) \leq c$ which means that the variance of the loss should be bounded with some constant c or in other words, the risk should be controlled. Notice that these kind of problems can also be considered in a form of unconstrained mathematical expectation optimization by using the merit function approach.

Now, suppose that we want to minimize the function $E(F(x,\xi))$ so that $G(x,\xi) \leq 0$ is satisfied for (almost) every ξ. This can easily become unfeasible, i.e. we can obtain the problem with empty feasible set. These constraints can be relaxed if we choose to satisfy $G(x,\xi) \leq 0$ with some high probability, but smaller than 1. In that case, we obtain the problem

$$\min E(F(x,\xi)) \quad \text{subject to} \quad P(G(x,\xi) \leq 0) \geq 1-\alpha,$$

where α is some small positive number, very often set to 0.05. The problems with this kind of constraints are called chance constrained

problems. Notice that $P(G(x,\xi) \leq 0) = E(I_{(-\infty,0)}(G(x,\xi)))$ where I stands for the indicator function, so the constraint can be approximated as any other expectation function - by sample average for instance. However, this function is usually discontinuous and therefore hard to work with. Another way of approaching this problem is

$$\min E(F(x,\xi)) \quad \text{subject to} \quad G(x,\xi_i) \leq 0,\ i = 1,\dots,N$$

where $\xi_1,\dots,\xi_N$ is some generated sample. The question is how large should N be so that an optimal solution of the previously stated problem satisfies originally stated chance constraint. This is another tough problem. However, some fairly sharp bounds are developed in the case of convex problems - where the objective function and the function G are convex with respect to the decision variable x. For further reference on this topic one can consult Shapiro [59].

In the next few sections we will mainly consider problems of the form (3.4) in the unconstrained optimization framework. These problems have been the subject of many research efforts and two main approaches have been developed. The first one is called Stochastic Approximation (SA) and its main advantage is solid convergence theory. It deals directly with the noisy data and adopts the steepest descent or Newton-like methods in stochastic framework. On the other hand, Sample Average Approximation (SAA) method transforms the problem into deterministic one which allows the application of the deterministic tools. However, the sample usually has to be very large which can be very expensive if the function evaluations are the main cost at every iteration.

3.2 Stochastic approximation methods

Let us begin by referring to "No free lunch theorems" established in Wolpert, Macready [68]. Basically the theorems state that no algo-

rithm is universally the best one. One algorithm can suite "perfectly" to one class of problems, while it performs poor on some other classes of problems. Moreover, there are some algorithms that are very successful in practice although there is no underlying convergence theory. This is not the case with Stochastic Approximation (SA) method. There is strongly developed convergence theory. Usually, the almost sure convergence is achievable. However, convergence assumptions are sometimes hard to verify or satisfy. The good thing about SA is even if the convergence assumptions are not verified, it can perform well in practise.

We will start by considering the SA algorithm for solving the systems of nonlinear equations. This approach is strongly related to unconstrained optimization problems. Since we are usually satisfied if we find a stationary point of the objective function f, the optimization problem can be viewed as the problem of solving $\nabla f(x) = 0$. SA algorithm is often referred to as the Robbins-Monro algorithm if the information about the derivatives is available. It is applicable on constrained problems, but we consider only the unconstrained case. The only difference is in adding the projection function which is applied on iterates in order to maintain the feasibility. Moreover, the convergence theory can be conducted throughout differential equations but we will not consider this kind of approach (see [62] for further references).

Consider the system of nonlinear equations

$$g(x) = 0,\ g : \mathbb{R}^n \to \mathbb{R}^n. \tag{3.5}$$

Suppose that we are able to obtain only the measurements with noise that depends on iteration as well as on decision variable

$$\hat{g}_k(x) = g(x) + \xi_k(x). \tag{3.6}$$

Then the SA is defined by

$$\hat{x}_{k+1} = \hat{x}_k - a_k \hat{g}_k(\hat{x}_k). \tag{3.7}$$

The iterates are denoted by $\hat{x}_k$ instead of x_k in order to emphasize their randomness which is the consequence of using random samples throughout the iteration process. The sequence of step sizes $\{a_k\}_{k\in\mathbb{N}}$ is also called the gain sequence and its influence on the convergence is huge. Therefore, the first assumption is the following.

S 1 *The gain sequence satisfies:* $a_k > 0$, $\lim_{k\to\infty} a_k = 0$, $\sum_{k=0}^{\infty} a_k = \infty$ *and* $\sum_{k=0}^{\infty} a_k^2 < \infty$.

The assumption that step sizes converge to zero is standard in stochastic algorithms [62]. The condition $\sum_{k=0}^{\infty} a_k = \infty$ is imposed in order to avoid inefficiently small step sizes. On the other hand, we do not want to have unstable behavior and that is why the summability condition on a_k^2 is here. Its role is to decrease the influence of the noise when the iterates come into a region around the solution. The example of a sequence that satisfies the previous assumption is

$$a_k = \frac{a}{(k+1)^{\alpha}} \tag{3.8}$$

where $\alpha \in (0.5, 1]$ and a is some positive constant. Assumptions S1 - S4 are applicable only when x^* is the unique solution of the considered system.

S 2 *For some symmetric, positive definite matrix* B *and for every* $\eta \in (0, 1)$,

$$\inf_{\eta<\|x-x^*\|<\frac{1}{\eta}} (x - x^*)^T B g(x) > 0.$$

S 3 *For all* x *and* k, $E(\xi_k(x)) = 0$.

The condition of zero mean is also standard in stochastic optimization. Its implication is that $\hat{g}_k(x)$ is unbiased estimator of $g(x)$.

S 4 *There exist constant $c > 0$ such that for all x and k,*

$$\|g(x)\|^2 + E(\|\xi_k(x)\|^2) \leq c(1 + \|x\|^2).$$

Notice that under the assumption S3, the previous condition is equal to $E(\|\hat{g}_k(x)\|^2) \leq c(1 + \|x\|^2)$ because

$$\begin{aligned} E(\|\hat{g}_k(x)\|^2) &= E(\|g(x)\|^2 + 2(g(x))^T\xi_k(x) + \|\xi_k(x)\|^2) \\ &= E(\|g(x)\|^2) + 2g(x)^T E(\xi_k(x)) + E(\|\xi_k(x)\|^2) \\ &= \|g(x)\|^2 + E(\|\xi_k(x)\|^2) \end{aligned}$$

Therefore, the mean of $\|\hat{g}_k(x)\|^2$ can not grow faster than a quadratic function of x. Under these assumptions, we can establish almost sure convergence of the SA algorithm.

Theorem 3.2.1 *[62] Consider the SA algorithm defined by (3.7). Suppose that the assumptions S1 - S4 hold and that x^* is a unique solution of the system (3.5). Then $\hat{x}_k$ converges almost surely to x^* as k tends to infinity.*

Recall that the gain sequence is mentioned as the key player in this algorithm. It has impact on stability as well as on convergence rate. Therefore, it is very important to estimate the best choice for step sizes. The result that helps is the asymptotic normality of $\hat{x}_k$. Under some regularity conditions (Fabian [21]), it can be shown that

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

where $\to^d$ denotes the convergence in distribution, α refers to (3.8) and Σ is some covariance matrix that depends on the gain sequence and on the Jacobian of g. Therefore, for large k the iterate $\hat{x}_k$ approximately has the normal distribution $\mathcal{N}(x^*, k^{-\frac{\alpha}{2}}\Sigma)$. Because of the assumption S1, the maximal convergence rate is obtained for $\alpha = 1$. However,

this reasoning is based on asymptotic result. Since the algorithms are finite in practice, it is often desirable to set $\alpha < 1$ because $\alpha = 1$ yields smaller steps. Moreover, if we want to minimize $\|\Sigma\|$, the ideal sequence would be $a_k = \frac{1}{k+1} J(x^*)^{-1}$ where $J(x)$ denotes the Jacobian matrix of g (Benveniste et al. [6]). Even though this result is purely theoretical, sometimes the Jacobian at x^* can be approximated by $J(\hat{x}_k)$ and that way we can enhance the rate of convergence.

If we look at (3.8), we see that large constant a may speedup the convergence by making larger steps, but it can have negative influence on the stability. One way to improve the stability is to put the so called stability constant $A > 0$ and obtain $a_k = a/(k + 1 + A)^\alpha$. Another way of maintaining the stability when the dimension of x is 1 is to use the idea from Kesten [36] and to decrease the step size when $\hat{x}_{k+1} - \hat{x}_k$ starts to change the sign frequently. This is the signal that we are in the domain of noise, i.e. we are probably close to the solution and therefore we need small steeps to avoid oscillations. The idea from [36] is generalized in Delyon, Juditsky [18] to fit the larger dimensions. Furthermore, the way of dealing with oscillations is to consider the sequence of averaged iterates $\frac{1}{k+1} \sum_{i=0}^{k} \hat{x}_i$. However, this is recommended only if the noise is strong. If the sequence of $\hat{x}_k$ already converges more or less monotonically towards the solution, then the averaging can only slow it down.

The important choice for gain sequence is a constant sequence. Although this sequence does not satisfy the assumption S1, it can be shown that constant step size can conduct us to a region that contains the solution. This result initiated development of a cascading steplength SA scheme by Nedic et al. [20] where the fixed step size is used until some neighborhood of the solution is reached. After that, in order to come closer to the solution, the step size is decreased and again the fixed step size is used until the ring around the solution is sufficiently tighten up. That way, the sequence of iterates is guided towards the solution.

Since our main concern is the problem of the form

$$\min_{x\in\mathbb{R}^n} f(x) = E(F(x,\xi)),$$

we will consider the special case of the SA algorithm which is referred to as the SA for stochastic gradient. We have already commented that the previous problem can be viewed as a problem of solving the system of nonlinear equations $g(x) = 0$ where

$$g(x) = \nabla f(x) = \nabla E(F(x,\xi)).$$

Recall that the assumption S3 in fact says that $\hat{g}(x)$ has to be unbiased estimator of $g(x)$. Therefore, we are interested in the case where $\frac{\partial}{\partial x}F(x,\xi)$ can be used to approximate the gradient $g(x)$. In other words, it is important to know when

$$\frac{\partial}{\partial x}E(F(x,\xi)) = E(\frac{\partial}{\partial x}F(x,\xi)). \tag{3.9}$$

We will state the relevant theorems at the end of this section. Now, suppose that (3.9) is true and that $\frac{\partial}{\partial x}F(x,\xi)$ is known. Then there are at least two options for an unbiased estimator $\hat{g}_k(\hat{x}_k)$. The first one is called the instantaneous gradient and it uses

$$\hat{g}_k(\hat{x}_k) = \frac{\partial}{\partial x}F(\hat{x}_k,\xi_k)$$

where ξ_k is a realization of the random variable ξ. The other basic form uses

$$\hat{g}_k(\hat{x}_k) = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial x}F(\hat{x}_k,\xi_i)$$

where $\xi_1,\ldots,\xi_N$ is a fixed sample realization that is used throughout the whole optimization process. This approach is highly related to

sample path methods that are going to be considered later. Of course, there are other approaches that may combine these two extremes. For further reference see [62] for instance.

The Robbins-Monro algorithm can only achieve the convergence rate of $k^{-\frac{1}{2}}$. In general, if the objective function has more than one optimum then the SA converges only to a local solution (Fu [27], Andradottir [1]). Therefore, in some applications random search directions are used to enhance the chances of finding the global optimum [62].

Now, we state the conditions which imply the equality (3.9).

Theorem 3.2.2 *[62] Suppose that Ω is the domain of the random vector ξ which has the probability density function $\varphi_\xi(\tilde{\xi})$ and that $F(x,\tilde{\xi})\varphi_\xi(\tilde{\xi})$ and $\frac{\partial}{\partial x}F(x,\tilde{\xi})\varphi_\xi(\tilde{\xi})$ are continuous on $\mathbb{R}^n \times \Omega$. Furthermore, suppose that there exist nonnegative functions $q_0(\tilde{\xi})$ and $q_1(\tilde{\xi})$ such that*

$$|F(x,\tilde{\xi})\varphi_\xi(\tilde{\xi})| \leq q_0(\tilde{\xi})$$

and

$$\|\frac{\partial}{\partial x}F(x,\tilde{\xi})\varphi_\xi(\tilde{\xi})\| \leq q_1(\tilde{\xi})$$

for all $(x,\tilde{\xi}) \in \mathbb{R}^n \times \Omega$ and $\int_\Omega q_i(\tilde{\xi})d\tilde{\xi} < \infty$ for $i = 0, 1$. Then

$$\frac{\partial}{\partial x}\int_\Omega F(x,\tilde{\xi})\varphi_\xi(\tilde{\xi})d\tilde{\xi} = \int_\Omega \frac{\partial}{\partial x}F(x,\tilde{\xi})\varphi_\xi(\tilde{\xi})d\tilde{\xi},$$

i.e. (3.9) holds.

Notice that if the function F is continuously-differentiable with respect to x, the functions F and φ_ξ are continuous with respect to $\tilde{\xi}$ and the function and its gradient are bounded, i.e. there exist positive

constants M_0 and M_1 such that $|F(x,\tilde{\xi})| \leq M_0$ and $\|\frac{\partial}{\partial x}F(x,\tilde{\xi})\| \leq M_1$ for all $\tilde{\xi}$, then the result holds with $q_i = M_i\varphi_\xi(\tilde{\xi})$, $i = 0, 1$ because

$$\int_\Omega M_i\varphi_\xi(\tilde{\xi})d\tilde{\xi} = M_i \int_\Omega \varphi_\xi(\tilde{\xi})d\tilde{\xi} = M_i.$$

Now we state another set of conditions from [59].

Theorem 3.2.3 *[59] Suppose that $F(\cdot,\xi)$ is differentiable at $\bar{x}$ for almost every ξ and the expectation $E(F(\bar{x},\xi))$ is well defined and finite valued. Furthermore, suppose that there exists a positive valued random variable $C(\xi)$ such that $E(C(\xi)) < \infty$ and for all x, y in a neighborhood of $\bar{x}$ and almost every ξ the following inequality holds*

$$|F(x,\xi) - F(y,\xi)| \leq C(\xi)\|x - y\|.$$

Then

$$\frac{\partial}{\partial x}E(F(\bar{x},\xi)) = E(\frac{\partial}{\partial x}F(\bar{x},\xi)).$$

In order to make this analysis complete, we state the conditions for well defined expectation function.

Theorem 3.2.4 *[59] Suppose that $F(\cdot,\xi)$ is continuous at $\bar{x}$ for almost every ξ and there exists function $Z(\xi)$ such that $|F(x,\xi)| \leq Z(\xi)$ for almost every ξ and all x in a neighborhood of $\bar{x}$. Furthermore, assume that there exists $E(Z(\xi))$ and it is finite. Then the expectation $E(F(x,\xi))$ is well defined for all x in a neighborhood of $\bar{x}$. Moreover, $f(x) = E(F(x,\xi))$ is continuous at $\bar{x}$.*

If we take a look at this set of conditions, then we can conclude that (3.9) holds if for example the function $F(\cdot,\xi)$ is continuously-differentiable and bounded. Suppose that $|F(x,\xi)| \leq M$ for every ξ where M is some positive constant. Then $Z(\xi)$ from Theorem 3.2.4 can

be identified with M since $E(M) = M$. Therefore, $f(x) = E(F(x, \xi))$ is well defined and also finite valued because

$$|f(x)| \leq E(|F(x, \xi)|) \leq E(M) = M.$$

Moreover, the differentiability of $F(\cdot, \xi)$ implies the Lipschitz-continuity, i.e there exists some positive constant L such that

$$|F(x, \xi) - F(y, \xi)| \leq L\|x - y\|$$

for all x, y. Therefore, L can be identified with $C(\xi)$ from Theorem 3.2.3 and (3.9) holds for all x.

Previous results assume that the random vector ξ has probability density function and therefore the expectation is defined in the integral form. If the noise has discrete distribution, then the existence of both expectations in (3.9) implies the equality between them if the expectations are finite valued.

3.3 Derivative-free stochastic approximation

One of the assumptions in the previous section is that the information on the gradient $\nabla_x F(x, \xi)$ is available. Even if the gradient came with the noise, we could use it to construct the search direction. However, this assumption is not too realistic because, often, the gradient is unattainable. This is the case, for example, when we are dealing with a so called "black box" mechanisms. In that case, we only have input-output information. In other words, we can only obtain the value of the objective function without knowing its analytical form. Moreover, there are examples where even if the gradient is known, it is very expensive to evaluate it. In order to overcome this difficulties,

the methods that approximate the derivatives using only the objective function evaluations are developed. The algorithms that use that methods are called derivative-free algorithms. Specially, if they are considered in the SA framework, they are referred to as the Kiefer-Wolfowitz type algorithms.

We start the review of derivative-free algorithms with the Finite Difference SA method (FDSA). This method uses the SA iterative rule (3.7) combined with the finite difference approximation of the gradient. Two variants of finite differences are the most common. One of them is central (two-sided symmetric) difference gradient estimator whose ith component is defined as

$$(\hat{g}_k)_i(\hat{x}_k) = \frac{\hat{f}(\hat{x}_k + c_k e_i) - \hat{f}(\hat{x}_k - c_k e_i)}{2c_k}, \tag{3.10}$$

where e_i is the ith column of the identity matrix. The sequence of positive numbers $\{c_k\}_{k\in\mathbb{N}}$ is playing an important role in the convergence theory. Before stating the needed assumptions, we define the alternative which is called one-sided finite difference gradient estimator.

$$(\hat{g}_k)_i(\hat{x}_k) = \frac{\hat{f}(\hat{x}_k + c_k e_i) - \hat{f}(\hat{x}_k)}{c_k}. \tag{3.11}$$

Notice that the first estimator (FDC) uses $2n$ evaluations of function, while the later one (FDF) uses $n + 1$ evaluations, where n is the dimension of the decision variable. However, FDC often provides better approximations. When it comes to c_k, it is intuitive to think that the small values of that parameter would provide better approximations. Indeed, smaller parameter c_k yields the smaller bias. On the other hand, small c_k can have very bad influence on the variance. Since the sequence $\{a_k\}_{k\in\mathbb{N}}$ also controls the influence of noise in some way, the following assumption is stated.

S 5 *The* $\{a_k\}_{k\in\mathbb{N}}$ *and* $\{c_k\}_{k\in\mathbb{N}}$ *are sequences of positive numbers that converge to zero and satisfy the following conditions*

$$\sum_{k=0}^{\infty} a_k = \infty, \quad \sum_{k=0}^{\infty} a_k c_k < \infty, \quad \sum_{k=0}^{\infty} a_k^2 c_k^{-2} < \infty.$$

Notice that c_k should tend to zero, but slower that a_k. The choice for the sequence c_k can be of the form $c_k = c/(k+1)^\gamma$ with $c, \gamma > 0$. Of course, γ is chosen in a way to satisfy the needed assumption.

The second condition is on the form of the objective function.

S 6 *There is a unique minimizer* x^* *such that for every* $\eta > 0$,

$$\inf_{\|x-x^*\|>\eta} \|g(x)\| > 0 \text{ and } \inf_{\|x-x^*\|>\eta} (f(x) - f(x^*)) > 0.$$

For the next assumption we need to define the objective function in the following form

$$\hat{f}_k(x) = f(x) + \varepsilon_k(x) \tag{3.12}$$

where ε_k represents the noise. Moreover, define $\mathfrak{I}_k = \{\hat{x}_0, \ldots, \hat{x}_k\}$. This means that $\mathfrak{I}_k$ contains the information about the history of the algorithm until the iteration k.

S 7 *For all* i *and* k, $E(\varepsilon_k(\hat{x}_k + c_k e_i) - \varepsilon_k(\hat{x}_k - c_k e_i)|\mathfrak{I}_k) = 0$ *almost surely and* $E((\varepsilon_k(\hat{x}_k \pm c_k e_i))^2|\mathfrak{I}_k) \leq C$ *almost surely for some* $C > 0$ *that is independent of* k *and* x.

Finally, although the derivatives are not known, we suppose that they do exist and we state the following assumption.

S 8 *The Hessian matrix* $\nabla^2 f(x)$ *exists for all* x *and it is uniformly bounded.*

Now we can state the convergence result.

Theorem 3.3.1 *[62] Consider the SA algorithm defined by (3.7) and (3.10). Suppose that the assumptions S5 - S8 hold. Then $\hat{x}_k$ converges almost surely to x^* as k tends to infinity.*

Asymptotic normality of the iterates (under some additional conditions [56], [21]) is also attainable for FDSA, i.e. we have

$$k^{\frac{\beta}{2}}(\hat{x}_k - x^*) \rightarrow^d \mathcal{N}(\mu_{FD}, \Sigma_{FD}),\ k \rightarrow \infty \tag{3.13}$$

where $\beta = \alpha - 2\gamma$. Unlike in the Robbins-Monro type, FDSA in general does not have $\mu_{FD} = 0$. This is the consequence of having the bias in gradient estimation. While in the derivative-based SA we can obtain unbiased estimators, the finite difference can provide only asymptotically unbiased estimator. Furthermore, the convergence assumptions imply that the best asymptotic convergence rate is $k^{-\frac{1}{3}}$. However, this rate can be improved in some special cases when common random numbers (CRN) are used. The CRN concept can be viewed as a tool for reducing the variance of the estimators, but it will be described latter in more details (see page 82). However, it was suggested that the CRN concept is not that effective in practise [27].

Selection of a good gain sequence is a difficult task. Asymptotic results can be useless sometimes because the optimization processes are finite in practice. Although there are some semiautomatic methods [62], it is not a rare situation where the sequences are being tuned by using trial and error technique. Badly tuned parameters can result in very bad performance of the algorithm. Moreover, finite differences can yield poor approximations if the noise is strong [62].

One of the flaws of FD is its cost. If the dimension of the problem is large, evaluating the objective function in $2n$ points can be inefficient. To overcome this difficulty, methods that use only few evaluations per iteration regardless of the dimension are developed. We will present

the method of this kind which is called the Simultaneous Perturbations (SP) method (Fu [27], Spall [62]). It usually takes only two function evaluations, but there are even cases where only one evaluation is needed. Asymptotically, SPSA methods provide similar results as FDSA, but they are more efficient if the dimension of the problem is larger than 2. However, it is suggested in [27] that one should probably prefer FD if the function evaluations are not too expensive.

The idea behind SP is to perturb all the components at the same time by using one vector which is random in general. Denote this vector by $\Delta_k = (\Delta_{k,1}, ..., \Delta_{k,n})^T$. Then, the approximation of the gradient needed for SA iteration (3.7) is obtain by

$$(\hat{g}_k)_i(\hat{x}_k) = \frac{\hat{f}(\hat{x}_k + c_k\Delta_k) - \hat{f}(\hat{x}_k - c_k\Delta_k)}{2c_k\Delta_{k,i}}. \tag{3.14}$$

In order to obtain almost sure convergence, it is assumed that random vectors Δ_k, $k = 0, 1, 2 \ldots$ are i.i.d. and the components of that vector are independent random variables with mean zero and finite inverse second moment. This means that $E(\Delta_{k,i}) = 0$ and there exists the constant C such that $E((\Delta_{k,i})^{-2}) \leq C$. Moreover, it is usually assumed that the distribution is symmetric around zero. For example, a valid distribution for $\Delta_{k,i}$ is the symmetric Bernoulli. In that case, $\Delta_{k,i}$ can take only values 1 and -1, both with probability 0.5. The uniform distribution is not valid and the same is true for the normal distribution. The SP approximation that allows the standard normal distribution for perturbation sequence is slightly different [27]. It is of the form

$$(\hat{g}_k)_i(\hat{x}_k) = \frac{\hat{f}(\hat{x}_k + c_k\Delta_k) - \hat{f}(\hat{x}_k - c_k\Delta_k)}{2c_k}\Delta_{k,i}. \tag{3.15}$$

In this case, the second moment is assumed to be bounded instead of its inverse. Although these variants of SP seem similar, the corresponding results can be significantly different [27].

Asymptotic normality like in the FD case (3.13) can be achieved under some additional conditions. Also, the almost sure convergence is proved with an enlarged set of assumption when compared to FD. The sequence $\{c_k\}_{k\in\mathbb{N}}$ retains the important role as in the previously stated methods. For more detailed convergence analysis one can see [62] for instance.

Notice that the methods stated in this section do not need information about the underlying distribution of the noise term. However, if we have some additional information we should use it. Suppose that the cumulative distribution function for $F(x,\xi)$ is known and denote it by G_x. Recall that the objective function is defined as $f(x) = E(F(x,\xi))$. Furthermore, using the ideas of the Monte Carlo sampling techniques [59], we can say that $F(x,\xi) = G_x^{-1}(U)$ where U is Uniformly distributed on interval $(0,1)$. Suppose that the interchange of the gradient and expectation operator is valid, i.e. the equality (3.9) holds. Then we have $\nabla f(x) = \nabla E(F(x,\xi)) = E(\nabla G_x^{-1}(U))$ and we can use the sample realization from the uniform $(0,1)$ distribution $u_1,\ldots,u_N$ to obtain the estimation

$$\hat{g}_k(\hat{x}_k) = \frac{1}{N}\sum_{i=1}^{N} \nabla G_{\hat{x}_k}^{-1}(u_i).$$

This method belongs to the class of direct gradient estimation and it is called Infinitesimal Perturbation Analysis (IPA). Another method of this kind is the Likelihood Ratio method (LR). It is also called the Score Function method. The basic idea is to use the density function of $F(x,\xi)$ (denote it by h_x) and find some suitable density function ψ such that h_x/ψ is well defined. Then under certain conditions such as

(3.9) we obtain

$$\begin{aligned}\nabla f(x) &= \int \nabla F(x,\xi) h_x(\xi) d\xi \\ &= \int \left(\nabla F(x,\xi) \frac{h_x(\xi)}{\psi(\xi)} \right) \psi(\xi) d\xi \\ &= E\left(\nabla F(x,Z) \frac{h_x(Z)}{\psi(Z)} \right)\end{aligned}$$

where the expectation is with respect to the random variable Z with the density function ψ. Therefore, we can use the sample $z_1, \ldots, z_N$ from the distribution ψ to make the approximation

$$\hat{g}_k(\hat{x}_k) = \frac{1}{N} \sum_{i=1}^{N} \nabla F(\hat{x}_k, z_i) \frac{h_{\hat{x}_k}(z_i)}{\psi(z_i)}.$$

This method is very unstable because bad choice for ψ can result in poor gradient approximation. Moreover, this is not exactly the derivative-free method because it uses the information about the gradient function. However, this method usually provides unbiased and strongly consistent estimators [1]. For more information about direct methods one can see [59], [27],[1].

Simultaneous perturbations can also be used to obtain the Hessian approximations like in Adaptive SPSA algorithm Spall [62], [61]. This method adopts Newton-like steps in stochastic framework to obtain the iterative rule

$$\hat{x}_{k+1} = \hat{x}_k - a_k \tilde{H}_k^{-1} \hat{g}_k(\hat{x}_k)$$

where $\tilde{H}_k$ is a positive definite approximation of the Hessian $\nabla^2 f(\hat{x}_k)$. In particular, $\tilde{H}_k = p(\bar{H}_k)$ where p is the projection operator on the space of positive definite matrices. Furthermore,

$$\bar{H}_k = \frac{k}{k+1} \bar{H}_{k-1} + \frac{1}{k+1} \hat{H}_k$$

where $\hat{H}_k = \frac{1}{2}(A_k + A_k^T)$ and

$$A_k = \frac{\hat{g}_k(\hat{x}_k + \tilde{c}_k\tilde{\Delta}_k) - \hat{g}_k(\hat{x}_k - \tilde{c}_k\tilde{\Delta}_k)}{2\tilde{c}_k}(\tilde{\Delta}_{k,1}^{-1}, ..., \tilde{\Delta}_{k,n}^{-1}).$$

Here, the random vector $\tilde{\Delta}_k$ and $\tilde{c}_k$ have the same role as in the gradient approximation and they are defined in the same manner as before. Moreover, if the simultaneous perturbations are used to obtain the gradient estimators as well, then $\tilde{\Delta}_k$ should have the same distribution as Δ_k, but they should be generated independently. Notice that definition of $\bar{H}_k$ is semi recursive in the sense that it uses the approximation of the Hessian at the previous iteration and the approximation obtained at the current one represented by the symmetric matrix $\hat{H}_k$. As the algorithm proceeds, more weight is put on the previous estimator in order to obtain the stability when the noise is strong. Almost sure convergence and asymptotic normality is also attainable in this approach [62].

There are other algorithms that use Newton-like directions in stochastic environment. For example, see Kao et al. [35], [34] where the BFGS formula is used to update the inverse Hessian approximation. At the end of this section, we give references for derivative-free methods in the trust region framework (Conn et al. [13], [14], [15]) where the derivatives are usually approximated by means of interpolation.

3.4 Sample average approximation

Sample average approximation (SAA) is a widely used technique for approaching the problems where the objective function is in the form of mathematical expectation (3.4). The basic idea is to approximate

the objective function with the sample mean

$$\hat{f}_N(x) = \frac{1}{N} \sum_{i=1}^{N} F(x, \xi_i) \tag{3.16}$$

where N is the size of a sample represented by random vectors $\xi_1, \ldots, \xi_N$. The usual approach is to generate the sample at the beginning of the optimization process. That way we can consider the function $\hat{f}_N$ as deterministic which allows us to use standard (deterministic) tools for optimization. Therefore, the problem to be solved is the deterministic optimization problem

$$\min_x \hat{f}_N(x) \tag{3.17}$$

where N is some substantially large but finite number. There are many problems of this form. Some of them are described at the beginning of this chapter: maximum likelihood estimation (3.1), regression (3.2) and data fitting least squares problems (3.3). They have been the issue of many research efforts. The following few chapters analyze the methods for solving this particular kind of problems. In the next section, we will describe some of the known methods that deal with the problem (3.17). We also consider the methods that deal with the sequence of problems of that form where the sample size N is changing. The later ones are sometimes referred to as the Variable Number Sample Path (VNSP) methods, while the first ones are usually called just Sample Path methods. The name is obtained from viewing the realization of noise terms $\xi_1, \ldots, \xi_N$ as the path (trajectory) that sample follows. One should distinguish two very similar names: Variable Number Sample Path and Variable Sample. Variable Number Sample Path implies only that the size of a sample N is allowed to change and it usually means that we are dealing with priory realized sample. So, the sample is cumulative. On the other hand, Variable Sample usually denotes the method that uses different sample realizations.

In this section we will focus on the quality of the solution of the SAA problem (3.17). In other words, our main concern is how does the SAA problem approximates the original problem (3.4) and what are the directions for choosing N such that the solution of the SAA problem provides a good approximation of the original problem solution.

Suppose that $f(x) = E(F(x, \xi))$ is well defined and finite. Furthermore, suppose that $\xi_1, \ldots, \xi_N$ are random variables with the same distribution as ξ. Then, the function $\hat{f}_N(x)$ defined by (3.16) is also a random variable since it depends on a random sample. Moreover, if $\xi_1, \ldots, \xi_N$ are independent, i.e. if the sample is i.i.d., then by the (strong) Law of Large Numbers we obtain the almost sure convergence of $\hat{f}_N(x)$. More precisely, for every x we have that

$$\lim_{N\to\infty} \hat{f}_N(x) = f(x) \tag{3.18}$$

with probability 1. However, this is only the pointwise convergence. Sometimes, the uniform convergence is needed, i.e. we want to know when the following results holds almost surely

$$\lim_{N\to\infty} \sup_{x\in X} |\hat{f}_N(x) - f(x)| = 0. \tag{3.19}$$

If this is true, then we say that $\hat{f}_N$ almost surely converges to f uniformly on the set X. Moreover, we say that the function $F(x, \xi)$, $x \in X$ is dominated by an integrable function if there exists a nonnegative function $M(\xi)$ such that $E(M(\xi)) < \infty$ and $\mathrm{P}(|F(x, \xi)| \leq M(\xi))=1$ for every $x \in X$. Notice that this condition holds if the function F is bounded with some finite constant M, i.e. if $|F(x, \xi)| \leq M$ for every $x \in X$ and almost every ξ. We state the relevant theorem.

Theorem 3.4.1 *[59] Suppose that X is nonempty, compact subset of $\mathbb{R}^n$ and that for any $x \in X$ the function $F(\cdot, \xi)$ is continuous at x*

for almost every ξ. Furthermore, suppose that the sample $\xi_1, \ldots, \xi_N$ is i.i.d. and that the function $F(x, \xi)$, $x \in X$ is dominated by an integrable function. Then $f(x) = E(F(x, \xi))$ is finite valued and continuous on X and $\hat{f}_N$ almost surely converges to f uniformly on X.

Now, let us consider the constrained optimization problem

$$\min_{x \in X} f(x) = E(F(x, \xi)) \tag{3.20}$$

where X is nonempty, closed subset of $\mathbb{R}^n$ which is determined by deterministic constraints and denote by X^* the set of optimal solutions of that problem. Also, let f^* be the optimal value of the objective function, i.e. $f^* = f(x^*)$ where $x^* \in X^*$. Furthermore, denote by $\hat{X}_N^*$ and $\hat{f}_N^*$ the set of optimal solutions and the corresponding optimal values, respectively, of the following problem

$$\min_{x \in X} \hat{f}_N(x) \tag{3.21}$$

where $\hat{f}_N$ is the sample average function (3.16). Notice that $\hat{X}_N^*$ and $\hat{f}_N^*$ are random since they also depend on the sample. The following result holds.

Theorem 3.4.2 *[59] Suppose that $\hat{f}_N$ almost surely converges to f uniformly on X when N tends to infinity. Then $\hat{f}_N^*$ almost surely converges to f^* as $N \to \infty$.*

Stronger assumptions are needed if we want to establish the convergence of the relevant optimal solutions.

Theorem 3.4.3 *[59] Suppose that there exists a compact set $C \subset \mathbb{R}^n$ such that X^* is nonempty and $X^* \subset C$. Assume that the function f is finite valued and continuous on C and that $\hat{f}_N$ converges to f almost surely, uniformly on C. Also, suppose that for N large enough the set $\hat{X}_N^*$ is nonempty and $\hat{X}_N^* \subset C$. Then $\hat{f}_N^* \to f^*$ and $Dev(\hat{X}_N^*, X^*) \to 0$ almost surely as $N \to \infty$.*

In the previous theorem, $Dev(A, B)$ denotes the distance between sets A and B. Moreover, recall that the previous two theorems consider the constrained optimization problem with the closed feasible set. Since our primary interest is in unconstrained optimization problems, we will shortly analyze the way of dealing with this gap. Namely, suppose that the assumptions of Theorem 3.4.3 are true, but for X from (3.20) and (3.21) equal to $\mathbb{R}^n$. Since C is assumed to be compact, we know that there exists a compact set D such that C is a true subset of D. That means that no solution of (3.17) for N large enough lies on the boundary of D. Furthermore, since the constraints are irrelevant unless they are active at the solution, the unconstrained optimization problem $\min_{x\in\mathbb{R}^n} f(x)$ is equivalent to $\min_{x\in D} f(x)$ and for every N large enough the corresponding SAA problems are also equivalent, i.e $\min_{x\in\mathbb{R}^n} \hat{f}_N(x)$ is equivalent to $\min_{x\in D} \hat{f}_N(x)$. Therefore, under the same conditions as in the previous theorem, we can obtain the convergence for the corresponding unconstrained problem. Moreover, the conditions can be relaxed if the problem is convex [59].

Now, let us fix x (for example, x can be a candidate solution) and suppose that $\hat{f}_N(x)$ converges to $f(x)$ almost surely. The important issue here is how fast does it converge. In other words, we want to estimate the error that we make by approximating the expectation function with the sample mean. Assume that the sample is i.i.d. Then $\hat{f}_N(x)$ is unbiased estimator of $f(x)$, i.e.

$$E(\hat{f}_N(x)) = f(x).$$

Moreover, we have that the variance of the estimator is given by

$$D(\hat{f}_N(x)) = \frac{1}{N}\sigma^2(x)$$

where $\sigma^2(x) = D(F(x, \xi))$ is assumed to be finite. Now, we can find the confidence interval, i.e. the error bound $c_N(x)$ such that inequality

$|\hat{f}_N(x) - f(x)| \leq c_N(x)$ is true with some high probability smaller than 1. Define $\delta \in (0,1)$. Our aim is to find $c_N(x)$ such that

$$P\left(|\hat{f}_N(x) - f(x)| \leq c_N(x)\right) = \delta.$$

Under the stated conditions, the Central Limit Theorem yields that the random variable $\hat{f}_N(x)$ is asymptotically normally distributed with $\mathcal{N}(f(x), \frac{1}{N}\sigma^2(x))$. Equivalently,

$$Y_N(x) = \frac{\hat{f}_N(x) - f(x)}{\sqrt{\frac{\sigma^2(x)}{N}}}$$

asymptotically has standard normal distribution. Let $Z : \mathcal{N}(0,1)$. This means that for large N it makes sense to approximate $\Phi_{Y_N(x)}(z)$ with $\Phi_Z(z)$ where $\Phi_W(z)$ denotes the cumulative distribution function of the relevant random variable W. In other words, we can make the following approximation

$$P\left(a \leq Y_N(x) \leq b\right) \approx P\left(a \leq Z \leq b\right)$$

and we have

$$\begin{aligned}
\delta &= P\left(-c_N(x) \leq \hat{f}_N(x) - f(x) \leq c_N(x)\right) \\
&= P\left(\frac{-c_N(x)}{\sqrt{\frac{\sigma^2(x)}{N}}} \leq Y_N(x) \leq \frac{c_N(x)}{\sqrt{\frac{\sigma^2(x)}{N}}}\right) \\
&\approx P\left(\frac{-c_N(x)}{\sqrt{\frac{\sigma^2(x)}{N}}} \leq Z \leq \frac{c_N(x)}{\sqrt{\frac{\sigma^2(x)}{N}}}\right).
\end{aligned}$$

Therefore we can approximate $\sqrt{N}c_N(x)/\sigma(x)$ with the quantile of standard normal distribution, or more precisely with $z_{\frac{1+\delta}{2}}$ such that

$\Phi_Z(z_{\frac{1+\delta}{2}}) = \frac{1+\delta}{2}$. Furthermore, if we approximate $\sigma^2(x)$ with the sample variance

$$\hat{\sigma}_N^2(x) = \frac{1}{N-1}\sum_{i=1}^{N}(F(x,\xi_i) - \hat{f}_N(x))^2 \tag{3.22}$$

we obtain the error bound estimation

$$\hat{c}_N(x) = \frac{\hat{\sigma}_N(x)}{\sqrt{N}} z_{\frac{1+\delta}{2}}. \tag{3.23}$$

Therefore, if we obtain a point $\bar{x}$ as the approximate solution of (3.17), we can use $\hat{f}_N(\bar{x}) + \hat{c}_N(\bar{x})$ to estimate the upper bound of the objective function value $f(\bar{x})$. Notice that for fixed x the error bound $\hat{c}_N(x)$ increases if δ increases. Furthermore, $\hat{c}_N(x)$ is also directly proportional to the variance of the estimator $D(\hat{f}_N(x))$. Therefore, some techniques for reducing that variance are developed. Some of them are the quasi-Monte Carlo and Latin hypercube sampling, as well as the likelihood ratio method mentioned earlier [59].

There are situations where we just want to compare two points and decide which one is better. For example, these two points can be the neighboring iterates of an algorithm $\hat{x}_k$ and $\hat{x}_{k+1}$ and we want to decide whether to accept the next iterate or not by estimating $f(\hat{x}_k) - f(\hat{x}_{k+1})$. In these kind of situations, the concept of common random numbers (CRN), i.e. using the same sample can be very useful especially if the iterates are close to each other. In that case, the sample average estimators are usually strongly positively correlated, i.e. the covariance $Cov(\hat{f}_N(\hat{x}_k), \hat{f}_N(\hat{x}_{k+1}))$ is significantly larger than zero and therefore

$$\begin{aligned} D(\hat{f}_N(\hat{x}_k) - \hat{f}_N(\hat{x}_{k+1})) &= D(\hat{f}_N(\hat{x}_k)) + D(\hat{f}_N(\hat{x}_{k+1})) \\ &- 2Cov(\hat{f}_N(\hat{x}_k), \hat{f}_N(\hat{x}_{k+1})) \\ &< D(\hat{f}_N(\hat{x}_k)) + D(\hat{f}_N(\hat{x}_{k+1})). \end{aligned}$$

On the other hand, if we use two independent samples to estimate $\hat{f}_N(\hat{x}_k)$ and $\hat{f}_N(\hat{x}_{k+1})$ we obtain

$$D(\hat{f}_N(\hat{x}_k) - \hat{f}_N(\hat{x}_{k+1})) = D(\hat{f}_N(\hat{x}_k)) + D(\hat{f}_N(\hat{x}_{k+1})).$$

Therefore, $\hat{f}_N(\hat{x}_k) - \hat{f}_N(\hat{x}_{k+1})$ probably provides more reliable information for choosing the better point if the CRN concept is used. Although there are many advantages of the CRN approach, using different samples can still be beneficial sometimes (Homem-de-Mello [33]).

Now, suppose that we have some candidate solution point $\bar{x}$ and we want not only to estimate the difference $\hat{f}_N(\bar{x}) - f(\bar{x})$ but also the gap defined by

$$g(\bar{x}) = f(\bar{x}) - f(x^*)$$

where x^* is the solution of the original unconstrained problem. Of course, $g(x) \geq 0$ for every x and our interest is in finding the upper bound. As before, denote by X^* and $\hat{X}_N^*$ the sets of optimal solutions and by f^* and $\hat{f}_N^*$ the optimal values of the problems (3.20) and (3.21) with $X = \mathbb{R}^n$, respectively. Then, for every x' we have that

$$\hat{f}_N(x') \geq \min_{x \in \mathbb{R}^n} \hat{f}_N(x) = \hat{f}_N^*.$$

Suppose that the sample is i.i.d.. Then the previous inequality implies

$$f(x') = E(\hat{f}_N(x')) \geq E(\hat{f}_N^*).$$

Since this is true for every x', we have that

$$\min_{x' \in \mathbb{R}^n} f(x') \geq E(\hat{f}_N^*).$$

The left hand side of the above inequality is equal to f^* and therefore we obtain that

$$E(\hat{f}_N^*) \leq f^*. \tag{3.24}$$

It can be shown [59] that $E(\hat{f}_N^*)$ is increasing with respect to sample size and that, under some additional conditions, $\hat{f}_N^*$ is asymptotically normally distributed with mean f^* and variance $\sigma^2(x^*)/\sqrt{N}$ where $X^* = \{x^*\}$. However, if X^* is not a singleton, the estimator $\hat{f}_N^*$ is asymptotically biased in general. More precisely, $E(\hat{f}_N^*)$ is typically smaller than f^*. Now, the idea is to find the confidence interval for the gap $g(\bar{x})$ by finding the confidence lower bound for $E(\hat{f}_N^*)$ and upper bound for $f(\bar{x})$.

Suppose that we have M independent samples of size N, i.e. we have i.i.d. sample $\xi_1^m, \ldots, \xi_N^m$, $m = 1, \ldots, M$. Denote by $\hat{f}_N^{m*}$ the relevant optimal values. Then we can form an unbiased estimator of the expectation $E(\hat{f}_N^*)$ by defining

$$\hat{f}_{N,M}^* = \frac{1}{M} \sum_{m=1}^{M} \hat{f}_N^{m*}.$$

Therefore, this estimator has the mean $E(\hat{f}_{N,M}^*) = E(\hat{f}_N^*)$ and the variance $D(\hat{f}_{N,M}^*) = D(\hat{f}_N^*)/M$ which we can estimate by

$$\hat{\sigma}_{N,M}^2 = \frac{1}{M} \left(\frac{1}{M-1} \sum_{m=1}^{M} \left(\hat{f}_N^{m*} - \hat{f}_{N,M}^* \right)^2 \right).$$

By the Central Limit Theorem, $\hat{f}_{N,M}^*$ has approximately normal distribution for large M. However, this approach indicates that we have to solve the optimization problem (3.21) M times and therefore M is usually rather modest. Therefore, we use Student's t-distribution to make the approximation. Denote by T_{M-1} the random variable that has the Student's distribution with $M-1$ degrees of freedom. Then

we have

$$\begin{aligned}\delta &= P\left(E(\hat{f}_N^*) > L_{N,M}\right) = P\left(\frac{\hat{f}_{N,M}^* - E(\hat{f}_N^*)}{\sqrt{D(\hat{f}_{N,M}^*)}} < \frac{\hat{f}_{N,M}^* - L_{N,M}}{\sqrt{D(\hat{f}_{N,M}^*)}}\right) \\ &= P\left(\frac{\hat{f}_{N,M}^* - E(\hat{f}_{N,M}^*)}{\sqrt{D(\hat{f}_{N,M}^*)}} < \frac{\hat{f}_{N,M}^* - L_{N,M}}{\sqrt{D(\hat{f}_{N,M}^*)}}\right) \\ &\approx P\left(T_{M-1} < \frac{\hat{f}_{N,M}^* - L_{N,M}}{\sqrt{D(\hat{f}_{N,M}^*)}}\right).\end{aligned}$$

Therefore, we can approximate the lower bound of the δ confidence interval by

$$\hat{L}_{N,M} = \hat{f}_{N,M}^* - t_{M-1,\delta}\hat{\sigma}_{N,M} \tag{3.25}$$

where $t_{M-1,\delta}$ is the quantile of Student's T_{M-1} distribution.

We can approximate $f(\bar{x})$ by using sample average $\hat{f}_{N'}(\bar{x})$ with some large enough sample size N'. Therefore, we can use the normal distribution to approximate the upper bound for one-sided confidence interval as follows.

$$\begin{aligned}\delta &= P\left(f(\bar{x}) \le \hat{f}_{N'}(\bar{x}) + U_{N'}(\bar{x})\right) \\ &= P\left(\frac{\hat{f}_{N'}(\bar{x}) - f(\bar{x})}{\sqrt{D(\hat{f}_{N'}(\bar{x}))}} > \frac{-U_{N'}(\bar{x})}{\sqrt{D(\hat{f}_{N'}(\bar{x}))}}\right) \\ &\approx P\left(Z > \frac{-U_{N'}(\bar{x})}{\sqrt{D(\hat{f}_{N'}(\bar{x}))}}\right).\end{aligned}$$

Here, Z represents standard normal distribution. If we denote its

quantile by z_δ, we obtain the upper bound estimate

$$\hat{f}_{N'}(\bar{x}) + \hat{U}_{N'}(\bar{x}) = \hat{f}_{N'}(\bar{x}) + z_\delta \frac{\hat{\sigma}_{N'}(\bar{x})}{\sqrt{N'}} \tag{3.26}$$

where $\hat{\sigma}_{N'}(\bar{x})$ is defined by (3.22). Finally, the confidence interval upper bound for the gap $g(\bar{x})$ is approximated by

$$\hat{f}_{N'}(\bar{x}) + z_\delta \frac{\hat{\sigma}_{N'}(\bar{x})}{\sqrt{N'}} - \hat{f}^*_{N,M} + t_{M-1,\delta}\hat{\sigma}_{N,M}.$$

This bound can be used, for example, as the stopping criterion in algorithms. In [59], the bounds for sample sizes such that the solutions of an approximate problem are nearly optimal for the true problem with some high probability are developed. However, they are mainly too conservative for practical applications in general. At the end, we mention that if the problem is constrained, then one may consider Stochastic Generalized Equations approach [59].

3.5 Variable number sample path methods

In this section we focus on methods that use variable sample sizes in order to solve the optimization problem (3.4). Roughly speaking, there are two approaches. The first one deals with unbounded sample size and the main issue is how to increase it during the optimization process. The second type of algorithms deals with finite sample size which is assumed to be determined before the process of optimization starts. It also contains methods that are applied on regression, maximum likelihood or least squares problems stated in the first section of this chapter. As we have seen, these kind of problems do not assume explicit or even implicit noise. Moreover, problems of the form

(3.17) can also be considered as "noise-free" if we adopt the approach of generating the sample at the beginning and observing the sample average objective function as deterministic one.

First, we review the relevant methods that deal with the unbounded sample size and use the so called diagonalization scheme. This scheme approximately solves the sequence of problems of the form (3.17). The sequence of problems is determined by the sequence of a sample sizes which yield different objective functions. The main issue is when to increase the sample size and switch to the next level.

For example, in Royset [54], the optimality function is used to determine when to switch to the larger sample size. It is defined by mapping $\theta : \mathbb{R}^n \to (-\infty, 0]$ which , under some conditions, satisfies $\theta(x) = 0$ if and only if x is a solution in some sense. Fritz-John optimality conditions are considered instead of KKT conditions because they alow generalization, i.e. constraint qualifications are not required. The focus is on the constraints, but in the unconstrained case the optimality function reduces to $\theta(x) = -\frac{1}{2}\|\nabla f(x)\|^2$. In general, optimality function is approximated by the sample average function θ_N and almost sure convergence of θ_N towards θ is stated together with asymptotic normality. The algorithm increases the sample size when $\theta_N \geq -\delta_1 \Delta(N)$, where δ_1 is some positive constant and Δ is a function that maps $\mathbb{N}$ into $(0, \infty)$ and satisfies $\lim_{N\to\infty} \Delta(N) = 0$. However, the dynamics of increasing the sample size is not specified.

Guidance for the optimal relation between the sample size and the error tolerance sequences is considered in [48]. Error tolerance is represented by the deviation of an approximate solution of problem (3.17) from a real solution of that problem. The measure of effectiveness is defined as the product of the deviation and the number of simulation calls. It is stated that the error tolerance should not be decreased faster than the sample size is increased. The dynamics of change depends on the convergence rate of numerical procedures used to solve the problems (3.17). Moreover, specific recommendations are given

for linear, sublinear and polynomial rates. For example, if the applied algorithm is linearly convergent, then the linear growth of a sample size is recommended, i.e. it can be set $N_{k+1} = \lceil 1.1 N_k \rceil$ for example. Also, in that case, polynomial (for example $N_{k+1} = \lceil N_k^{1.1} \rceil$) or exponential ($N_{k+1} = \lceil e^{N_k^{1.1}} \rceil$) growth are not recommended. Furthermore, it is implied that the error tolerance sequence should be of the form $K/\sqrt{N_k}$ where K is some positive constant.

Recall that almost sure convergence of $\hat{f}_N(x)$ towards $f(x)$ is achieved if the sample is i.i.d. and the considered functions are well defined and finite. However, if the sample is not i.i.d. or we do not have cumulative sample, the almost sure convergence is achievable only if the sample size N increases at the certain rate. This is the main issue of the paper [33]. It is stated that the sample size sequence $\{N_k\}_{k \in \mathbb{N}}$ should satisfy $\sum_{k=1}^{\infty} \alpha^{N_k}$ for every $\alpha \in (0, 1)$. Then, if the function $F(x, \xi)$ is bounded in some sense and $\hat{f}_N(x)$ is asymptotically unbiased estimator of $f(x)$, the almost sure convergence mentioned above is achieved. For example, $N_k \geq \sqrt{k}$ satisfies the considered condition. However, too fast increase of the sample size can result in an inefficient algorithm. Therefore, it is suggested that statistical t-test should be applied in order to decide when to go up to the next level, i.e. to increase the sample size. Namely, after every K iterations the test is applied to show if the significant improvement in f is achieved when comparing the subsequent iterates. If this is true, N is not increased. On the other hand, if the algorithm starts to crawl then the sample should probably get bigger. Besides the possibility of using the relevant statistical tools, different samples in different iterations can also help algorithm to overcome the trap of a single sample path.

In Polak, Royset [50] the focus is on the finite sample size N although the almost sure convergence is addressed. The problem under consideration is with constraints, but penalty function is used to transform it into an unconstrained optimization problem. The idea is to

vary the sample size during the process, or more precisely, to approximately solve the sequence of problems in a form (3.17) with $N = N_i$, $i = 1, \ldots, s$ applying n_i iterations at every stage i. The sample size is nondecreasing. Before starting to solve this sequence of problems, an additional problem is solved in order to find the optimal sequence of sample sizes N_i and iterations n_i, as well as the number of stages s. This is done by minimizing the overall cost $\sum_{i=1}^{s} n_i w(N_i)$ where $w(N)$ is the estimated cost of one iteration of the algorithm applied on $\hat{f}_N$. The constraint for this problem is motivated by the stopping criterion $f(x_k) - f^* \leq \varepsilon(f(x_0) - f^*)$ where f^* is the optimal value of the objective function. The left-hand side of this inequality is estimated by using the confidence interval bound for $f(x)$ and parameter $\theta \in (0, 1)$ that determines the linear rate of convergence which is assumed for algorithm applied on $\hat{f}_N$.

Now, we refer to the relevant methods that focus on updating the sample size at every iteration. Therefore, these methods may deal with different function at every iteration of the optimization process. The first one described below deals with an unbounded sample size in general, while the remaining two focus mainly on solving the problem (3.17) with some large but finite N.

In Deng, Ferris [22] the unconstrained optimization problem is considered, but the derivatives are assumed to be unavailable. Therefore, a quadratic model function Q_k^N is used to approximate the function $\hat{f}_{N_k}$ at every iteration in some region determined by the trust region radius Δ_k. This is done by using the interpolation proposed by Powell [51]. The candidate iterate is found within the trust region and therefore the trust region framework is applied. The points used for interpolation $y^1, \ldots, y^L$ are also used to estimate the posterior distributions of the gradient g_k^∞ of the model function for f. More precisely, if we denote by X^N the matrix that contains $F(y^i, \xi_j)$, $i = 1, \ldots, L$, $j = 1, \ldots, N$, then the posterior distribution of the gradient, i.e. the

distribution of $g_k^\infty | X^N$, is approximated by the normal distribution. It is known that the candidate point x_{k+1} satisfies

$$Q_k^{N_k}(x_k) - Q_k^{N_k}(x_{k+1}) \geq h(\|g_k^{N_k}\|)$$

where the function h is known but we will not specify it here. The candidate point is obtain by observing the function $\hat{f}_{N_k}$ and the question is whether that point is also good for the original function model. This is examined by observing the event

$$E_k^{N_k} : \; Q_k^{N_k}(x_k) - Q_k^{N_k}(x_{k+1}) < h(\|g_k^\infty\|).$$

If the probability of that event $P(E_k^{N_k})$ is sufficiently small, then there is no need to increase the sample size. On the other hand, the sample size should be increased until this is satisfied. The trial sample sizes are updated with some incremental factor. The probability $P(E_k^{N_k})$ is approximated by the so called Bayes risk $P(E_k^{N_k} | X^{N_k})$, i.e

$$P(E_k^{N_k}) \approx P(Q_k^{N_k}(x_k) - Q_k^{N_k}(x_{k+1}) < h(\|g_k^\infty | X^{N_k}\|)).$$

Furthermore, the simulations from the approximated posterior distribution are used to obtain the relative frequency approximation of the Bayes risk. Almost sure convergence is analyzed and the sample size is unbounded in general. Moreover, the authors constructed an example where the sample size remains bounded during the whole optimization process and almost sure convergence is still obtained.

Data fitting applications are considered in [26] where the objective function can be considered as the sample average function in the form

$$f(x) = \hat{f}_N(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x).$$

The authors consider quasi-Newton methods and therefore the gradient information is needed. In order to combine two diametral approaches: using the full gradient $\nabla f(x)$ and using $\nabla f_i(x)$ for some

$i \in \{1, 2, \ldots, N\}$ as an approximation of the gradient, they constructed the hybrid algorithm that increases the number of functions f_i whose gradients are evaluated in order to obtain the gradient approximation. This hybrid algorithm can be considered as the increasing sample size method where the sample size is bounded by N. The main concern is the rate of convergence and the convergence analysis is done with the assumption of a constant step size. Two approaches are considered: deterministic and stochastic sampling. The deterministic sampling assumes that if the sample size is N_k then the gradients to be evaluated $\nabla f_i(x_k)$ are determined in advance. For example, we can use first N_k functions to obtain the gradient approximation $g_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \nabla f_i(x_k)$. On the other hand, the stochastic sampling assumes that the gradients to be evaluated are chosen randomly. It is stated that R-linear convergence can be achieved if the sample sizes satisfy $\left(\frac{N-N_k}{N}\right)^2 = \mathcal{O}(\gamma^k)$ in deterministic and $\frac{N-N_k}{NN_k} = \mathcal{O}(\gamma^k)$ in stochastic case for some $\gamma \in (0, 1)$. Moreover, q-linear convergence is also analyzed but under stronger conditions. In numerical experiments for instance, the dynamics of $N_{k+1} = \lceil \min\{1.1N_k + 1, N\} \rceil$ is used.

Finally, we refer to the algorithm which uses the trust region framework and focuses on the finite sample size problem (3.17). The important characteristic of that approach is that it allows the sample size N_k to decrease during the optimization process (Bastin et al. [4], [3]). The model function for $\hat{f}_{N_k}$ is formed at every iteration and the basic idea for updating the sample size is to compare the decrease in the model function with the confidence interval bound approximation of the form (3.23). Roughly speaking, the sample size is determined in a way that provides good agreement of these two measures. More details about this reasoning are to be presented in the next chapter.

Chapter 4

Line search methods with variable sample size

In this chapter, we introduce the optimization method that uses the line search technique described in Chapter 2. The line search framework is one of the two most important features of the considered method and it will be further developed in the direction of nonmonotone line search within the following chapter. The other important characteristic is allowing the sample size to oscillate (Krejić, Krklec [37]) which complicates the convergence analysis since we are working with a different functions during the optimization process. This part of the thesis represents the original contribution. But let us start by defining the problem.

The problem under consideration is

$$\min_{x \in \mathbb{R}^n} f(x). \tag{4.1}$$

Function $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be in the form of mathematical expectation

$$f(x) = E(F(x, \xi)),$$

where $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, ξ is a random vector $\xi : \Omega \to \mathbb{R}^m$ and $(\Omega, \mathcal{F}, P)$ is a probability space. The form of mathematical expectation makes this problem difficult to solve as very often one can not find its analytical form. This is the case even if the analytical form of F is known, which is assumed in this chapter.

One way of dealing with this kind of problem is to use sample averaging in order to approximate the original objective function as follows

$$f(x) \approx \hat{f}_N(x) = \frac{1}{N} \sum_{i=1}^{N} F(x, \xi_i). \tag{4.2}$$

Here N represents the size of sample that is used to make approximation (4.2). An important assumption is that we form the sample by random vectors $\xi_1, \ldots, \xi_N$ that are independent and identically distributed. If F is bounded then the Law of Large Numbers [59] implies that for every x almost surely

$$\lim_{N \to \infty} \hat{f}_N(x) = f(x). \tag{4.3}$$

In practical applications one can not have an unbounded sample size but can get close to the original function by choosing a sample size that is large enough but still finite. So, we will focus on finding an optimal solution of

$$\min_{x \in \mathbb{R}^n} \ \hat{f}_N(x), \tag{4.4}$$

where N is a fixed integer and $\xi_1, \ldots, \xi_N$ is a sample realization that is generated at the beginning of the optimization process. Thus the problem we are considering is in fact deterministic and standard optimization tools are applicable. As we have seen, this approach is called the sample path method or the stochastic average approximation (SAA) method and it is the subject of many research efforts ([59], [62]). The main disadvantage of the SAA method is the need to calculate the expensive objective function defined by (4.2) in each iteration.

As N in (4.4) needs to be large, the evaluations of $\hat{f}_N$ become very costly. That is particularly true in the practical applications where the output parameters of models are expensive to calculate. Given that almost all optimization methods include some kind of gradient information, or even second order information, the cost becomes even higher.

As one can see in Chapter 3, various attempts to reduce the costs of SAA methods are presented in the literature. Roughly speaking, the main idea is to use some kind of variable sample size strategy and work with smaller samples whenever possible, at least at the beginning of the optimization process. One can distinguish two types of variable sample size results. The first type deals with unbounded samples and seeks convergence in stochastic sense not allowing the sample size to decrease ([22],[33],[50], Pasuphaty [47] and [48]). The second type of algorithm deals directly with problems of type (4.4) and seeks convergence towards stationary points of that problem. The algorithms proposed in [3] and [4] introduce a variable sample size strategy that allows a decrease of the sample size as well as an increase during the optimization process. Roughly speaking, the main idea is to use the decrease of the function value and a measure of the width of the confidence interval to determine the change in sample size. The optimization process is conducted in the trust region framework. We will adopt these ideas to the line search framework and propose an algorithm that allows both an increase and decrease of sample size during the optimization process. Given that the final goal is to make the overall process less costly, we also introduce an additional safeguard rule that prohibits unproductive sample decreases [37]. As common for this kind of problems, the measure of cost is the number of function evaluations (Moré, Wild [45]).

4.1 Preliminaries

In order to solve (4.4) we will assume that we know the analytical form of a gradient $\nabla_x F(x, \xi)$. This implies that we are able to calculate the true gradient of the function $\hat{f}_N$, that is

$$\nabla \hat{f}_N(x) = \frac{1}{N} \sum_{i=1}^{N} \nabla_x F(x, \xi_i).$$

Once the sample is generated, we consider the function $\hat{f}_N$ and the problem (4.4) as deterministic (Fu [28]). This approach simplifies the definition of stationary points which is much more complicated in a stochastic environment. It also provides the standard optimization tools described in Chapter 2. The key issue is the variable sample scheme.

Suppose that we are at the iteration k, i.e. at the point x_k. Every iteration has its own sample size N_k, therefore we are considering the function

$$\hat{f}_{N_k}(x) = \frac{1}{N_k} \sum_{i=1}^{N_k} F(x, \xi_i).$$

We perform line search along the direction p_k which is decreasing for the considered function, i.e. it satisfies the condition

$$p_k^T \nabla \hat{f}_{N_k}(x_k) < 0. \tag{4.5}$$

In order to obtain a sufficient decrease of the objective function, we use the backtracking technique to find a step size α_k which satisfies the Armijo condition

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \hat{f}_{N_k}(x_k) + \eta \alpha_k p_k^T \nabla \hat{f}_{N_k}(x_k), \tag{4.6}$$

for some $\eta \in (0, 1)$. More precisely, starting from $\alpha = 1$, we decrease α by multiplying it with $\beta \in (0, 1)$ until the Armijo condition (4.6) is

satisfied. This can be done in a finite number of trials if the iteration x_k is not a stationary point of $\hat{f}_{N_k}$ assuming that this function is continuously differentiable and bounded from below.

After the suitable step size α_k is found, we define the next iterate as $x_{k+1} = x_k + \alpha_k p_k$. Now, the main issue is how to determine a suitable sample size N_{k+1} for the following iteration. In the algorithm that we propose the rule for determining N_{k+1} is based on three parameters: the decrease measure dm_k, the lack of precision denoted by $\varepsilon_\delta^{N_k}(x_k)$ and the safeguard rule parameter ρ_k. The two measures of progress, dm_k and $\varepsilon_\delta^{N_k}(x_k)$ are taken from [4] and [3] and adopted to suit the line search methods while the third parameter is introduced to avoid an unproductive decrease of the sample size as will be explained below.

The decrease measure is defined as

$$dm_k = -\alpha_k p_k^T \nabla \hat{f}_{N_k}(x_k). \tag{4.7}$$

This is exactly the decrease in the linear model function, i.e.

$$dm_k = m_k^{N_k}(x_k) - m_k^{N_k}(x_{k+1}),$$

where

$$m_k^{N_k}(x_k + s) = \hat{f}_{N_k}(x_k) + s^T \nabla \hat{f}_{N_k}(x_k).$$

The lack of precision represents an approximate measure of the width of confidence interval for the original objective function f, i.e.

$$\varepsilon_\delta^{N_k}(x_k) \approx c,$$

where

$$P(f(x_k) \in [\hat{f}_{N_k}(x_k) - c, \hat{f}_{N_k}(x_k) + c]) \approx \delta.$$

The confidence level δ is usually equal to 0.9, 0.95 or 0.99. It will be an input parameter of our algorithm. We know that $c = \sigma(x_k)\alpha_\delta/\sqrt{N_k}$, where $\sigma(x_k)$ is the standard deviation of random variable $F(x_k, \xi)$ and

α_δ is the quantile of the normal distribution, i.e. $P(-\alpha_\delta \leq X \leq \alpha_\delta) = \delta$, where $X : \mathcal{N}(0,1)$. Usually we can not find $\sigma(x_k)$ so we use the centered sample variance estimator

$$\hat{\sigma}^2_{N_k}(x_k) = \frac{1}{N_k - 1} \sum_{i=1}^{N_k} (F(x_k, \xi_i) - \hat{f}_{N_k}(x_k))^2.$$

Finally, we define the lack of precision as

$$\varepsilon_\delta^{N_k}(x_k) = \hat{\sigma}_{N_k}(x_k) \frac{\alpha_\delta}{\sqrt{N_k}}. \tag{4.8}$$

The algorithm that provides a candidate N_k^+ for the next sample size will be described in more detail in the following section. The main idea is to compare the previously defined lack of precision and the decrease measure. Roughly speaking if the decrease in function's value is large compared to the width of the confidence interval then we decrease the sample size in the next iteration. In the opposite case, when the decrease is relatively small in comparison with the precision then we increase the sample size. Furthermore, if the candidate sample size is lower than the current one, that is if $N_k^+ < N_k$, one more test is applied before making the final decision about the sample size to be used in the next iteration. In that case, we calculate the safeguard parameter ρ_k. It is defined throughout the ratio between the decrease in the candidate function and the function that has been used to obtain the next iteration, that is

$$\rho_k = \left| \frac{\hat{f}_{N_k^+}(x_k) - \hat{f}_{N_k^+}(x_{k+1})}{\hat{f}_{N_k}(x_k) - \hat{f}_{N_k}(x_{k+1})} - 1 \right|. \tag{4.9}$$

The role of ρ_k is to prevent an unproductive sample size decrease i.e. we calculate the progress made by the new point and the candidate sample size and compare it with the progress achieved with N_k. Ideally,

the ratio is equal to 1 and $\rho_k = 0$. However, if ρ_k is relatively large then these two decrease measures are too different and we do not allow a decrease of the sample size.

Now, we present the assumptions needed for the further analysis.

C 1 *Random vectors $\xi_1, \ldots, \xi_N$ are independent and identically distributed.*

A 1 *For every ξ, $F(\cdot, \xi) \in C^1(\mathbb{R}^n)$.*

A 2 *There exists a constant $M_1 > 0$ such that for every ξ, x $\|\nabla_x F(x, \xi)\| \leq M_1$.*

A 3 *There exists constant M_F such that for every ξ, x, $M_F \leq F(x, \xi)$.*

A 4 *There exists constant M_{FF} such that for every ξ, x, $F(x, \xi) \leq M_{FF}$.*

The role of the first assumption is already clear. It ensures that our approximation function $\hat{f}_{N_k}$ is, in fact, a centered estimator of the function f at each point. This is not a fundamental assumption that makes the upcoming algorithm convergent, but it is important for making the problem (4.4) close to the original one for N large enough.

The assumption A1 ensures the continuity and differentiability of F as well as of $\hat{f}_N$. More formally, we have the following lemma.

Lemma 4.1.1 *If the assumption A1 is satisfied, then for every $N \in \mathbb{N}$ the function $\hat{f}_N$ is in $C^1(\mathbb{R}^n)$.*

One of the crucial assumptions for proving the convergence result is A3. Moreover, the assumption A3 makes our problem solvable since it implies the following result.

Lemma 4.1.2 *If the assumption A3 holds, then* $M_F \leq \hat{f}_N(x)$ *is true for all* $x \in \mathbb{R}^n$ *and every* $N \in \mathbb{N}$.

An analogous result can be obtained if the function F is bounded from above. Both results can be proved just by using the fact that the sample average function is a linear combination of functions $F(\cdot, \xi_i)$, $i = 1, \ldots, N$.

Lemma 4.1.3 *If the assumption A4 holds, then* $\hat{f}_N(x) \leq M_{FF}$ *is true for all* $x \in \mathbb{R}^n$ *and every* $N \in \mathbb{N}$.

Moreover, the previously stated assumptions imply the boundedness of the sample average function's gradient as stated below.

Lemma 4.1.4 *If the assumptions A1 and A2 hold, then for every* $x \in \mathbb{R}^n$ *and every* $N \in \mathbb{N}$ *holds* $\|\nabla \hat{f}_N(x)\| \leq M_1$.

Proof. Let N be an arbitrary positive integer. Then for every $x \in \mathbb{R}^n$ we have

$$\|\nabla \hat{f}_N(x)\| = \|\frac{1}{N}\sum_{i=1}^{N} \nabla_x F(x, \xi_i)\| \leq \frac{1}{N}\sum_{i=1}^{N} \|\nabla_x F(x, \xi_i)\| \leq M_1.$$

■

An important consequence of the previous assumptions is that the interchange between the mathematical expectation and the gradient operator is allowed [59], i.e. the following is true

$$\nabla_x E(F(x, \xi)) = E(\nabla_x F(x, \xi)). \tag{4.10}$$

Having this in mind, we can use the Law of Large Numbers again, and conclude that for every x almost surely

$$\lim_{N \to \infty} \nabla \hat{f}_N(x) = \nabla f(x).$$

This justifies using $\nabla \hat{f}_N(x)$ as an approximation of the measure of stationarity for problem (4.1). We have influence on that approximation because we can change the sample size N and, hopefully, make problem (4.4) closer to problem (4.1). Therefore (4.10), together with assumption C1, helps us measure the performance of our algorithm regarding (4.1). Finally, previously stated results together with Lemma 2.2.1 will guaranty that the considered line search is well defined.

4.2 The algorithms

The method that we are going to present is constructed to solve the problem (4.4) with the sample size N equal to some N_{max} which is considered as an input parameter. We assume that the suitable maximal sample size N_{max} can be determined without entering into the details of such a process (some guidance is given in Chapter 3). More precisely, we are searching for a stationary point of the function $\hat{f}_{N_{max}}$. The sample realization that defines the objective function $\hat{f}_{N_{max}}$ is generated at the beginning of the optimization process. Therefore, we can say that the aim of the algorithm is to find a point x which satisfies

$$\|\nabla \hat{f}_{N_{max}}(x)\| = 0.$$

As already stated, the algorithm is constructed to let the sample size vary across the iterations and to let it decrease if appropriate. Moreover, under some mild conditions, the maximal sample size is eventually reached. Let us state the main algorithm here leaving the additional ones to be stated later.

ALGORITHM 1

S0 Input parameters: $N_{max}, N_0^{min} \in \mathbb{N}$, $x_0 \in \mathbb{R}^n$, $\delta, \eta, \beta, \nu_1, d \in (0, 1)$.

S1 Generate the sample realization: $\xi_1, \ldots, \xi_{N_{max}}$.

Set $k = 0$, $N_k = N_0^{min}$.

S2 Compute $\hat{f}_{N_k}(x_k)$ and $\varepsilon_\delta^{N_k}(x_k)$ using (4.2) and (4.8).

S3 Test

If $\|\nabla \hat{f}_{N_k}(x_k)\| = 0$ and $N_k = N_{max}$ then STOP.

If $\|\nabla \hat{f}_{N_k}(x_k)\| = 0$, $N_k < N_{max}$ and $\varepsilon_\delta^{N_k}(x_k) > 0$ put $N_k = N_{max}$ and $N_k^{min} = N_{max}$ and go to step S2.

If $\|\nabla \hat{f}_{N_k}(x_k)\| = 0$, $N_k < N_{max}$ and $\varepsilon_\delta^{N_k}(x_k) = 0$ put $N_k = N_k + 1$ and $N_k^{min} = N_k^{min} + 1$ and go to step S2.

If $\|\nabla \hat{f}_{N_k}(x_k)\| > 0$ go to step S4.

S4 Determine p_k such that $p_k^T \nabla \hat{f}_{N_k}(x_k) < 0$.

S5 Find the smallest nonnegative integer j such that $\alpha_k = \beta^j$ satisfies

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \hat{f}_{N_k}(x_k) + \eta \alpha_k p_k^T \nabla \hat{f}_{N_k}(x_k).$$

S6 Set $s_k = \alpha_k p_k, \quad x_{k+1} = x_k + s_k$ and compute dm_k using (4.7).

S7 Determine the candidate sample size N_k^+ using Algorithm 2.

S8 Determine the sample size N_{k+1} using Algorithm 3.

S9 Determine the lower bound of the sample size N_{k+1}^{min}.

S10 Set $k = k + 1$ and go to step S2.

Before stating the auxiliary algorithms, let us briefly comment on this one. The point x_0 is an arbitrary starting point. The sample realization generated in step S1 is the one that is used during the whole optimization process. For simplicity, if the required sample size is $N_k < N_{max}$, we can take the first N_k realizations in order to calculate all relevant values. On the other hand, N_0^{min} is the lowest sample size that is going to be used in the algorithm. The role of the lower sample bound N_k^{min} will be clear after we state the remaining algorithms. The same is true for parameters d and ν_1.

Notice that the algorithm terminates after a finite number of iterations only if x_k is a stationary point of the function $\hat{f}_{N_{max}}$. Moreover, step S3 guarantees that we have a decreasing search direction in step S5, therefore the backtracking is well defined.

As we already mentioned, one of the main issues is how to determine the sample size that is going to be used in the next iteration. Algorithms 2 and 3 stated below provide details. Algorithm 2 leads us to the candidate sample size N_k^+. Acceptance of that candidate is decided within Algorithm 3. We will explain latter how to update N_k^{min}. For now, the important thing is that the lower bound is determined before we get to step S7 and it is considered as an input parameter in the algorithm described below. Notice that the following algorithm is constructed to provide

$$N_k^{min} \leq N_k^+ \leq N_{max}.$$

ALGORITHM 2

S0 Input parameters: $dm_k,\ N_k^{min},\ \varepsilon_\delta^{N_k}(x_k),\ \nu_1 \in (0,1), d \in (0,1]$.

S1 Determine N_k^+

1) $dm_k = d\ \varepsilon_\delta^{N_k}(x_k) \quad \rightarrow \quad N_k^+ = N_k$.

2) $dm_k > d\ \varepsilon_\delta^{N_k}(x_k)$
Starting with $N = N_k$, while $dm_k > d\ \varepsilon_\delta^{N}(x_k)$ and $N > N_k^{min}$, decrease N by 1 and calculate $\varepsilon_\delta^{N}(x_k) \quad \rightarrow \quad N_k^+$.

3) $dm_k < d\ \varepsilon_\delta^{N_k}(x_k)$

i) $dm_k \geq \nu_1 d\ \varepsilon_\delta^{N_k}(x_k)$
Starting with $N = N_k$, while $dm_k < d\ \varepsilon_\delta^{N}(x_k)$ and $N < N_{max}$, increase N by 1 and calculate $\varepsilon_\delta^{N}(x_k) \quad \rightarrow$ N_k^+.

ii) $dm_k < \nu_1 d\ \varepsilon_\delta^{N_k}(x_k) \quad \rightarrow \quad N_k^+ = N_{max}$.

The basic idea for this kind of reasoning can be found in [3] and [4]. The main idea is to compare two main measures of the progress, dm_k and $\varepsilon_\delta^{N_k}(x_k)$, and to keep them close to each other.

Let us consider dm_k as the benchmark. If $dm_k < d\ \varepsilon_\delta^{N_k}(x_k)$, we say that $\varepsilon_\delta^{N_k}(x_k)$ is too large or that we have a lack of precision. That implies that the confidence interval is too wide and we are trying to narrow it down by increasing the sample size and therefore reducing the error made by approximation (4.2). On the other hand, in order to work with a sample size as small as possible, if $dm_k > d\ \varepsilon_\delta^{N_k}(x_k)$ we deduce that it is not necessary to have that much precision and we are trying to reduce the sample size.

On the other hand, if we set the lack of precision as the benchmark, we have the following reasoning. If the reduction measure dm_k is too small in comparison with $\varepsilon_\delta^{N_k}(x_k)$, we say that there is not much that can be done for the function $\hat{f}_{N_k}$ in the sense of decreasing its value and we move on to the next level, trying to get closer to the final objective function $\hat{f}_{N_{max}}$ if possible.

The previously described mechanism provides us with the candidate for the upcoming sample size. Before accepting it, we have one more test. First of all, if the precision is increased, that is if $N_k \leq N_k^+$, we continue with $N_{k+1} = N_k^+$. However, if we have the signal that we

should decrease the sample size, i.e. if $N_k^+ < N_k$, then we compare the reduction that is already obtained using the current step s_k and the sample size N_k with the reduction this step would provide if the sample size was N_k^+. In order to do that, we compute ρ_k using (4.9). If ρ_k is relatively large, we do not approve the reduction because these two functions are too different and we choose to work with more precision and therefore put $N_{k+1} = N_k$. More formally, the algorithm is described as follows. Notice that it provides

$$N_{k+1} \geq N_k^+.$$

ALGORITHM 3

S0 Input parameters: $N_k^+,\ N_k,\ x_k,\ x_{k+1}$.

S1 Determine N_{k+1}

1) If $N_k^+ \geqslant N_k$ then $N_{k+1} = N_k^+$.

2) If $N_k^+ < N_k$ compute

$$\rho_k = \left| \frac{\hat{f}_{N_k^+}(x_k) - \hat{f}_{N_k^+}(x_{k+1})}{\hat{f}_{N_k}(x_k) - \hat{f}_{N_k}(x_{k+1})} - 1 \right|.$$

i) If $\rho_k < \frac{N_k - N_k^+}{N_k}$ put $N_{k+1} = N_k^+$.

ii) If $\rho_k \geq \frac{N_k - N_k^+}{N_k}$ put $N_{k+1} = N_k$.

As it was already explained, this safeguard algorithm is supposed to prohibit an unproductive decrease in the sample size. However, the right-hand side of the previous inequality implies that if the proposed decrease $N_k - N_k^+$ is relatively large, then the chances for accepting the smaller sample size are larger. This reasoning is supported by numerical testings because the large decrease in the sample size was

almost always productive. On the other hand, we are more rigorous if, for example, $N_k = N_{max}$ and $N_k^+ = N_{max} - 1$.

Various other reasonings are possible. As it will be clear after the convergence analysis in chapters 5 and 6, the only thing that matters is that the relation $N_{k+1} \geq N_k^+$ is satisfied. For example, instead of (4.9) we could compare the sample variance, i.e.

$$\rho_k = \left| \frac{\hat{\sigma}^2_{N_k^+}(x_{k+1})}{\hat{\sigma}^2_{N_k}(x_{k+1})} - 1 \right|.$$

However, this definition was not that successful in practical implementations that we considered. The one that provided good results was the following. Instead of (4.9), we can define

$$\rho_k = \frac{\hat{f}_{N_k^+}(x_k) - \hat{f}_{N_k^+}(x_{k+1})}{\hat{f}_{N_k}(x_k) - \hat{f}_{N_k}(x_{k+1})}$$

and forbid the decrease if $\rho_k < \eta_0$ where η_0 is a fixed parameter smaller than 1. Although the reasoning is not that clear as for Algorithm 3, the results were highly competitive.

Now we will describe how to update the lower bound N_k^{min}.

- If $N_{k+1} \leq N_k$ then $N_{k+1}^{min} = N_k^{min}$.
- If $N_{k+1} > N_k$ and
  - N_{k+1} is a sample size which has not been used so far then $N_{k+1}^{min} = N_k^{min}$.
  - N_{k+1} is a sample size which had been used and we have made a big enough decrease of the function $\hat{f}_{N_{k+1}}$, then $N_{k+1}^{min} = N_k^{min}$.

– N_{k+1} is a sample size which had been used and we have not made a big enough decrease of the function $\hat{f}_{N_{k+1}}$, then $N_{k+1}^{min} = N_{k+1}$.

We say that we have not made big enough decrease of the function $\hat{f}_{N_{k+1}}$ if the following inequality is true

$$\hat{f}_{N_{k+1}}(x_{h(k)}) - \hat{f}_{N_{k+1}}(x_{k+1}) < \frac{N_{k+1}}{N_{max}}(k+1-h(k))\varepsilon_{\delta}^{N_{k+1}}(x_{k+1}),$$

where $h(k)$ is the iteration at which we started to use the sample size N_{k+1} for the last time. For example, if $k = 7$ and $(N_0, ..., N_8) = (3, 6, 6, 4, \mathbf{6}, 6, 3, 3, 6)$, then $N_k = 3$, $N_{k+1} = 6$ and $h(k) = 4$. So, the idea is that if we come back to some sample size N_{k+1} that we had already used and if, since then, we have not done much in order to decrease the value of $\hat{f}_{N_{k+1}}$ we choose not to go below that sample size anymore, i.e. we put it as the lower bound. Notice that if we rearrange the previous inequality, we obtain the average decrease of the function $\hat{f}_{N_{k+1}}$ since the iteration $h(k)$ on the left-hand side

$$\frac{\hat{f}_{N_{k+1}}(x_{h(k)}) - \hat{f}_{N_{k+1}}(x_{k+1})}{(k+1-h(k))} < \frac{N_{k+1}}{N_{max}}\varepsilon_{\delta}^{N_{k+1}}(x_{k+1}).$$

The decrease is compared to the lack of precision throughout the ratio N_{k+1}/N_{max}. This means that we are requesting the stronger decrease if the function $\hat{f}_{N_{k+1}}$ is closer to $\hat{f}_{N_{max}}$. That way we point out that we are not that interested in what is happening with the function that is far away from the objective one. However, using some positive constant instead of the ratio N_{k+1}/N_{max} is also an option [3]. At the end, notice that the sequence of the sample size lower bounds is nondecreasing.

4.3 Convergence analysis

This section is devoted to the convergence results for Algorithm 1. The following important lemma states that after a finite number of iterations the sample size N_{max} is reached and kept until the end.

Lemma 4.3.1 *Suppose that the assumptions A1 and A3 are true. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$. Then, either Algorithm 1 terminates after a finite number of iterations with $N_k = N_{max}$ or there exists $q \in \mathbb{N}$ such that for every $k \geq q$ the sample size is $N_k = N_{max}$.*

Proof. First of all, recall that Algorithm 1 terminates only if $\|\nabla \hat{f}_{N_k}(x_k)\| = 0$ and $N_k = N_{max}$. Therefore, we will consider the case where the number of iterations is infinite. Again, notice that Algorithm 3 implies that $N_{k+1} \geq N_k^+$ is true for every k. Now, let us prove that sample size can not be stacked at a size that is lower than the maximal one.

Suppose that there exists $\tilde{n} > n_1$ such that for every $k \geq \tilde{n}$ $N_k = N^1 < N_{max}$. We have already explained that step S3 of Algorithm 1 provides the decreasing search direction p_k at every iteration. Therefore, denoting $g_k^{N_k} = \nabla \hat{f}_{N_k}(x_k)$, we know that for every $k \geq \tilde{n}$

$$\hat{f}_{N^1}(x_{k+1}) \leq \hat{f}_{N^1}(x_k) + \eta \alpha_k (g_k^{N^1})^T p_k,$$

i.e., for every $s \in \mathbb{N}$

$$\hat{f}_{N^1}(x_{\tilde{n}+s}) \leq \hat{f}_{N^1}(x_{\tilde{n}+s-1}) + \eta \alpha_{\tilde{n}+s-1} (g_{\tilde{n}+s-1}^{N^1})^T p_{\tilde{n}+s-1} \leq \ldots$$

$$\leq \hat{f}_{N^1}(x_{\tilde{n}}) + \eta \sum_{j=0}^{s-1} \alpha_{\tilde{n}+j} (g_{\tilde{n}+j}^{N^1})^T p_{\tilde{n}+j}. \tag{4.11}$$

Now, from (4.11) and Lemma 4.1.2 we know that

$$-\eta \sum_{j=0}^{s-1} \alpha_{\tilde{n}+j} (g_{\tilde{n}+j}^{N^1})^T p_{\tilde{n}+j} \leq \hat{f}_{N^1}(x_{\tilde{n}}) - \hat{f}_{N^1}(x_{\tilde{n}+s}) \leq \hat{f}_{N^1}(x_{\tilde{n}}) - M_F. \tag{4.12}$$

The inequality (4.12) is true for every s so

$$0 \leq \sum_{j=0}^{\infty} -\alpha_{\tilde{n}+j} (g_{\tilde{n}+j}^{N^1})^T p_{\tilde{n}+j} \leq \frac{\hat{f}_{N^1}(x_{\tilde{n}}) - M_F}{\eta} := C.$$

Therefore

$$\lim_{j \to \infty} -\alpha_{\tilde{n}+j} (\nabla \hat{f}_{N^1}(x_{\tilde{n}+j}))^T p_{\tilde{n}+j} = 0. \tag{4.13}$$

Let us consider the Algorithm 2 and iterations $k > \tilde{n}$. The possible scenarios are the following.

1) $dm_k = d\ \varepsilon_\delta^{N_k}(x_k)$. This implies

$$-\alpha_k (g_k^{N_k})^T p_k = d\ \varepsilon_\delta^{N_k}(x_k) \geq d\ \kappa$$

2) $dm_k > d\ \varepsilon_\delta^{N_k}(x_k)$. This implies

$$-\alpha_k (g_k^{N_k})^T p_k > d\ \varepsilon_\delta^{N_k}(x_k) \geq d\ \kappa$$

3) $dm_k < d\ \varepsilon_\delta^{N_k}(x_k)$ and $dm_k \geq \nu_1 d\ \varepsilon_\delta^{N_k}(x_k)$. In this case we have

$$-\alpha_k (g_k^{N_k})^T p_k \geq \nu_1 d\ \varepsilon_\delta^{N_k}(x_k) \geq \nu_1 d\ \kappa$$

4) The case $dm_k < \nu_1 d\ \varepsilon_\delta^{N_k}(x_k)$ is impossible because it would yield $N_{k+1} \geq N_k^+ = N_{max} > N^1$.

Therefore, in every possible case we know that for every $k > \tilde{n}$

$$-\alpha_k (g_k^{N^1})^T p_k \geq \kappa d \, \nu_1 := \tilde{C} > 0$$

and therefore

$$\liminf_{k \to \infty} -\alpha_k (g_k^{N^1})^T p_k \geq \tilde{C} > 0,$$

which is in contradiction with (4.13).

We have just proved that sample size can not stay on $N^1 < N_{max}$. Therefore, the remaining two possible scenarios are as follows:

L1 There exists $\tilde{n}$ such that $N_k = N_{max}$ for every $k \geq \tilde{n}$.

L2 The sequence of sample sizes oscillates.

Let us consider the scenario L2. Suppose that there exists $\bar{k}$ such that $N_{\bar{k}}^{min} = N_{max}$. Since the sequence of sample size lower bounds $\{N_k^{min}\}_{k \in \mathbb{N}}$ is nondecreasing, this would imply that $N_k^{min} = N_{max}$ for every $k \geq \bar{k}$. But this implies $N_k = N_{max}$ for every $k > \bar{k}$, i.e. we obtain the scenario L1 where the sample size can not oscillate. Therefore, if we consider the scenario L2, we know that for every k

$$N_k^{min} < N_{max}.$$

This means that N_k^{min} increases only at finitely many iterations. Recall that the sample size lower bound increases only when $N_{k+1}^{min} = N_{k+1}$. Then we have

$$N_{k+1}^{min} = N_{k+1} > N_k \geq N_{k-1}^{+} \geq N_{k-1}^{min}.$$

Notice that, according to the proposed mechanism for updating N_k^{min}, updating form $N_{k+1}^{min} = N_{k+1}$ happens only if N_{k+1} is the sample size which had been used already, $N_{k+1} > N_k$ and the obtained decrease in $\hat{f}_{N_{k+1}}$ was not good enough. Therefore, we conclude that there exists an iteration r_1 such that for every $k \geq r_1$ we have one of the following scenarios:

M1 $N_{k+1} \leq N_k$

M2 $N_{k+1} > N_k$ and we have enough decrease in $\hat{f}_{N_{k+1}}$

M3 $N_{k+1} > N_k$ and we did not use the sample size N_{k+1} before

Now, let $\bar{N}$ be the maximal sample size that is used at infinitely many iterations. Furthermore, define the set of iterations $\bar{K}_0$ at which sample size changes to $\bar{N}$. The definition of $\bar{N}$ implies that there exists iteration r_2 such that for every $k \in \bar{K}_0$, $k \geq r_2$ the sample size is increased to $\bar{N}$, i.e.

$$N_k < N_{k+1} = \bar{N}.$$

Define $r = \max\{r_1, r_2\}$ and set $\bar{K} = \bar{K}_0 \bigcap \{r, r+1, \ldots\}$. Clearly, each iteration in $\bar{K}$ excludes the scenario M1. Moreover, taking out the first member of a sequence $\bar{K}$ and retaining the same notation for the remaining sequence we can exclude the scenario M3 as well. This leaves us with M2 as the only possible scenario for iterations in $\bar{K}$. Therefore, for every $k \in \bar{K}$ the following is true

$$\hat{f}_{\bar{N}}(x_{h(k)}) - \hat{f}_{\bar{N}}(x_{k+1}) \geq \frac{N_{k+1}}{N_{max}}(k + 1 - h(k))\varepsilon_{\delta}^{\bar{N}}(x_{k+1}).$$

Now, defining the set of iterations $K_1 = \bar{K} \bigcap \{n_1, n_1 + 1, \ldots\}$ we can say that for every $k \in K_1$ we have

$$\hat{f}_{\bar{N}}(x_{h(k)}) - \hat{f}_{\bar{N}}(x_{k+1}) \geq \frac{N_{k+1}}{N_{max}}\kappa \geq \frac{1}{N_{max}}\kappa = \bar{C} > 0.$$

Recall that $h(k)$ defines the iteration at which we started to use the sample size $\bar{N}$ for the last time before the iteration $k+1$. Therefore, the previous inequality implies that we have reduced the function $\hat{f}_{\bar{N}}$ for the positive constant $\bar{C}$ infinitely many times, which is in contradiction with Lemma 4.1.2. From everything above, we conclude that the only

possible scenario is in fact L1, i.e. there exists iteration $\tilde{n}$ such that for every $k \geq \tilde{n}, \quad N_k = N_{max}$. ■

Now, we prove the main result. Before we state the theorem, we will make one more assumption about the search direction.

B 1 *The sequence of directions p_k is bounded and satisfies the following implication:*

$$\lim_{k \in K} p_k^T \nabla \hat{f}_{N_k}(x_k) = 0 \ \Rightarrow \ \lim_{k \in K} \nabla \hat{f}_{N_k}(x_k) = 0,$$

for any subset of iterations K.

This assumption allow us to consider the general descent direction but it is obviously satisfied for $p_k = -\nabla \hat{f}_{N_k}(x_k)$. Furthermore quasi-Newton directions also satisfy the assumption under the standard conditions for such methods such as uniform boundedness of the inverse Hessian approximation.

Theorem 4.3.1 *Suppose that the assumptions A1, A3 and B1 are true. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$ and that the sequence $\{x_k\}_{k \in \mathbb{N}}$ generated by Algorithm 1 is bounded. Then, either Algorithm 1 terminates after a finite number of iterations at a stationary point of function $\hat{f}_{N_{max}}$ or every accumulation point of the sequence $\{x_k\}_{k \in \mathbb{N}}$ is a stationary point of $\hat{f}_{N_{max}}$.*

Proof. First of all, recall that Algorithm 1 terminates only if $\|\nabla \hat{f}_{N_{max}}(x_k)\| = 0$, that is if the point x_k is stationary for the function $\hat{f}_{N_{max}}$. Therefore, we consider the case where the number of iterations is infinite. In that case, the construction of Algorithm 1 provides a decreasing search direction at every iteration. Furthermore, Lemma

4.3.1 implies the existence of iteration $\hat{n}$ such that for every $k \geq \hat{n}$ $N_k = N_{max}$ and

$$\hat{f}_{N_{max}}(x_{k+1}) \leq \hat{f}_{N_{max}}(x_k) + \eta\alpha_k(g_k^{N_{max}})^T p_k,$$

where $g_k^{N_{max}} = \nabla \hat{f}_{N_{max}}(x_k)$. Equivalently, for every $s \in \mathbb{N}$

$$\hat{f}_{N_{max}}(x_{\hat{n}+s}) \leq \hat{f}_{N_{max}}(x_{\hat{n}+s-1}) + \eta\alpha_{\hat{n}+s-1}(g_{\hat{n}+s-1}^{N_{max}})^T p_{\hat{n}+s-1} \leq ...$$

$$\leq \hat{f}_{N_{max}}(x_{\hat{n}}) + \eta \sum_{j=0}^{s-1} \alpha_{\hat{n}+j}(g_{\hat{n}+j}^{N_{max}})^T p_{\hat{n}+j}.$$

Again, this inequality and Lemma 4.1.2 imply

$$-\eta \sum_{j=0}^{s-1} \alpha_{\hat{n}+j}(g_{\hat{n}+j}^{N_{max}})^T p_{\hat{n}+j} \leq \hat{f}_{N_{max}}(x_{\hat{n}}) - \hat{f}_{N_{max}}(x_{\hat{n}+s}) \leq \hat{f}_{N_{max}}(x_{\hat{n}}) - M_F.$$

This is true for every $s \in \mathbb{N}$, therefore

$$0 \leq \sum_{j=0}^{\infty} -\alpha_{\hat{n}+j}(g_{\hat{n}+j}^{N_{max}})^T p_{\hat{n}+j} \leq \frac{\hat{f}_{N_{max}}(x_{\hat{n}}) - M_F}{\eta} := C.$$

This inequality implies

$$\lim_{k\to\infty} \alpha_k(\nabla \hat{f}_{N_{max}}(x_k))^T p_k = 0. \tag{4.14}$$

Now, let x^* be an arbitrary accumulation point of sequence of iterations $\{x_k\}_{k\in\mathbb{N}}$, i.e. let K be the subset $K \subseteq \mathbb{N}$ such that

$$\lim_{k\in K} x_k = x^*.$$

If the sequence of step sizes $\{\alpha_k\}_{k\in K}$ is bounded from below, i.e. if there exists $\hat{\alpha} > 0$ such that $\alpha_k \geq \hat{\alpha}$ for every $k \in K$ sufficiently large, then (4.14) implies

$$\lim_{k\in K} (\nabla \hat{f}_{N_{max}}(x_k))^T p_k = 0.$$

This result, together with assumption B1 and Lemma 4.1.1, implies

$$\nabla \hat{f}_{N_{max}}(x^*) = \lim_{k \in K} \nabla \hat{f}_{N_{max}}(x_k) = 0.$$

Now, suppose that there exists a subset $K_1 \subseteq K$ such that $\lim_{k \in K_1} \alpha_k = 0$. This implies the existence of $\hat{k}$ such that for every $k \in K_2 = K_1 \cap \{\max\{\hat{n}, \hat{k}\}, \max\{\hat{n}, \hat{k}\} + 1, ...\}$ the step size α_k that satisfies the Armijo condition (4.6) is smaller than 1. That means that for every $k \in K_2$ there exists α'_k such that $\alpha_k = \beta \alpha'_k$ and

$$\hat{f}_{N_{max}}(x_k + \alpha'_k p_k) > \hat{f}_{N_{max}}(x_k) + \eta \alpha'_k (\nabla \hat{f}_{N_{max}}(x_k))^T p_k,$$

which is equivalent to

$$\frac{\hat{f}_{N_{max}}(x_k + \alpha'_k p_k) - \hat{f}_{N_{max}}(x_k)}{\alpha'_k} > \eta (\nabla \hat{f}_{N_{max}}(x_k))^T p_k.$$

By Mean Value Theorem there exists $t_k \in [0, 1]$ such that previous inequality is equivalent to

$$p_k^T \nabla \hat{f}_{N_{max}}(x_k + t_k \alpha'_k p_k) > \eta (\nabla \hat{f}_{N_{max}}(x_k))^T p_k. \tag{4.15}$$

Notice that $\lim_{k \in K_2} \alpha'_k = 0$ and recall that the sequence of search directions is assumed to be bounded. Therefore, there exists p^* and subset $K_3 \subseteq K_2$ such that $\lim_{k \in K_3} p_k = p^*$. Now, taking limit in (4.15) and using Lemma 4.1.1, we obtain

$$(\nabla \hat{f}_{N_{max}}(x^*))^T p^* \geq \eta (\nabla \hat{f}_{N_{max}}(x^*))^T p^*. \tag{4.16}$$

On the other hand, we know that $\eta \in (0, 1)$ and p_k is the descent search direction, i.e. $(\nabla \hat{f}_{N_{max}}(x_k))^T p_k < 0$ for every $k \in K_3$. This implies that

$$(\nabla \hat{f}_{N_{max}}(x^*))^T p^* \leq 0.$$

The previous inequality and (4.16) imply that

$$\lim_{k\in K_3} (\nabla \hat{f}_{N_{max}}(x_k))^T p_k = (\nabla \hat{f}_{N_{max}}(x^*))^T p^* = 0.$$

Finally, according to assumption B1,

$$\nabla \hat{f}_{N_{max}}(x^*) = \lim_{k\in K_3} \nabla \hat{f}_{N_{max}}(x_k) = 0.$$

which completes the proof. ■

Chapter 5

Nonmonotone line search with variable sample size

In the previous chapter we introduced the strategy that allows us to vary the sample size during the optimization process. The main goal of this strategy is to save some function evaluations and make the optimization process less expensive. The Armijo rule has been the main tool for providing the decrease that ensures the global convergence under few additional conditions. Backtracking technique has been proposed in order to impose the condition of sufficient decrease. More precisely, the step size has been decreased until a sufficient decrease is attained and every trial cost us the number of function evaluations which is equal to the current sample size. The question is: Can we reduce the number of trials and still provide global convergence of the algorithm? The answer lies in nonmonotone line search techniques. The idea is strongly correlated to the variable sample size philosophy - we do not want to impose strict conditions in early iterations when we are, most probably, far away from a solution which we are searching for. Another strong motivation for using the nonmonotone line search is coming from an environment where the search direc-

tion is not necessary the descent one. This happens, for instance, when derivatives are not affordable. This scenario is very realistic in stochastic optimization framework where input-output information is very often the only thing we can count on. In this case, it is useful to consider nonmonotone rules which do not require the decrease in objective function at every iteration. Moreover, when it comes to global convergence, there is at least one more thing that goes in favor of nonmonotone techniques. Namely, numerical results suggest that nonmonotone techniques have more chances of finding global optimum than their monotone counterparts [71], [17], [53].

Having all this in mind, we will introduce algorithms that use nonmonotone line search rules which are adopted into the variable sample size framework. The main difference regarding the previously stated Algorithm 1 is in the step where the line search is performed. The nonmonotone line search rules we use here are described in section 5.1. Section 5.2 is devoted to the convergence analysis with a general (possibly nondescent) search direction. Section 5.3 deals with the descent search direction convergence analysis and the R-linear convergence rate (Krejić, Krklec Jerinkić [38]).

5.1 The algorithm and the line search

The problem that we are observing is the same as in the monotone line search framework. We consider

$$\min_{x\in\mathbb{R}^n} \hat{f}_{N_{max}}(x),$$

where N_{max} is some substantially large but finite positive integer. Just like in the previous chapter, the sample size is allowed to vary across iterations and therefore we are observing different functions $\hat{f}_{N_k}$ during the optimization process. Recall that in Chapter 4 the search direction

p_k is assumed to be decreasing for $\hat{f}_{N_k}$ and the line search was in the form of the standard Armijo rule

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \hat{f}_{N_k}(x_k) + \eta \alpha_k p_k^T \nabla \hat{f}_{N_k}(x_k).$$

In order to enlarge the set of problems on which Algorithm 1 can be applied, we will generalize the previously stated condition and write it in a slightly different manner.

Consider the line search that seeks for a step size α_k that satisfies the condition

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \varepsilon_k - \eta dm_k(\alpha_k), \tag{5.1}$$

where the parameters $\tilde{C}_k$, ε_k and η and function $dm_k(\alpha_k)$ are to be explained as follows.

Let us first consider the measure of decrease represented by the function $dm_k(\alpha)$. The main property that dm_k has to posses is positivity. Considering this function, we will consider two main cases. The first one is

$$dm_k(\alpha) = -\alpha p_k^T \nabla \hat{f}_{N_k}(x_k). \tag{5.2}$$

This definition is used only if $p_k^T \nabla \hat{f}_{N_k}(x_k) < 0$, i.e. if the search direction is decreasing one. Only if this is true, we will have the desired property $dm_k(\alpha) > 0$ for every $\alpha > 0$. This definition of dm_k will usually be used with parameter $\eta \in (0, 1)$ like in the standard Armijo line search. The second option is to put

$$dm_k(\alpha) = \alpha^2 \beta_k, \tag{5.3}$$

where β_k is a positive number which belongs to a sequence that satisfies the following assumption.

C 2 *$\{\beta_k\}_{k \in \mathbb{N}}$ is a bounded sequence of positive numbers with the property*

$$\lim_{k \in K} \beta_k = 0 \Rightarrow \lim_{k \in K} \nabla \hat{f}_{N_{max}}(x_k) = 0,$$

for every infinite subset of indices $K \subseteq \mathbb{N}$.

This kind of sequence is introduced in [23] and it is described in more detail in section 2.3. The previous assumption is crucial for proving the convergence result. Recall that besides some increasingly accurate approximation of $\|\nabla \hat{f}_{N_{max}}(x_k)\|$, a suitable choice for β_k can even be some positive constant, for example $\beta_k = 1$. Moreover, for simplicity and without loss of generality we can set $\eta = 1$ when the definition (5.3) is considered. In that case, any positive constant is also a valid choice for η since it can be viewed as the part of the sequence of β_k. More precisely, we can consider $\eta\beta_k$ instead of β_k because it does not affect the previously stated assumption.

One of the main things that makes line search (5.1) well defined even for nondescent search direction is the parameter ε_k. This parameter is given by the sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$ usually defined at the beginning of the optimization process. We state the following assumption with a remark that if we are not able to ensure the descent search direction, ε_k is assumed to be be positive.

C 3 *$\{\varepsilon_k\}_{k\in\mathbb{N}}$ is a sequence of nonnegative numbers such that* $\sum_{k=0}^{\infty} \varepsilon_k = \varepsilon < \infty$.

Finally, let us comment the parameters $\tilde{C}_k$ mentioned in the line search (5.1). The motivation for introducing this parameter comes from Zhang, Hager [71] where C_k is a convex combination of objective function values in the previous iterations. In that paper, the descent search directions are considered and the line search

$$f(x_k + \alpha_k p_k) \leq C_k + \eta\alpha_k p_k^T \nabla f(x_k)$$

is well defined since it is proved that $C_k \geq f(x_k)$ for every k where f is the objective function. However, we are dealing with a different function at every iteration and $C_k \geq \hat{f}_{N_k}(x_k)$ needs not to be true. In order to make our algorithm well defined, we need an additional safeguard. We define

$$\tilde{C}_k = \max\{C_k, \hat{f}_{N_k}(x_k)\}. \tag{5.4}$$

That way we ensure $\tilde{C}_k \geq \hat{f}_{N_k}(x_k)$. Definition of C_k is conceptually the same like in the deterministic case but it is slightly modified to fit the variable sample size scheme. Therefore, we define C_k recursively with

$$C_{k+1} = \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} C_k + \frac{1}{Q_{k+1}} \hat{f}_{N_{k+1}}(x_{k+1}), \quad C_0 = \hat{f}_{N_0}(x_0), \tag{5.5}$$

where

$$Q_{k+1} = \tilde{\eta}_k Q_k + 1, \quad Q_0 = 1, \quad \tilde{\eta}_k \in [0, 1]. \tag{5.6}$$

Parameter $\tilde{\eta}_k$ determines the level of monotonicity regarding C_k. Notice that $\tilde{\eta}_{k-1} = 0$ yields $\tilde{C}_k = C_k = \hat{f}_{N_k}(x_k)$. On the other hand, $\tilde{\eta}_k = 1$ for every k treats all previous function values equally yielding the average

$$C_k = \frac{1}{k+1} \sum_{i=0}^{k} \hat{f}_{N_i}(x_i). \tag{5.7}$$

In order to cover all the relevant nonmonotone line search rules, we will let $\tilde{C}_k$ be defined in the following manner as well. Instead of (5.4), we can consider

$$\tilde{C}_k = \max\{\hat{f}_{N_k}(x_k), \ldots, \hat{f}_{N_{\max\{k-M+1,0\}}}(x_{\max\{k-M+1,0\}})\}, \tag{5.8}$$

where $M \in \mathbb{N}$ is arbitrary but fixed. This way, we are trying to decrease the maximal value of the response function in the previous M iterations. The similar rule can be found in [23] for example.

Now, we will state lemmas considering Q_k and C_k. The following result can also be found in [71].

Lemma 5.1.1 *Suppose that Q_k is defined by (5.6). Then for every $k \in \mathbb{N}_0$*

$$1 \leq Q_k \leq k + 1. \tag{5.9}$$

Proof. The proof will be conducted by induction. Since $Q_0 = 1$ by definition, (5.9) is true for $k = 0$. Furthermore, since $\tilde{\eta}_0 \in [0, 1]$ it follows $\tilde{\eta}_0 Q_0 \geq 0$ and it is easy to see that $Q_1 \geq 1$. On the other hand, $Q_1 = \tilde{\eta}_0 Q_0 + 1 \leq Q_0 + 1 = 2$. Now, suppose that $1 \leq Q_k \leq k + 1$ is true. This inequality together with $\tilde{\eta}_k \in [0, 1]$ imply that $Q_{k+1} = \tilde{\eta}_k Q_k + 1 \leq Q_k + 1 \leq k + 2$. Moreover, assumptions $\tilde{\eta}_k \geq 0$ and $Q_k \geq 1$ also imply that $Q_{k+1} \geq 1$. ■

It is stated in [71] that C_k is a convex combination of previous function values where a fixed, deterministic function is considered throughout the iterations. The next lemma is a generalization of that result since we have the sequence of different functions $\hat{f}_{N_k}$.

Lemma 5.1.2 *Suppose that C_k is defined by (5.5) and Q_k is defined by (5.6). Then for every $k \in \mathbb{N}_0$, C_k is a convex combination of $\hat{f}_{N_0}(x_0), ..., \hat{f}_{N_k}(x_k)$.*

Proof. For $k = 0$, the statement obviously holds. Further, the definition of C_0 and previous lemma imply

$$C_1 = \frac{\tilde{\eta}_0 Q_0}{Q_1} C_0 + \frac{1}{Q_1} \hat{f}_{N_1}(x_1) = \frac{\tilde{\eta}_0 Q_0}{\tilde{\eta}_0 Q_0 + 1} \hat{f}_{N_0}(x_0) + \frac{1}{\tilde{\eta}_0 Q_0 + 1} \hat{f}_{N_1}(x_1)$$

which is a convex combination of $\hat{f}_{N_0}(x_0)$ and $\hat{f}_{N_1}(x_1)$. Now, suppose that $C_k = \sum_{i=0}^{k} \alpha_i^k \hat{f}_{N_i}(x_i)$ where $\sum_{i=0}^{k} \alpha_i^k = 1$ and $\alpha_i^k \geq 0$ for $i = 0, 1, ..., k$. Let us prove that this implies that C_{k+1} is a convex combination of $\hat{f}_{N_0}(x_0), ..., \hat{f}_{N_{k+1}}(x_{k+1})$. By definition (5.5) we have

$$C_{k+1} = \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} \sum_{i=0}^{k} \alpha_i^k \hat{f}_{N_i}(x_i) + \frac{1}{Q_{k+1}} \hat{f}_{N_{k+1}}(x_{k+1}) = \sum_{i=0}^{k+1} \alpha_i^{k+1} \hat{f}_{N_i}(x_i),$$

where

$$\alpha_{k+1}^{k+1} = \frac{1}{Q_{k+1}} \quad \text{and} \quad \alpha_i^{k+1} = \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} \alpha_i^k \quad \text{for} \quad i = 0, 1, ..., k.$$

Obviously, $\alpha_i^{k+1} \geq 0$ for $i = 0, 1, ..., k+1$. Since $Q_{k+1} = \tilde{\eta}_k Q_k + 1$ we have

$$\sum_{i=0}^{k+1} \alpha_i^{k+1} = \sum_{i=0}^{k} \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} \alpha_i^k + \frac{1}{Q_{k+1}} = \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} \sum_{i=0}^{k} \alpha_i^k + \frac{1}{Q_{k+1}} = \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} + \frac{1}{Q_{k+1}} = 1.$$

Finally, by induction we conclude that C_k is a convex combination of $\hat{f}_{N_0}(x_0), ..., \hat{f}_{N_k}(x_k)$ for every $k \in \mathbb{N}_0$. ■

Lemma 5.1.3 *Suppose that the assumptions of Lemma 5.1.2 hold. Then for every* $k \in \mathbb{N}_0$

1) if the assumption A3 holds, $M_F \leq C_k$.

2) if the assumption A4 holds, $C_k \leq M_{FF}$.

Proof. According to Lemma 5.1.2, we know that $C_k = \sum_{i=0}^{k} \alpha_i^k \hat{f}_{N_i}(x_i)$ where $\sum_{i=0}^{k} \alpha_i^k = 1$ and $\alpha_i^k \geq 0$ for $i = 0, 1, ..., k$. If the assumption A3 holds, then Lemma 4.1.2 implies that $M_F \leq \hat{f}_{N_i}(x_i)$ for every i and

$$C_k \geq \sum_{i=0}^{k} \alpha_i^k M_F = M_F.$$

On the other hand, if assumption A4 holds Lemma 4.1.3 imply that $\hat{f}_{N_i}(x_i) \leq M_{FF}$ for every i. Therefore,

$$C_k \leq \sum_{i=0}^{k} \alpha_i^k M_{FF} = M_{FF}.$$

■

The following technical lemma is significant for the convergence analysis. It distinguishes two cases regarding C_k or more precisely regarding $\tilde{\eta}_k$. It turns out that average (5.7) is the special case in terms of the convergence analysis too. The consequence of the following lemma is that we obtain a stronger result by excluding $\tilde{\eta}_k = 1$.

Lemma 5.1.4 *Suppose that $\tilde{\eta}_k \in [\eta_{min}, \eta_{max}]$ for every k where $0 \leq \eta_{min} \leq \eta_{max} \leq 1$ and Q_k is defined by (5.6).*

1) If $\eta_{min} = 1$, then $\lim_{k\to\infty} Q_k^{-1} = 0$.

2) If $\eta_{max} < 1$, then $\lim_{k\to\infty} Q_k^{-1} > 0$.

Proof. In the case where $\eta_{min} = 1$, it follows that $Q_k = k + 1$ for every $k \in \mathbb{N}$ and therefore $\lim_{k\to\infty} Q_k^{-1} = 0$, thus the result holds. Now, let us consider the second case, i.e. let us suppose that $\eta_{max} < 1$. First, we will show that

$$0 \leq Q_k \leq \sum_{i=0}^{k} \eta_{max}^i \tag{5.10}$$

for every $k \in \mathbb{N}_0$. Nonnegativity of Q_k has already been discussed. The other inequality needs to be proved. This will be done by induction. Of course, $Q_0 = 1 = \eta_{max}^0$ and $Q_1 = \tilde{\eta}_0 Q_0 + 1 \leq \eta_{max} + \eta_{max}^0$. If we suppose $Q_k \leq \sum_{i=0}^{k} \eta_{max}^i$ is true, then

$$Q_{k+1} = \tilde{\eta}_k Q_k + 1 \leq \tilde{\eta}_k \sum_{i=0}^{k} \eta_{max}^i + 1 \leq \sum_{i=0}^{k} \eta_{max}^{i+1} + \eta_{max}^0 = \sum_{i=0}^{k+1} \eta_{max}^i.$$

Therefore, (5.10) is true for every $k \in \mathbb{N}_0$.

Now, since $\eta_{max} < 1$, we know that for every $k \in \mathbb{N}_0$

$$0 \leq Q_k \leq \sum_{i=0}^{k} \eta_{max}^i \leq \sum_{i=0}^{\infty} \eta_{max}^i = \frac{1}{1 - \eta_{max}}$$

which furthermore implies

$$\lim_{k\to\infty} Q_k^{-1} \geq 1 - \eta_{max} > 0.$$

■

Now, we can state the main algorithm of this chapter. The additional algorithms are already stated in the previous chapter and they are not going to be changed. The main modifications are in steps S4-S6 of Algorithm 1. As it is mentioned at the beginning of this chapter, search direction needs not to be decreasing in general and the line search rule is changed. Consequently, the definition of dm_k is altered and therefore the input parameter of Algorithm 2 is modified, but the mechanism for searching N_k^+ remains the same. Another important difference between Algorithm 1 and Algorithm 4 is that the new one does not have a stopping criterion. This is because, in general, we do not have the exact gradient of the function $\hat{f}_{N_{max}}$.

ALGORITHM 4

S0 Input parameters: $M, N_{max}, N_0^{min} \in \mathbb{N}$, $x_0 \in \mathbb{R}^n$, $\delta, \beta, \nu_1 \in (0,1)$, $\eta \in (0,1]$, $0 \leq \eta_{min} \leq \eta_{max} \leq 1$, $\{\varepsilon_k\}_{k\in\mathbb{N}}$ satisfying assumption C3.

S1 Generate the sample realization: $\xi_1, \ldots, \xi_{N_{max}}$.
Set $N_0 = N_0^{min}$, $C_0 = \hat{f}_{N_0}(x_0)$, $Q_0 = 1$, $\tilde{C}_0 = C_0$, $k = 0$.

S2 Compute $\hat{f}_{N_k}(x_k)$ using (4.2).

S3 Compute $\varepsilon_\delta^{N_k}(x_k)$ using (4.8).

S4 Determine the search direction p_k.

S5 Find the smallest nonnegative integer j such that $\alpha_k = \beta^j$ satisfies

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \varepsilon_k - \eta dm_k(\alpha_k).$$

S6 Set $s_k = \alpha_k p_k$ and $x_{k+1} = x_k + s_k$.

S7 Determine the candidate sample size N_k^+ using Algorithm 2 and $dm_k = dm_k(\alpha_k)$.

S8 Determine the sample size N_{k+1} using Algorithm 3.

S9 Determine the lower bound of sample size N_{k+1}^{min}.

S10 Determine $\tilde{C}_{k+1}$ using (5.4) or (5.8).

S11 Set $k = k + 1$ and go to step S2.

5.2 General search direction

In this section, we analyze the case where the search direction might be nondescent. Just like in the previous chapter, the convergence analysis is conducted in two main stages. First, we prove that Algorithm 4 uses $N_k = N_{max}$ for every k large enough. The second part then relies on the deterministic algorithm analysis applied on the function $\hat{f}_{N_{max}}$. In order to prove that the sample size eventually becomes N_{max}, we need to prove that a subsequence of $\{dm_k(\alpha_k)\}_{k\in\mathbb{N}}$ tends to zero. This is done by considering two definitions of $\tilde{C}_k$ separately. Results regarding the line search with $\tilde{C}_k = \max\{C_k, \hat{f}_{N_k}(x_k)\}$ are stated in the following two lemmas.

Lemma 5.2.1 *Consider the Algorithm 4 with $\tilde{C}_k$ defined by (5.4). Suppose that the assumptions A1 and A3 are satisfied and there exists $\tilde{n} \in \mathbb{N}$ such that $N_k = N$ for every $k \geq \tilde{n}$. Then for every $k \geq \tilde{n}$*

$$\tilde{C}_{k+1} \leq \tilde{C}_k + \varepsilon_k - \eta \frac{dm_k}{Q_{k+1}} \tag{5.11}$$

and for every $s \in \mathbb{N}$

$$\tilde{C}_{\tilde{n}+s} \leq \tilde{C}_{\tilde{n}} + \sum_{j=0}^{s-1} \varepsilon_{\tilde{n}+j} - \eta \sum_{j=0}^{s-1} \frac{dm_{\tilde{n}+j}}{Q_{\tilde{n}+j+1}}. \tag{5.12}$$

Proof. First of all, recall that the line search implies that for every $k \geq \tilde{n}$

$$\hat{f}_N(x_{k+1}) \leq \tilde{C}_k + \varepsilon_k - \eta dm_k \tag{5.13}$$

where $dm_k = dm_k(\alpha_k)$. Furthermore, $C_k \leq \max\{C_k, \hat{f}_{N_k}(x_k)\} = \tilde{C}_k$. Therefore, the following is true for every $k \geq \tilde{n}$

$$\begin{aligned} C_{k+1} &= \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} C_k + \frac{1}{Q_{k+1}} \hat{f}_N(x_{k+1}) \\ &\leq \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} \tilde{C}_k + \frac{1}{Q_{k+1}} (\tilde{C}_k + \varepsilon_k - \eta dm_k) \\ &= \tilde{C}_k + \frac{\varepsilon_k}{Q_{k+1}} - \eta \frac{dm_k}{Q_{k+1}} \end{aligned}$$

The last equality is a consequence of the definition $Q_{k+1} = \tilde{\eta}_k Q_k + 1$. Moreover, Lemma 5.1.1 implies that $Q_{k+1} \geq 1$ and we obtain

$$C_{k+1} \leq \tilde{C}_k + \varepsilon_k - \eta \frac{dm_k}{Q_{k+1}}. \tag{5.14}$$

Now, (5.13) and (5.14) imply

$$\begin{aligned} \tilde{C}_{k+1} &= \max\{C_{k+1}, \hat{f}_N(x_{k+1})\} \\ &\leq \max\{\tilde{C}_k + \varepsilon_k - \eta \frac{dm_k}{Q_{k+1}}, \tilde{C}_k + \varepsilon_k - \eta dm_k\} \\ &= \tilde{C}_k + \varepsilon_k - \eta \frac{dm_k}{Q_{k+1}}. \end{aligned}$$

Furthermore, by using the induction argument we can conclude that the previous inequality implies that (5.12) holds for every $s \in \mathbb{N}$. ■

Lemma 5.2.2 *Suppose that the assumptions A1 and A3 are satisfied and there exists $\tilde{n} \in \mathbb{N}$ such that $N_k = N$ for every $k \geq \tilde{n}$. Then Algorithm 4 with $\tilde{C}_k$ defined by (5.4) satisfies*

$$\liminf_{k \to \infty} dm_k(\alpha_k) = 0.$$

Moreover, if $\eta_{max} < 1$ *it follows that*

$$\lim_{k\to\infty} dm_k(\alpha_k) = 0.$$

Proof. The assumptions of the previous lemma are satisfied, therefore we know that (5.12) holds for every $s \in \mathbb{N}$. Recall that the sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$ satisfies the assumption C3 by definition of Algorithm 4. Furthermore, by using the result of Lemma 4.1.2 we obtain

$$0 \leq \eta \sum_{j=0}^{s-1} \frac{dm_{\tilde{n}+j}}{Q_{\tilde{n}+j+1}} \leq \tilde{C}_{\tilde{n}} - M_F + \varepsilon := C.$$

Now, letting s tend to infinity we obtain

$$0 \leq \sum_{j=0}^{\infty} \frac{dm_{\tilde{n}+j}}{Q_{\tilde{n}+j+1}} \leq \frac{C}{\eta} < \infty. \tag{5.15}$$

If we suppose that $dm_k \geq \bar{d} > 0$ for all k sufficiently large, we would obtain the contradiction with (5.15) because Lemma 5.1.1 would imply

$$\sum_{j=0}^{\infty} \frac{dm_{\tilde{n}+j}}{Q_{\tilde{n}+j+1}} \geq \sum_{j=0}^{\infty} \frac{\bar{d}}{\tilde{n}+j+2} = \infty.$$

Therefore, there must exist a subset of iterations K such that $\lim_{k\in K} dm_k = 0$ and the statement follows.

Finally, assume that $\eta_{max} < 1$. From (5.15) we conclude that $\lim_{k\to\infty} Q_k^{-1} dm_k = 0$. Since Lemma 5.1.4 implies that $\lim_{k\to\infty} Q_k^{-1} > 0$ it follows that $\lim_{k\to\infty} dm_k = 0$. This completes the proof. ■

Next, we consider an analogous statement for

$$\tilde{C}_k = \max\{\hat{f}_{N_k}(x_k), \ldots, \hat{f}_{N_{\max\{k-M+1,0\}}}(x_{\max\{k-M+1,0\}})\}.$$

The result stated in previous lemma concerning $\eta_{max} < 1$ is attainable in this case as well, but under stronger assumptions on the search directions and the objective function. However, the existence of a subsequence of $\{dm_k(\alpha_k)\}_{k\in\mathbb{N}}$ that vanishes will be enough to prove the first stage result in the convergence analysis. The proof of the following lemma relies on the proof of the Proposition 1 from [23].

Lemma 5.2.3 *Suppose that the assumptions A1 and A3 are satisfied and there exists $\tilde{n} \in \mathbb{N}$ such that $N_k = N$ for every $k \geq \tilde{n}$. Then Algorithm 4 with $\tilde{C}_k$ defined by (5.8) satisfies*

$$\liminf_{k\to\infty} dm_k(\alpha_k) = 0.$$

Proof. The assumptions of this lemma imply the existence of $\tilde{n}$ such that $N_k = N$ for every $k \geq \tilde{n}$. Without loss of generality, we can assume that $\tilde{n} \geq M$ where M defines the number of previous iterations that are considered in (5.8). Therefore, after a finite number of iterations the function $\hat{f}_N$ is considered. If we define $dm_k = dm_k(\alpha_k)$ then the line search rule implies that for every $k \geq \tilde{n}$

$$\hat{f}_N(x_{k+1}) \leq \tilde{C}_k + \varepsilon_k - \eta dm_k.$$

Furthermore, if we define $s(k) = \tilde{n} + kM$ then by the definition of $\tilde{C}_k$ we have

$$\tilde{C}_{s(k)} = \max\{\hat{f}_N(x_{s(k)}), \ldots, \hat{f}_N(x_{s(k)-M+1})\}.$$

Finally, let $v(k)$ be the index such that $\tilde{C}_{s(k)} = \hat{f}_N(x_{v(k)})$ and notice that $v(k) \in \{s(k-1)+1, \ldots, s(k-1)+M\}$.

The next step is to prove that for every $k \in \mathbb{N}$ and every $j \in \{1, 2, \ldots, M\}$ the following holds

$$\hat{f}_N(x_{s(k)+j}) \leq \tilde{C}_{s(k)} + \sum_{i=0}^{j-1} \varepsilon_{s(k)+i} - \eta dm_{s(k)+j-1}. \tag{5.16}$$

This will be proved by induction. For $j = 1$ the result follows from the line search rule directly

$$\hat{f}_N(x_{s(k)+1}) \leq \tilde{C}_{s(k)} + \varepsilon_{s(k)} - \eta dm_{s(k)}.$$

Suppose that (5.16) holds for every $1 < j < M$. Since ε_k and ηdm_k are nonnegative it follows that

$$\hat{f}_N(x_{s(k)+j}) \leq \tilde{C}_{s(k)} + \sum_{i=0}^{j-1} \varepsilon_{s(k)+i} \leq \tilde{C}_{s(k)} + \sum_{i=0}^{M-1} \varepsilon_{s(k)+i}$$

for every $1 \leq j < M$. Finally, for $j + 1$ we obtain

$$\begin{aligned}
\hat{f}_N(x_{s(k)+j+1}) &\leq \tilde{C}_{s(k)+j} + \varepsilon_{s(k)+j} - \eta dm_{s(k)+j} \\
&\leq \max\{\hat{f}_N(x_{s(k)+j}), \ldots, \hat{f}_N(x_{s(k)+1}), \tilde{C}_{s(k)}\} \\
&+ \varepsilon_{s(k)+j} - \eta dm_{s(k)+j} \\
&\leq \tilde{C}_{s(k)} + \sum_{i=0}^{j-1} \varepsilon_{s(k)+i} + \varepsilon_{s(k)+j} - \eta dm_{s(k)+j} \\
&= \tilde{C}_{s(k)} + \sum_{i=0}^{j} \varepsilon_{s(k)+i} - \eta dm_{s(k)+j}.
\end{aligned}$$

Therefore, (5.16) holds. Moreover,

$$\hat{f}_N(x_{s(k)+j}) \leq \tilde{C}_{s(k)} + \sum_{i=0}^{M-1} \varepsilon_{s(k)+i} - \eta dm_{s(k)+j-1}. \tag{5.17}$$

For $\tilde{C}_{s(k+1)}$ we have

$$\tilde{C}_{s(k+1)} = \max\{\hat{f}_N(x_{s(k)+M}), \ldots, \hat{f}_N(x_{s(k)+1})\} = \hat{f}_N(x_{v(k+1)}).$$

Furthermore, using the previous equality and (5.17) we conclude that

$$\tilde{C}_{s(k+1)} \leq \tilde{C}_{s(k)} + \sum_{i=0}^{M-1} \varepsilon_{s(k)+i} - \eta dm_{v(k+1)-1}. \tag{5.18}$$

The previous inequality holds for every $k \in \mathbb{N}$. If we sum up these inequalities, for arbitrary $m \in \mathbb{N}$ we obtain

$$\tilde{C}_{s(m+1)} \leq \tilde{C}_{s(1)} + \sum_{k=1}^{m} \sum_{i=0}^{M-1} \varepsilon_{s(k)+i} - \eta \sum_{k=1}^{m} dm_{v(k+1)-1}. \tag{5.19}$$

By definition of $\tilde{C}_k$ and Lemma 4.1.2 it follows that $\tilde{C}_k \geq M_F$ for every $k \in \mathbb{N}$. Moreover, assumption C3 implies that the following inequality holds for every $m \in \mathbb{N}$

$$\sum_{k=1}^{m} \sum_{i=0}^{M-1} \varepsilon_{s(k)+i} \leq \varepsilon < \infty. \tag{5.20}$$

Having all this in mind, from (5.19) follows that for every $m \in \mathbb{N}$

$$0 \leq \eta \sum_{k=1}^{m} dm_{v(k+1)-1} \leq \tilde{C}_{s(1)} + \varepsilon - M_F < \infty.$$

Finally, letting m tend to infinity we obtain

$$\lim_{k\to\infty} dm_{v(k)-1} = 0,$$

which is equivalent to $\liminf_{k\to\infty} dm_k(\alpha_k) = 0$. ■

The previous two lemmas imply the following result concerning Algorithm 4.

Corollary 5.2.1 *Suppose that the assumptions A1 and A3 are satisfied and there exists $\tilde{n} \in \mathbb{N}$ such that $N_k = N$ for every $k \geq \tilde{n}$. Then*

$$\liminf_{k\to\infty} dm_k(\alpha_k) = 0.$$

Lemma 5.2.4 *Suppose that the assumptions A1 and A3 are satisfied. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$. Then there exists $q \in \mathbb{N}$ such that for every $k \geq q$ the sample size used by Algorithm 4 is maximal, i.e. $N_k = N_{max}$.*

Proof. First of all, recall that Algorithm 4 does not have a stopping criterion and the number of iterations is infinite by default. Notice that Algorithm 3 implies that $N_{k+1} \geq N_k^+$ is true for every k. Now, let us prove that sample size can not be stacked at a size that is lower than the maximal one.

Suppose that there exists $\tilde{n} > n_1$ such that for every $k \geq \tilde{n} \quad N_k = N^1 < N_{max}$ and define $dm_k = dm_k(\alpha_k)$. In that case, Corollary 5.2.1 implies that $\liminf_{k\to\infty} dm_k = 0$. On the other hand, we have that $\varepsilon_\delta^{N^1}(x_k) \geq \kappa > 0$ for every $k \geq \tilde{n}$ which means that $\nu_1 d\ \varepsilon_\delta^{N_k}(x_k)$ is bounded from below for every k sufficiently large. Therefore, there exists at least one $p \geq \tilde{n}$ such that $dm_p < \nu_1 d\ \varepsilon_\delta^{N^1}(x_p)$. However, the construction of Algorithm 2 would then imply $N_p^+ = N_{max}$ and we would have $N_{p+1} = N_{max} > N^1$ which is in contradiction with the current assumption that sample size stays at N^1.

We have just proved that sample size can not stay on $N^1 < N_{max}$. Therefore, the remaining two possible scenarios are as follows:

L1 There exists $\tilde{n}$ such that $N_k = N_{max}$ for every $k \geq \tilde{n}$.

L2 The sequence of sample sizes oscillates.

Let us consider the scenario L2. Suppose that there exists $\bar{k}$ such that $N_{\bar{k}}^{min} = N_{max}$. Since the sequence of sample size lower bounds $\{N_k^{min}\}_{k\in\mathbb{N}}$ is nondecreasing, this would imply that $N_k^{min} = N_{max}$ for every $k \geq \bar{k}$. But this implies $N_k = N_{max}$ for every $k > \bar{k}$, i.e. we obtain the scenario L1 where the sample size can not oscillate. Therefore, if we consider the scenario L2, we know that for every k

$$N_k^{min} < N_{max}.$$

This means that N_k^{min} increases only at finitely many iterations. Recall that the sample size lower bound increases only when $N_{k+1}^{min} = N_{k+1}$. Then we have

$$N_{k+1}^{min} = N_{k+1} > N_k \geq N_{k-1}^{+} \geq N_{k-1}^{min}.$$

Notice that, according to the proposed mechanism for updating N_k^{min}, updating form $N_{k+1}^{min} = N_{k+1}$ happens only if N_{k+1} is the sample size which had been used already, $N_{k+1} > N_k$ and the obtained decrease in $\hat{f}_{N_{k+1}}$ was not good enough. Therefore, we conclude that there exists an iteration r_1 such that for every $k \geq r_1$ we have one of the following scenarios:

M1 $N_{k+1} \leq N_k$

M2 $N_{k+1} > N_k$ and we have enough decrease in $\hat{f}_{N_{k+1}}$

M3 $N_{k+1} > N_k$ and we did not use the sample size N_{k+1} before

Now, let $\bar{N}$ be the maximal sample size that is used at infinitely many iterations. Furthermore, define the set of iterations $\bar{K}_0$ at which sample size changes to $\bar{N}$. The definition of $\bar{N}$ implies that there exists iteration r_2 such that for every $k \in \bar{K}_0$, $k \geq r_2$ the sample size is increased to $\bar{N}$, i.e.

$$N_k < N_{k+1} = \bar{N}.$$

Define $r = \max\{r_1, r_2\}$ and set $\bar{K} = \bar{K}_0 \bigcap \{r, r+1, \ldots\}$. Clearly, each iteration in $\bar{K}$ excludes the scenario M1. Moreover, taking out the first member of a sequence $\bar{K}$ and retaining the same notation for the remaining sequence we can exclude the scenario M3 as well. This leaves us with M2 as the only possible scenario for iterations in $\bar{K}$. Therefore, for every $k \in \bar{K}$ the following is true

$$\hat{f}_{\bar{N}}(x_{h(k)}) - \hat{f}_{\bar{N}}(x_{k+1}) \geq \frac{N_{k+1}}{N_{max}}(k + 1 - h(k))\varepsilon_{\delta}^{\bar{N}}(x_{k+1}).$$

Now, defining the set of iterations $K_1 = \bar{K} \bigcap \{n_1, n_1 + 1, \ldots\}$ we can say that for every $k \in K_1$ we have

$$\hat{f}_{\bar{N}}(x_{h(k)}) - \hat{f}_{\bar{N}}(x_{k+1}) \geq \frac{N_{k+1}}{N_{max}}\kappa \geq \frac{1}{N_{max}}\kappa = \bar{C} > 0.$$

Recall that $h(k)$ defines the iteration at which we started to use the sample size $\bar{N}$ for the last time before the iteration $k + 1$. Therefore, previous inequality implies that we have reduced the function $\hat{f}_{\bar{N}}$ for the positive constant $\bar{C}$ infinitely many times, which is in contradiction with Lemma 4.1.2. From everything above, we conclude that the only possible scenario is in fact L1, i.e. there exists iteration $\tilde{n}$ such that for every $k \geq \tilde{n}$, $\quad N_k = N_{max}$. ■

At the beginning of this chapter we introduced two possibilities for dm_k. In this section general search direction is considered. Therefore, we consider the case where dm_k is defined by (5.3), i.e. where $dm_k(\alpha) = \alpha^2\beta_k$ and the line search is

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \varepsilon_k - \alpha_k^2\beta_k. \tag{5.21}$$

In the following theorems we state the convergence result concerning Algorithm 4 with this line search rule.

Theorem 5.2.1 *Suppose that the assumptions A1, A3 and C2 are satisfied. Furthermore, suppose that there exist a positive constant* κ *and* $n_1 \in \mathbb{N}$ *such that* $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ *for every* $k \geq n_1$ *and that the sequences of search directions* $\{p_k\}_{k\in\mathbb{N}}$ *and iterates* $\{x_k\}_{k\in\mathbb{N}}$ *of Algorithm 4 with the line search rule (5.21) are bounded. Then there exists an accumulation point* (x^*, p^*) *of the sequence* $\{(x_k, p_k)\}_{k\in\mathbb{N}}$ *that satisfies the following inequality*

$$p^{*T}\nabla \hat{f}_{N_{max}}(x^*) \geq 0.$$

Proof. Notice that under these assumptions, Lemma 5.2.4 implies the existence of $\tilde{n} \in \mathbb{N}$ such that $N_k = N_{max}$ for every $k \geq \tilde{n}$. Moreover, Corollary 5.2.1 then implies that there exists a subset $K_0 \subseteq \mathbb{N}$ such that $\lim_{k\in K_0} dm_k(\alpha_k) = \lim_{k\in K_0} \alpha_k^2\beta_k = 0$. Furthermore, since $\{(x_k, p_k)\}_{k\in\mathbb{N}}$ is bounded, there exists at least one subset $K \subseteq K_0$ and points x^* and p^* such that $\lim_{k\in K} x_k = x^*$ and $\lim_{k\in K} p_k = p^*$. Therefore it follows that

$$\lim_{k\in K} \alpha_k^2\beta_k = 0. \tag{5.22}$$

Suppose that a subsequence of the step sizes $\{\alpha_k\}_{k\in K}$ is bounded from below. In that case (5.22) implies $\lim_{k\in K} \beta_k = 0$ which together with the assumption C2 yields $\lim_{k\in K} \nabla \hat{f}_{N_{max}}(x_k) = 0$. Furthermore, Lemma 4.1.1 implies $\nabla \hat{f}_{N_{max}}(x^*) = 0$ which is even stronger result than the one we want to prove.

Now, let us consider the remaining case. Suppose that there exists a subset $K_1 \subseteq K$ such that $\lim_{k\in K_1} \alpha_k = 0$. This and the backtracking rule that we use in our algorithm imply the existence of $\hat{k} \in \mathbb{N}$ such that for every $k \geq \hat{k}, k \in K_1$ the step size α_k that satisfies condition (5.21) is smaller than 1. That means that for every $k \geq \max\{\hat{k}, \tilde{n}\}, k \in K_1$ there exists α_k' such that $\alpha_k = \beta\alpha_k'$ and

$$\hat{f}_{N_{max}}(x_k + \alpha_k' p_k) > \tilde{C}_k + \varepsilon_k - (\alpha_k')^2\beta_k \geq \hat{f}_{N_{max}}(x_k) - (\alpha_k')^2\beta_k,$$

which is equivalent to

$$\frac{\hat{f}_{N_{max}}(x_k + \alpha'_k p_k) - \hat{f}_{N_{max}}(x_k)}{\alpha'_k} > -\alpha'_k \beta_k.$$

By the Mean Value Theorem there exists $t_k \in [0,1]$ such that

$$p_k^T \nabla \hat{f}_{N_{max}}(x_k + t_k \alpha'_k p_k) > -\alpha'_k \beta_k,$$

Now,

$$\begin{aligned} -\alpha'_k \beta_k &< p_k^T \nabla \hat{f}_{N_{max}}(x_k + t_k \alpha'_k p_k) \\ &= p_k^T (\nabla \hat{f}_{N_{max}}(x_k + t_k \alpha'_k p_k) - \nabla \hat{f}_{N_{max}}(x_k)) + p_k^T \nabla \hat{f}_{N_{max}}(x_k) \\ &\leq \|p_k\| \|\nabla \hat{f}_{N_{max}}(x_k + t_k \alpha'_k p_k) - \nabla \hat{f}_{N_{max}}(x_k)\| + p_k^T \nabla \hat{f}_{N_{max}}(x_k). \end{aligned}$$

Assumptions of this theorem imply that the sequences $\{p_k\}_{k\in\mathbb{N}}$ and $\{\beta_k\}_{k\in\mathbb{N}}$ are bounded. Furthermore, $\lim_{k\in K_1} \alpha_k = 0$ implies $\lim_{k\in K_1} \alpha'_k = 0$ and Lemma 4.1.1 implies the continuity of the gradient $\nabla \hat{f}_{N_{max}}$. Therefore the previous inequality obviously yields

$$\lim_{k\in K_1} p_k^T \nabla \hat{f}_{N_{max}}(x_k) \geq 0,$$

which together with the fact that $K_1 \subseteq K$ implies the result

$$p^{*T} \nabla \hat{f}_{N_{max}}(x^*) \geq 0.$$

■

The following theorem provides a bit stronger result since under the stated assumptions we are able to prove that the whole sequence of $dm_k(\alpha_k)$ tends to zero.

Theorem 5.2.2 *Suppose that the assumptions of Theorem 5.2.1 are satisfied and that the line search rule uses $\tilde{C}_k$ defined by (5.4) with $\eta_{max} < 1$. Then every accumulation point (x^*, p^*) of the sequence $\{(x_k, p_k)\}_{k\in\mathbb{N}}$ satisfies the following inequality*

$$p^{*T} \nabla \hat{f}_{N_{max}}(x^*) \geq 0.$$

Proof. First of all, we know that under these assumptions Lemma 5.2.4 implies the existence of $\tilde{n} \in \mathbb{N}$ such that $N_k = N_{max}$ for every $k \geq \tilde{n}$. Furthermore, since we have that $\eta_{max} < 1$, Lemma 5.2.2 implies

$$\lim_{k\to\infty} \alpha_k^2 \beta_k = 0.$$

Now, let (x^*, p^*) be an arbitrary accumulation point of the sequence $\{(x_k, p_k)\}_{k\in\mathbb{N}}$ and define $K \subseteq \mathbb{N}$ such that $\lim_{k\in K}(x_k, p_k) = (x^*, p^*)$. Then (5.22) is true and the rest of the proof is as in Theorem 5.2.1. ■

Roughly speaking, the previous two theorems give the conditions under which we would encounter the point where descent search direction is unattainable. In order to make these points stationary for function $\hat{f}_{N_{max}}$, we need additional assumptions considering search directions p_k. One of them is already stated in previous chapter and the other one is as follows.

B 2 *Search directions p_k satisfy the following condition*

$$\lim_{k\to\infty} p_k^T \nabla \hat{f}_{N_{max}}(x_k) \leq 0.$$

Notice that this assumption is satisfied if we are able to produce descent search directions eventually. For example, it will be satisfied if increasingly accurate finite differences are used to approximate the gradient, or more precisely if

$$(p_k)_i = -\frac{\hat{f}_{N_k}(x_k + h_k e_i) - \hat{f}_{N_k}(x_k - h_k e_i)}{2h_k}, \quad i = 1, \ldots n$$

with $h_k \to 0$ when $k \to \infty$.

Theorem 5.2.3 *Suppose that the assumptions of Theorem 5.2.1 are satisfied and that the search directions satisfy the assumptions B1 and B2. Then there exists an accumulation point of $\{x_k\}_{k\in\mathbb{N}}$ which is stationary for the function $\hat{f}_{N_{max}}$.*

Proof. Theorem 5.2.1 implies the existence of an accumulation point (x^*, p^*) of the sequence $\{(x_k, p_k)\}_{k\in\mathbb{N}}$ that satisfies

$$p^{*T}\nabla \hat{f}_{N_{max}}(x^*) \geq 0. \tag{5.23}$$

Let $K \subseteq \mathbb{N}$ be the subset of indices such that $\lim_{k\in K}(x_k, p_k) = (x^*, p^*)$. Since the search directions are bounded by assumption B1 and $\nabla \hat{f}_{N_{max}}$ is continuous as a consequence of Lemma 4.1.1, assumption B2 implies that

$$p^{*T}\nabla \hat{f}_{N_{max}}(x^*) = \lim_{k\in K} p_k^T \nabla \hat{f}_{N_{max}}(x_k) \leq 0$$

which together with (5.23) gives the following result

$$p^{*T}\nabla \hat{f}_{N_{max}}(x^*) = 0. \tag{5.24}$$

Finally, assumption B1 implies that $\nabla \hat{f}_{N_{max}}(x^*) = 0$. ■

The assumptions B1 and B2 in combination with the assumptions of Theorem 5.2.2 provide a stronger result.

Theorem 5.2.4 *Suppose that the assumptions of Theorem 5.2.2 are satisfied and that the search directions satisfy the assumptions B1 and B2. Then every accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$ is stationary for $\hat{f}_{N_{max}}$.*

Proof. Let x^* be an arbitrary accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$ and let K be the subset of indices such that $\lim_{k\in K}(x_k, p_k) = (x^*, p^*)$. Then Theorem 5.2.2 implies that

$$p^{*T}\nabla \hat{f}_{N_{max}}(x^*) \geq 0. \tag{5.25}$$

Since the search directions are bounded, we can always find such subset K. The rest of the proof is as in Theorem 5.2.3. We obtain $\nabla \hat{f}_{N_{max}}(x^*) = 0$ and since x^* is arbitrary, the result follows. ■

Notice that nonmonotone line search rules proposed in this section yield the same result regarding achievement of the maximal sample size N_{max} as in the previous chapter. Therefore, the convergence results again rely on the deterministic analysis applied to the function $\hat{f}_{N_{max}}$. The main result is the existence of an accumulation point which is stationary for $\hat{f}_{N_{max}}$ without imposing the assumption of descent search directions. Moreover, if the parameter $\tilde{C}_k$ is defined by (5.4) with $\eta_{max} < 1$, every accumulation point is stationary under the same assumptions.

5.3 Descent search direction

This section is devoted to the case where the exact gradient of function $\hat{f}_{N_k}$ is available and the descent search direction is used at every iteration. In that case, ε_k needs not to be positive to ensure that the line search is well defined. However, setting $\varepsilon_k > 0$ can be beneficial for the algorithm performance since it increases the chances for larger step sizes. Furthermore, we define $dm_k(\alpha)$ by (5.2) as it is proposed earlier. Two possibilities for $\tilde{C}_k$ divide the analysis in two parts.

First, we consider $\tilde{C}_k$ defined by (5.4). This leads us to the line search defined by

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \varepsilon_k + \eta \alpha_k p_k^T \nabla \hat{f}_{N_k}(x_k),$$
$$\tilde{C}_k = \max\{C_k, \hat{f}_{N_k}(x_k)\}. \tag{5.26}$$

This framework yields the possibility for obtaining the convergence result where every accumulation point is stationary for the relevant objective function. Moreover, the R-linear rate of convergence is attainable if we assume that ε_k tends to zero R-linearly.

The conditions for the global convergence are stated in the following theorem. The convergence result again depends on the choice of parameter η_{max}, i.e. eliminating $\eta_{max} = 1$ provides a stronger result.

Theorem 5.3.1 *Suppose that the assumptions A1, A3 and B1 hold. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$ and that the sequence $\{x_k\}_{k\in\mathbb{N}}$ generated by Algorithm 4 with the line search (5.26) and descent search directions $\{p_k\}_{k\in\mathbb{N}}$ is bounded. Then, there exist a subsequence of iterates $\{x_k\}_{k\in\mathbb{N}}$ that tends to a stationary point of $\hat{f}_{N_{max}}$. Moreover, if $\eta_{max} < 1$ then every accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$ is a stationary point of $\hat{f}_{N_{max}}$.*

Proof. First, notice that under the previously stated assumptions Lemma 5.2.4 implies the existence of $\hat{n} \in \mathbb{N}$ such that $N_k = N_{max}$ for every $k \geq \hat{n}$. Furthermore, Lemma 5.2.2 implies that there exists $K_0 \subseteq \mathbb{N}$ such that

$$\lim_{k\in K_0} \alpha_k p_k^T \nabla \hat{f}_{N_{max}}(x_k) = \liminf_{k\to\infty} -dm_k(\alpha_k) = 0. \tag{5.27}$$

Since the iterates of Algorithm 4 are assumed to be bounded, there exists at least one accumulation point x^* of sequence $\{x_k\}_{k\in K_0}$. Therefore, there exists $K \subseteq K_0$ such that

$$\lim_{k\in K} x_k = x^*.$$

If the sequence of step sizes is bounded from below, then the result follows from Theorem 5.2.3 and Theorem 5.2.4 because this is the special case of the line search considered in the previous section with

$$\beta_k = -\frac{\eta p_k^T \nabla \hat{f}_{N_{max}}(x_k)}{\alpha_k}.$$

Notice that this sequence satisfies the assumption C2 if α_k is bounded from below since p_k and $\nabla \hat{f}_{N_{max}}(x_k)$ are bounded by the assumptions of this theorem. More precisely, search directions are bounded by the assumption B1 and the boundedness of the gradient follows from

boundedness of $\{x_k\}_{k\in\mathbb{N}}$ and the assumptions A1. Moreover, the assumption B2 is obviously satisfied for the descent search directions.

Now, suppose that there exists a subset $K_1 \subseteq K$ such that $\lim_{k\in K_1} \alpha_k = 0$. This implies the existence of $\hat{k}$ such that for every $k \in K_2 = K_1 \cap \{\max\{\hat{n}, \hat{k}\}, \max\{\hat{n}, \hat{k}\} + 1, ...\}$ the step size α_k that satisfies the condition (5.26) is smaller than 1. That means that for every $k \in K_2$ there exists α'_k such that $\alpha_k = \beta\alpha'_k$ and

$$\hat{f}_{N_{max}}(x_k + \alpha'_k p_k) > \tilde{C}_k + \varepsilon_k + \eta\alpha'_k(\nabla\hat{f}_{N_{max}}(x_k))^T p_k.$$

Since $\tilde{C}_k \geq \hat{f}_{N_k}(x_k)$ by definition and $\varepsilon_k \geq 0$, we have that for every $k \in K_2$

$$\hat{f}_{N_{max}}(x_k + \alpha'_k p_k) > \hat{f}_{N_{max}}(x_k) + \eta\alpha'_k(\nabla\hat{f}_{N_{max}}(x_k))^T p_k,$$

which is equivalent to

$$\frac{\hat{f}_{N_{max}}(x_k + \alpha'_k p_k) - \hat{f}_{N_{max}}(x_k)}{\alpha'_k} > \eta(\nabla\hat{f}_{N_{max}}(x_k))^T p_k.$$

By the Mean Value Theorem there exists $t_k \in [0, 1]$ such that previous inequality is equivalent to

$$p_k^T \nabla\hat{f}_{N_{max}}(x_k + t_k\alpha'_k p_k) > \eta(\nabla\hat{f}_{N_{max}}(x_k))^T p_k. \tag{5.28}$$

Notice that $\lim_{k\in K_2} \alpha'_k = 0$ and recall that the sequence of search directions is assumed to be bounded. Therefore, there exist p^* and a subset $K_3 \subseteq K_2$ such that $\lim_{k\in K_3} p_k = p^*$. Now, taking limit in (5.28) and using Lemma 4.1.1, we obtain

$$(\nabla\hat{f}_{N_{max}}(x^*))^T p^* \geq \eta(\nabla\hat{f}_{N_{max}}(x^*))^T p^*. \tag{5.29}$$

On the other hand, we know that $\eta \in (0, 1)$ and p_k is a descent direction, i.e. $(\nabla\hat{f}_{N_{max}}(x_k))^T p_k < 0$ for every $k \in K_3$. This implies that

$$(\nabla\hat{f}_{N_{max}}(x^*))^T p^* \leq 0.$$

Previous inequality and (5.29) imply that

$$\lim_{k\in K_3}(\nabla \hat{f}_{N_{max}}(x_k))^T p_k = (\nabla \hat{f}_{N_{max}}(x^*))^T p^* = 0.$$

Again, according to assumption B1,

$$\nabla \hat{f}_{N_{max}}(x^*) = \lim_{k\in K_3} \nabla \hat{f}_{N_{max}}(x_k) = 0.$$

which completes this part of the proof.

At the end, let us consider the case where $\eta_{max} < 1$. Under this assumption the result of Lemma 5.2.2 implies

$$\lim_{k\to\infty} \alpha_k(\nabla \hat{f}_{N_{max}}(x_k))^T p_k = \lim_{k\to\infty} -dm_k = 0$$

instead of (5.27). Now, if we define x^* to be an arbitrary accumulation point of the sequence of iterates $\{x_k\}_{k\in\mathbb{N}}$, the rest of the proof is the same as in the first part and we obtain $\nabla \hat{f}_{N_{max}}(x^*) = 0$. Therefore, if $\eta_{max} < 1$ every accumulation point is stationary for $\hat{f}_{N_{max}}$. ■

After proving the global convergence result, we will analyze the convergence rate. Following the ideas from [71] and [17], we will prove that R-linear convergence for strongly convex functions can be obtained. Of course, some additional assumptions are needed. The first one is the strong convexity of the objective function.

A 5 *For every ξ, $F(\cdot, \xi)$ is a strongly convex function.*

The consequence of this assumption is that for every sample size N, $\hat{f}_N$ is a strongly convex function since it is defined by (4.2). Therefore, there exists $\gamma > 0$ such that for every N and every $x, y \in \mathbb{R}^n$

$$\hat{f}_N(x) \geq \hat{f}_N(y) + (\nabla \hat{f}_N(y))^T(x - y) + \frac{1}{2\gamma}\|x - y\|^2. \tag{5.30}$$

We continue with the analysis by proving the following lemma.

Lemma 5.3.1 *Suppose that the assumptions A1 and A5 are satisfied and x^* is an unique minimizer of the function $\hat{f}_N$. Then there exists positive constant γ such that for every $x \in \mathbb{R}^n$*

$$\frac{1}{2\gamma}\|x - x^*\|^2 \leq \hat{f}_N(x) - \hat{f}_N(x^*) \leq \gamma\|\nabla \hat{f}_N(x)\|^2$$

Proof. Since the function $\hat{f}_N$ is strongly convex, it has an unique minimizer x^* and we know that $\nabla \hat{f}_N(x^*) = 0$. Now, from (5.30) follows the existence of $\gamma > 0$ such that

$$\frac{1}{2\gamma}\|x - x^*\|^2 \leq \hat{f}_N(x^*) - \hat{f}_N(x) - (\nabla \hat{f}_N(x))^T(x^* - x)$$

and

$$\frac{1}{2\gamma}\|x - x^*\|^2 \leq \hat{f}_N(x) - \hat{f}_N(x^*).$$

If we sum up the previous two inequalities we obtain

$$\frac{1}{\gamma}\|x - x^*\|^2 \leq (\nabla \hat{f}_N(x))^T(x - x^*).$$

Furthermore, since $(\nabla \hat{f}_N(x))^T(x - x^*) \leq \|\nabla \hat{f}_N(x)\|\|x - x^*\|$, there follows

$$\|x - x^*\| \leq \gamma\|\nabla \hat{f}_N(x)\|. \tag{5.31}$$

Now, define $x(t) = x^* + t(x - x^*)$ for $t \in [0, 1]$ and let us consider the function $g(t) = \hat{f}_N(x(t))$. The function $g(t)$ is convex on $[0, 1]$ with the derivative $g'(t) = (\nabla \hat{f}_N(x(t)))^T(x - x^*)$. It has the unique minimizer $t = 0$ since $g'(0) = (\nabla \hat{f}_N(x^*))^T(x - x^*) = 0$. Furthermore, $g'(t)$ is increasing on $[0, 1]$ and

$$g'(t) \leq g'(1) = (\nabla \hat{f}_N(x))^T(x - x^*) \leq \|\nabla \hat{f}_N(x)\|\|(x - x^*)\|.$$

Now,

$$\begin{aligned}\hat{f}_N(x) - \hat{f}_N(x^*) &= \int_0^1 (\nabla \hat{f}_N(x^* + t(x - x^*)))^T (x - x^*) dt \\ &= \int_0^1 g'(t) dt \leq \|\nabla \hat{f}_N(x)\| \|x - x^*\| \\ &\leq \gamma \|\nabla \hat{f}_N(x)\|^2,\end{aligned}$$

where the last inequality comes from (5.40). ■

Now, we will prove that after a finite number of iterations, all the remaining iterates of the considered algorithm belong to a level set. This level set will not depend on the starting point x_0 as it is usual in deterministic framework, but on the point where the sample size becomes maximal and remains unchanged until the end of the optimization process.

Lemma 5.3.2 *Suppose that the assumptions of Lemma 5.2.4 are satisfied. Then there exists $q \in \mathbb{N}$ such that for every $k \geq q$ the iterate x_k belongs to the level set*

$$\mathcal{L} = \{x \in \mathbb{R}^n \mid \hat{f}_{N_{max}}(x) \leq \tilde{C}_q + \varepsilon\}. \tag{5.32}$$

Proof. Recall that Lemma 5.2.4 implies the existence of a finite number $\tilde{n}$ such that $N_k = N_{max}$ for every $k \geq \tilde{n}$. In that case, the assumptions of Lemma 5.2.1 are satisfied and we have that for every $s \in \mathbb{N}$ inequality (5.12) is true. Therefore, we conclude that for every $s \in \mathbb{N}$

$$\tilde{C}_{\tilde{n}+s} \leq \tilde{C}_{\tilde{n}} + \sum_{j=0}^{s-1} \varepsilon_{\tilde{n}+j} - \eta \sum_{j=0}^{s-1} \frac{dm_{\tilde{n}+j}}{Q_{\tilde{n}+j+1}} \leq \tilde{C}_{\tilde{n}} + \varepsilon.$$

Since $\hat{f}_{N_{max}}(x_{\tilde{n}+s}) \leq \tilde{C}_{\tilde{n}+s}$ by definition, we obtain that for every $k \geq \tilde{n}$

$$\hat{f}_{N_{max}}(x_k) \leq \tilde{C}_{\tilde{n}} + \varepsilon$$

which completes the proof. ■

This result is especially useful when a strongly convex function is considered. It is known that level sets of strongly convex functions are bounded. Therefore, we will have a bounded sequence of iterates which was the assumption in the previous analysis. In order to obtain R-linear convergence, we assume the Lipschitz continuity of the gradient function and impose the additional assumption on the search directions and on the sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$.

A 6 *Gradient $\nabla_x F(\cdot,\xi)$ is Lipschitz continuous on any bounded set.*

This assumption implies the Lipschitz continuity of the sample average function gradient $\nabla\hat{f}_N$ since

$$\begin{aligned}\|\nabla\hat{f}_N(x)-\nabla\hat{f}_N(y)\| &\leq \frac{1}{N}\sum_{i=1}^{N}\|\nabla_x F(x,\xi_i)-\nabla_x F(y,\xi_i)\|\\ &\leq \frac{1}{N}\sum_{i=1}^{N}L\|x-y\|\\ &= L\|x-y\|,\end{aligned}$$

where L is the Lipschitz constant of $\nabla_x F(x,\xi)$ on the relevant bounded set.

The following assumption on the directions is used in the deterministic case as a tool for proving the global convergence results [17], [71].

B 3 *There are positive constants c_1 and c_2 such that the search directions p_k satisfy*

$$p_k^T\nabla\hat{f}_{N_k}(x_k)\leq -c_1\|\nabla\hat{f}_{N_k}(x_k)\|^2$$

and

$$\|p_k\|\leq c_2\|\nabla\hat{f}_{N_k}(x_k)\|$$

for all k sufficiently large.

Under these assumptions, we can prove the following.

Theorem 5.3.2 *Suppose that the assumptions A1, A3, A5, A6, B1 and B3 are satisfied. Furthermore, suppose that there exist* $n_1 \in \mathbb{N}$ *and a positive constant* κ *such that* $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ *for every* $k \geq n_1$ *and that the sequence* $\{x_k\}_{k\in\mathbb{N}}$ *is generated by Algorithm 4 with the line search (5.26) and* $\eta_{max} < 1$. *Then there exist a constant* $\theta \in (0,1)$, *a finite number* q *and an unique minimizer* x^* *of the function* $\hat{f}_{N_{max}}$ *such that for every* $k \in \mathbb{N}$

$$\hat{f}_{N_{max}}(x_{q+k}) - \hat{f}_{N_{max}}(x^*) \leq \theta^k(\tilde{C}_q - \hat{f}_{N_{max}}(x^*)) + \sum_{j=1}^{k} \theta^{j-1}\varepsilon_{q+k-j}.$$

Proof. First, notice that the assumptions of this theorem imply the existence of a finite number $\tilde{n}$ such that $N_k = N_{max}$ for every $k \geq \tilde{n}$. Moreover, it follows that there exists a finite integer $q \geq \tilde{n}$ such that for every $k \geq q$ the iterate x_k belongs to the level set (5.32), i.e. $\hat{f}_{N_{max}}(x_k) \leq \tilde{C}_q + \varepsilon$. Furthermore, strong convexity of the function $\hat{f}_{N_{max}}$ implies the boundedness and convexity of that level set. Therefore, there exists at least one accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$. Under the assumption $\eta_{max} < 1$, Theorem 5.3.1 implies that every accumulation point of the iterative sequence is stationary for the function $\hat{f}_{N_{max}}$. On the other hand, strong convexity of the objective function implies that there is only one minimizer of the function $\hat{f}_{N_{max}}$ which is also the unique stationary point. Therefore, we can conclude that the whole sequence of iterates converges towards the unique stationary point of the function $\hat{f}_{N_{max}}$. Denoting that point by x^*, we have

$$\lim_{k\to\infty} x_k = x^*.$$

The assumptions of Lemma 5.2.1 are also satisfied and we know

that for every $k \geq q$

$$\tilde{C}_{k+1} \leq \tilde{C}_k + \varepsilon_k - \eta \frac{dm_k(\alpha_k)}{Q_{k+1}}.$$

Since we are assuming the descent search directions, we have

$$dm_k(\alpha_k) = -\alpha_k p_k^T \nabla \hat{f}_{N_{max}}(x_k)$$

for every $k \geq q$. Moreover, we have proved in Lemma 5.1.4 that $0 \leq Q_k \leq (1 - \eta_{max})^{-1}$ for every k. Therefore

$$\tilde{C}_{k+1} \leq \tilde{C}_k + \varepsilon_k - \eta(1 - \eta_{max}) dm_k(\alpha_k) \tag{5.33}$$

for every $k \geq q$.

The next step of this proof is to obtain the lower bound for the line search step size α_k for $k \geq q$. In order to do that, we will distinguish two types of iterations. The first type is when the full step is accepted, i.e. when $\alpha_k = 1$. The second one is when $\alpha_k < 1$. Then there exists $\alpha'_k = \alpha_k / \beta$ such that

$$\begin{aligned} \hat{f}_{N_{max}}(x_k + \alpha'_k p_k) &> \tilde{C}_k + \varepsilon_k + \eta \alpha'_k p_k^T \nabla \hat{f}_{N_{max}}(x_k) \\ &\geq \hat{f}_{N_{max}}(x_k) + \eta \alpha'_k p_k^T \nabla \hat{f}_{N_{max}}(x_k). \end{aligned}$$

On the other hand, the assumption A6 implies the Lipschitz continuity of the gradient $\nabla \hat{f}_{N_{max}}$ on $\{x \in \mathbb{R}^n | x = x_k + t p_k, t \in [0, 1], k \geq q\}$. Therefore, there exists a Lipschitz constant $L > 0$ such that the fol-

lowing holds

$$\begin{aligned}
\hat{f}_{N_{max}}(x_k+\alpha'_k p_k) &= \hat{f}_{N_{max}}(x_k)+\int_0^1 (\nabla \hat{f}_{N_{max}}(x_k+t\alpha'_k p_k))^T \alpha'_k p_k dt \\
&= \int_0^1 (\nabla \hat{f}_{N_{max}}(x_k+t\alpha'_k p_k)-\nabla \hat{f}_{N_{max}}(x_k))^T \alpha'_k p_k dt \\
&+ \hat{f}_{N_{max}}(x_k)+\alpha'_k(\nabla \hat{f}_{N_{max}}(x_k))^T p_k \\
&\leq \int_0^1 Lt(\alpha'_k)^2\|p_k\|^2 dt \\
&+ \hat{f}_{N_{max}}(x_k)+\alpha'_k(\nabla \hat{f}_{N_{max}}(x_k))^T p_k \\
&= \frac{L}{2}(\alpha'_k)^2\|p_k\|^2+\hat{f}_{N_{max}}(x_k)+\alpha'_k(\nabla \hat{f}_{N_{max}}(x_k))^T p_k.
\end{aligned}$$

Combining the previous two inequalities we obtain

$$\eta\alpha'_k p_k^T \nabla \hat{f}_{N_{max}}(x_k) < \frac{L}{2}(\alpha'_k)^2\|p_k\|^2+\alpha'_k(\nabla \hat{f}_{N_{max}}(x_k))^T p_k.$$

Dividing by α'_k and using the fact that $\alpha_k = \beta\alpha'_k$, by rearranging previous inequality we obtain

$$\alpha_k \geq \frac{-(\nabla \hat{f}_{N_{max}}(x_k))^T p_k 2\beta(1-\eta)}{L\|p_k\|^2}. \tag{5.34}$$

Furthermore, the assumption B3 implies the existence of a constant $c_1 > 0$ such that

$$-(\nabla \hat{f}_{N_{max}}(x_k))^T p_k \geq c_1\|\nabla \hat{f}_{N_{max}}(x_k)\|^2 \tag{5.35}$$

and we obtain

$$\alpha_k \geq \frac{c_1\|\nabla \hat{f}_{N_{max}}(x_k)\|^2 2\beta(1-\eta)}{L\|p_k\|^2}. \tag{5.36}$$

The assumption B3 implies the existence of a constant $c_2 > 0$ such that

$$\frac{\|\nabla \hat{f}_{N_{max}}(x_k)\|^2}{\|p_k\|^2} \geq \frac{1}{c_2^2}.$$

Putting the previous inequality in (5.36) we conclude that for every $k \geq q$

$$\alpha_k \geq \min\{1, \frac{c_1 2\beta(1-\eta)}{Lc_2^2}\}. \tag{5.37}$$

After obtaining the lower bound for the step size, we will prove that for every $k \geq q$

$$dm_k(\alpha_k) \geq \bar{\beta}_0 \|\nabla \hat{f}_{N_{max}}(x_k)\|^2 \tag{5.38}$$

where $\bar{\beta}_0 = \min\{c_1, \frac{c_1^2 2\beta(1-\eta)}{c_2^2 L}\}$. If $\alpha_k = 1$, it follows from (5.35) that

$$dm_k(\alpha_k) \geq c_1 \|\nabla \hat{f}_{N_{max}}(x_k)\|^2.$$

On the other hand, if $\alpha_k < 1$

$$\begin{aligned} dm_k(\alpha_k) &= -\alpha_k p_k^T \nabla \hat{f}_{N_{max}}(x_k) \\ &\geq c_1 \|\nabla \hat{f}_{N_{max}}(x_k)\|^2 \, \frac{c_1 2\beta(1-\eta)}{Lc_2^2} \\ &= \|\nabla \hat{f}_{N_{max}}(x_k)\|^2 \, \frac{c_1^2 2\beta(1-\eta)}{Lc_2^2}. \end{aligned}$$

Therefore, (5.38) holds and subtracting $\hat{f}_{N_{max}}(x^*)$ on both sides of inequality (5.33) we obtain

$$\tilde{C}_{k+1} - \hat{f}_{N_{max}}(x^*) \leq \tilde{C}_k - \hat{f}_{N_{max}}(x^*) + \varepsilon_k - \bar{\beta}_1 \|\nabla \hat{f}_{N_{max}}(x_k)\|^2 \tag{5.39}$$

where $\bar{\beta}_1 = \eta(1-\eta_{max})\bar{\beta}_0$. Before proving the main result, we need one more inequality. Having

$$\begin{aligned}\|\nabla \hat{f}_{N_{max}}(x_{k+1})\| - \|\nabla \hat{f}_{N_{max}}(x_k)\| &\leq \|\nabla \hat{f}_{N_{max}}(x_{k+1}) - \nabla \hat{f}_{N_{max}}(x_k)\| \\ &\leq L\|x_{k+1} - x_k\| \\ &= L\alpha_k\|p_k\| \\ &\leq Lc_2\|\nabla \hat{f}_{N_{max}}(x_k)\|\end{aligned}$$

we obtain

$$\|\nabla \hat{f}_{N_{max}}(x_{k+1})\| \leq (1 + Lc_2)\|\nabla \hat{f}_{N_{max}}(x_k)\|. \tag{5.40}$$

Now, we want to prove that there exists $\theta \in (0,1)$ such that for every $k \geq q$

$$\tilde{C}_{k+1} - \hat{f}_{N_{max}}(x^*) < \theta(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)) + \varepsilon_k.$$

Define

$$b = \frac{1}{\bar{\beta}_1 + \gamma(Lc_2 + 1)^2}.$$

Again, we will have two types of iterations for $k \geq q$ but this time regarding $\|\nabla \hat{f}_{N_{max}}(x_k)\|$. First, assume that

$$\|\nabla \hat{f}_{N_{max}}(x_k)\|^2 < b(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)).$$

In that case, Lemma 5.3.1 and inequality (5.40) imply

$$\begin{aligned}\hat{f}_{N_{max}}(x_{k+1}) - \hat{f}_{N_{max}}(x^*) &\leq \gamma\|\nabla \hat{f}_{N_{max}}(x_{k+1})\|^2 \\ &\leq \gamma((1 + Lc_2)\|\nabla \hat{f}_{N_{max}}(x_k)\|)^2 \\ &< \gamma(1 + Lc_2)^2 b(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)).\end{aligned}$$

Setting $\theta_1 = \gamma(1 + Lc_2)^2 b$ we obtain

$$\hat{f}_{N_{max}}(x_{k+1}) - \hat{f}_{N_{max}}(x^*) < \theta_1(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)). \tag{5.41}$$

Notice that $\theta_1 \in (0,1)$ since

$$\theta_1 = \frac{\gamma(1+Lc_2)^2}{\bar{\beta}_1 + \gamma(Lc_2+1)^2}.$$

If $\tilde{C}_{k+1} = \hat{f}_{N_{max}}(x_{k+1})$, then (5.41) obviously implies

$$\tilde{C}_{k+1} - \hat{f}_{N_{max}}(x^*) < \theta_1(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)).$$

If $\tilde{C}_{k+1} = C_{k+1}$, then

$$\begin{aligned}
\tilde{C}_{k+1} - \hat{f}_{N_{max}}(x^*) &= C_{k+1} - \hat{f}_{N_{max}}(x^*) \\
&= \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} C_k + \frac{\hat{f}_{N_{max}}(x_{k+1})}{Q_{k+1}} - \frac{\tilde{\eta}_k Q_k + 1}{Q_{k+1}} \hat{f}_{N_{max}}(x^*) \\
&\leq \frac{\tilde{\eta}_k Q_k}{Q_{k+1}} (\tilde{C}_k - \hat{f}_{N_{max}}(x^*)) \\
&+ \frac{\hat{f}_{N_{max}}(x_{k+1}) - \hat{f}_{N_{max}}(x^*)}{Q_{k+1}} \\
&\leq (1 - \frac{1}{Q_{k+1}})(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)) \\
&+ \frac{\theta_1(\tilde{C}_k - \hat{f}_{N_{max}}(x^*))}{Q_{k+1}} \\
&= (1 - \frac{1-\theta_1}{Q_{k+1}})(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)) \\
&\leq (1 - (1-\eta_{max})(1-\theta_1))(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)).
\end{aligned}$$

In the last inequality, we used the fact that $Q_{k+1} \leq (1-\eta_{max})^{-1}$. Therefore, we conclude that

$$\tilde{C}_{k+1} - \hat{f}_{N_{max}}(x^*) \leq \bar{\theta}_1(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)) \tag{5.42}$$

where $\bar{\theta}_1 = \max\{\theta_1, 1-(1-\eta_{max})(1-\theta_1)\} \in (0,1)$.

On the other hand, if

$$\|\nabla \hat{f}_{N_{max}}(x_k)\|^2 \geq b(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)),$$

inequality (5.39) implies

$$\begin{aligned} \tilde{C}_{k+1} - \hat{f}_{N_{max}}(x^*) &\leq \tilde{C}_k - \hat{f}_{N_{max}}(x^*) + \varepsilon_k - \bar{\beta}_1 b(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)) \\ &= \bar{\theta}_2(\tilde{C}_k - \hat{f}_{N_{max}}(x^*)) + \varepsilon_k \end{aligned}$$

where $\bar{\theta}_2 = 1 - b\bar{\beta}_1$ and therefore $\bar{\theta}_2 \in (0,1)$ since

$$\bar{\theta}_2 = 1 - \frac{\bar{\beta}_1}{\bar{\beta}_1 + \gamma(Lc_2 + 1)^2}.$$

Gathering all the types of iterations, we conclude that for every $k \in \mathbb{N}_0$

$$\tilde{C}_{q+k+1} - \hat{f}_{N_{max}}(x^*) \leq \theta(\tilde{C}_{q+k} - \hat{f}_{N_{max}}(x^*)) + \varepsilon_{q+k}$$

where $\theta = \max\{\bar{\theta}_1, \bar{\theta}_2\}$ and therefore $\theta \in (0,1)$. By the induction argument, the previous inequality implies that for every $k \in \mathbb{N}$ the following holds

$$\tilde{C}_{q+k} - \hat{f}_{N_{max}}(x^*) \leq \theta^k(\tilde{C}_q - \hat{f}_{N_{max}}(x^*)) + \sum_{j=1}^{k} \theta^{j-1}\varepsilon_{q+k-j}.$$

Finally, recalling that $\hat{f}_{N_k}(x_k) \leq \tilde{C}_k$ by definition, we obtain

$$\hat{f}_{N_{max}}(x_{q+k}) - \hat{f}_{N_{max}}(x^*) \leq \theta^k(\tilde{C}_q - \hat{f}_{N_{max}}(x^*)) + \sum_{j=1}^{k} \theta^{j-1}\varepsilon_{q+k-j}.$$

At the end, notice that $\tilde{C}_q - \hat{f}_{N_{max}}(x^*) \geq 0$ since

$$\tilde{C}_q = \max\{\hat{f}_{N_{max}}(x_q), C_q\} \geq \hat{f}_{N_{max}}(x_q) \geq \hat{f}_{N_{max}}(x^*).$$

■

In order to prove R-linear convergence, we impose a stronger assumption on the sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$. Recall that Algorithm 4 assumes that this sequence satisfies assumption C3. Notice that the following assumption implies C3.

C 4 *The sequence of nonnegative numbers $\{\varepsilon_k\}_{k\in\mathbb{N}}$ converges to zero R-linearly.*

Under this assumption, we can prove the following result.

Lemma 5.3.3 *If the assumption C4 is satisfied, then for every $\theta \in (0,1)$ and $q \in \mathbb{N}$*

$$s_k = \sum_{j=1}^{k} \theta^{j-1} \varepsilon_{q+k-j}$$

converges to zero R-linearly.

Proof. Assumption C4 implies the existence of a constant $\rho \in (0,1)$ and a constant $C > 0$ such that $\varepsilon_k \leq C\rho^k$ for every $k \in \mathbb{N}$. Now, since $\rho, \theta \in (0,1)$, we can define $\gamma = \max\{\rho, \theta\} < 1$ such that for every $k \in \mathbb{N}$

$$\begin{aligned} s_k &= \sum_{j=1}^{k} \theta^{j-1} \varepsilon_{q+k-j} \leq \sum_{j=1}^{k} \theta^{j-1} C \rho_{q+k-j} \\ &\leq \sum_{j=1}^{k} C\gamma_{q+k-1} \leq C\gamma^{q-1} \sum_{j=1}^{k} \gamma_k \\ &= C_1 a_k \end{aligned}$$

where $C_1 = C\gamma^{q-1}$ and $a_k = k\gamma^k$. Now we want to prove that the sequence $\{a_k\}_{k\in\mathbb{N}}$ converges to zero R-linearly. Define

$$s = \frac{1+\gamma}{2\gamma}.$$

Since $\gamma < 1$ we have that $s > 1$. Furthermore, we define a sequence $\{c_k\}_{k\in\mathbb{N}}$ as follows

$$c_1 = \left(s^{(\ln s)^{-1}-1} \ln s\right)^{-1},$$

$$c_{k+1} = c_k \frac{ks}{k+1}.$$

This sequence can also be presented as

$$c_k = c_1 \frac{s^{k-1}}{k}.$$

In order to prove that $c_k \geq 1$ for every $k \in \mathbb{N}$, we define the function

$$f(x) = \frac{s^{x-1}}{x}$$

and search for its minimum on the interval $(0, \infty)$. As

$$f'(x) = \frac{s^{x-1}}{x^2} \left(x \ln s - 1\right),$$

the stationary point is $x^* = (\ln s)^{-1} > 0$, i.e. it satisfies $x^* \ln s = 1$. Since

$$f''(x^*) = \frac{s^{x^*} \ln s}{s x^{*2}} > 0,$$

f attains its minimum at x^* and there follows that for every $k \in \mathbb{N}$

$$\frac{s^{k-1}}{k} = f(k) \geq f(x^*) = s^{(\ln s)^{-1}-1} \ln s.$$

Therefore,

$$c_k = c_1 \frac{s^{k-1}}{k} \geq \left(s^{(\ln s)^{-1}-1} \ln s\right)^{-1} \left(s^{(\ln s)^{-1}-1} \ln s\right) = 1.$$

Now, let us define the sequence $b_k = a_k c_k$. Notice that $a_k \leq b_k$. Moreover, we have that

$$b_{k+1} = a_{k+1}c_{k+1} = (k+1)\gamma^{k+1}c_k s\frac{k}{k+1} = s\gamma k\gamma^k c_k = tb_k$$

where $t = s\gamma = \frac{1+\gamma}{2}$ and therefore $t \in (0,1)$. Thus, there exists $B > 0$ such that $b_k \leq Bt^k$. Finally, we obtain

$$s_k \leq C_1 a_k \leq C_1 b_k \leq C_1 B t^k,$$

i.e. we can conclude that the sequence $\{s_k\}_{k\in\mathbb{N}}$ converges to zero R-linearly. ■

Finally, we state the conditions under which R-linear convergence can be achieved.

Theorem 5.3.3 *Suppose that the assumptions of Theorem 5.3.2 are satisfied together with the assumption C4. Then there are constants* $\theta_3 \in (0,1)$, $q \in \mathbb{N}$ *and* $M_q > 0$ *such that for every* $k \in \mathbb{N}$

$$\|x_{q+k} - x^*\| \leq \theta_3^k M_q.$$

Proof. Theorem 5.3.2 implies the existence of $\theta \in (0,1)$ and $q \in \mathbb{N}$ such that for every $k \in \mathbb{N}$

$$\hat{f}_{N_{max}}(x_{q+k}) - \hat{f}_{N_{max}}(x^*) \leq \theta^k M + \sum_{j=1}^{k} \theta^{j-1}\varepsilon_{q+k-j}$$

where $M = \tilde{C}_q - \hat{f}_{N_{max}}(x^*) \geq 0$. Moreover, Lemma 5.3.3 implies that there exists $t \in (0,1)$ and a positive constant S such that

$$\sum_{j=1}^{k} \theta^{j-1}\varepsilon_{q+k-j} \leq St^k.$$

Therefore, if we define $G = M + S$ and $\theta_2 = \max\{\theta, t\}$ we obtain that $\theta_2 < 1$ and

$$\hat{f}_{N_{max}}(x_{q+k}) - \hat{f}_{N_{max}}(x^*) \leq \theta_2^k G.$$

Furthermore, Lemma 5.3.1 implies the existence of a positive constant γ such that for every $k \in \mathbb{N}$

$$\frac{1}{2\gamma}\|x_{q+k} - x^*\|^2 \leq \hat{f}_{N_{max}}(x_{q+k}) - \hat{f}_{N_{max}}(x^*).$$

Therefore,

$$\|x_{q+k} - x^*\|^2 \leq \theta_2^k G 2\gamma$$

and

$$\|x_{q+k} - x^*\| \leq \left(\sqrt{\theta_2}\right)^k \sqrt{G2\gamma}.$$

Defining $\theta_3 = \sqrt{\theta_2}$ and $M_q = \sqrt{G2\gamma}$ we obtain the result. ■

The rest of this section is devoted to the line search with $\tilde{C}_k$ being the maximum of the previous M function values (5.8). In the previous section, where the general search direction was considered and $dm_k(\alpha) = \alpha^2\beta_k$, we have managed to prove the existence of an accumulation point of the sequence of iterates which is stationary for the function $\hat{f}_{N_{max}}$. However, under some auxiliary assumptions we are able to obtain the result where every accumulation point is stationary for $\hat{f}_{N_{max}}$. The descent search directions are assumed and therefore the line search is defined by

$$\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \varepsilon_k + \eta\alpha_k p_k^T \nabla \hat{f}_{N_k}(x_k),$$

$$\tilde{C}_k = \max\{\hat{f}_{N_k}(x_k), \ldots, \hat{f}_{N_{\max\{k-M+1,0\}}}(x_{\max\{k-M+1,0\}})\}. \tag{5.43}$$

Similar line search rule was observed by Dai [17], but with $\varepsilon_k = 0$. A more detailed description of that paper is given in section 2.3. We will begin the analysis by proving the existence of a level set that contains all the iterates x_k for k sufficiently large.

Lemma 5.3.4 *Suppose that the assumptions A1 and A3 are satisfied. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$ and that the sequence $\{x_k\}_{k\in\mathbb{N}}$ is generated by Algorithm 4 with the line search (5.43). Then there exists a finite iteration $\tilde{n}$ such that for every $k > \tilde{n}$ the iterate x_k belongs to the level set*

$$\mathcal{L} = \{x \in \mathbb{R}^n \mid \hat{f}_{N_{max}}(x) \leq \tilde{C}_{\tilde{n}+M} + \varepsilon\}. \tag{5.44}$$

Proof. Lemma 5.2.4 implies the existence of $\tilde{n}$ such that for every $k \geq \tilde{n}$ the sample size is $N_k = N_{max}$. Therefore, the conditions of Lemma 5.2.3 are satisfied. Since the proof of that lemma is conducted for unspecified decrease measure $dm_k \geq 0$, we can conclude that the inequality (5.19) is true for every $m \in \mathbb{N}$, i.e.

$$\tilde{C}_{s(m+1)} \leq \tilde{C}_{s(1)} + \sum_{k=1}^{m}\sum_{i=0}^{M-1} \varepsilon_{s(k)+i} - \eta \sum_{k=1}^{m} dm_{v(k+1)-1} \leq \tilde{C}_{s(1)} + \varepsilon$$

where $s(m) = \tilde{n} + mM$ and $\hat{f}_{N_{max}}(x_{v(m)}) = \tilde{C}_{s(m)}$. In fact, we obtain that for every $k \in \mathbb{N}$

$$\tilde{C}_{s(k)} \leq \tilde{C}_{s(1)} + \varepsilon. \tag{5.45}$$

Moreover, since $\tilde{C}_{s(k)} = \max\{\hat{f}_{N_{max}}(x_{s(k-1)+1}), \ldots, \hat{f}_{N_{max}}(x_{s(k-1)+M})\}$ we have that for every $j \in \{1, \ldots, M\}$ and every $k \in \mathbb{N}$

$$\hat{f}_{N_{max}}(x_{s(k-1)+j}) \leq \tilde{C}_{s(k)}.$$

Notice that $\tilde{C}_{s(1)} = \max\{\hat{f}_{N_{max}}(x_{\tilde{n}+1}), \ldots, \hat{f}_{N_{max}}(x_{\tilde{n}+M})\}$. Therefore, for every $k > \tilde{n}$

$$\hat{f}_{N_{max}}(x_k) \leq \tilde{C}_{s(1)} + \varepsilon = \tilde{C}_{\tilde{n}+M} + \varepsilon$$

which completes the proof. ■

Next, we prove the convergence result.

Theorem 5.3.4 *Suppose that the assumptions A1, A3, A6 and B3 are satisfied and that the level set (5.44) is bounded. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$ and that the sequence $\{x_k\}_{k\in\mathbb{N}}$ is generated by Algorithm 4 with the line search (5.43). Then every accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$ is stationary for the function $\hat{f}_{N_{max}}$.*

Proof. Under these assumptions, Lemma 5.2.4 implies the existence of $\tilde{n}$ such that for every $k \geq \tilde{n}$ the sample size is $N_k = N_{max}$. Then, Lemma 5.2.3 implies that $\liminf_{k\to\infty} dm_k(\alpha_k) = 0$. More precisely, the subset K such that

$$\lim_{k\in K} dm_k(\alpha_k) = 0 \tag{5.46}$$

is defined as $K = \{v(k) - 1\}_{k\in\mathbb{N}}$ where $v(k)$ is the iteration where the maximum was obtained. More precisely, $\hat{f}_{N_{max}}(x_{v(k)}) = \tilde{C}_{s(k)}$ where $\tilde{C}_{s(k)} = \max\{\hat{f}_{N_{max}}(x_{s(k)}), \ldots, \hat{f}_{N_{max}}(x_{s(k)-M+1})\}$ and $s(k) = \tilde{n} + kM$. Notice that

$$v(k) \in \{\tilde{n} + (k-1)M + 1, \ldots, \tilde{n} + kM\}$$

and

$$v(k+1) \in \{\tilde{n} + kM + 1, \ldots, \tilde{n} + (k+1)M\}.$$

Therefore

$$v(k+1) - v(k) \leq 2M - 1.$$

Moreover, this result implies that for every $k \in \mathbb{N}$, $k \geq \tilde{n}$ there exists $\tilde{k} \geq k$, $\tilde{k} \in K$ such that

$$\tilde{k} - k \leq 2M - 2. \tag{5.47}$$

Notice that Lemma 5.3.4 implies that all the iterates x_k, $k > \tilde{n}$ belong to the level set (5.44) which is assumed to be bounded. As it

was derived in the proof of Theorem 5.3.2, assumption B3 together with the Lipschitz continuity assumption A6 implies the existence of the constants $c_3 = 1 + c_2 L$ and $\bar{\beta}_0$ such that for every $k > \tilde{n}$

$$\|\nabla \hat{f}_{N_{max}}(x_{k+1})\| \leq c_3 \|\nabla \hat{f}_{N_{max}}(x_k)\| \tag{5.48}$$

and

$$dm_k(\alpha_k) \geq \bar{\beta}_0 \|\nabla \hat{f}_{N_{max}}(x_k)\|^2.$$

The last inequality and (5.46) together imply

$$\lim_{k \in K} \|\nabla \hat{f}_{N_{max}}(x_k)\| = 0. \tag{5.49}$$

Furthermore, (5.47) and (5.48) imply that for every $k \in \mathbb{N}$, $k > \tilde{n}$ there exists $\tilde{k} \geq k$, $\tilde{k} \in K$ such that

$$\|\nabla \hat{f}_{N_{max}}(x_k)\| \leq c_3^{2M-2} \|\nabla \hat{f}_{N_{max}}(x_{\tilde{k}})\|.$$

Letting $k \to \infty$ in the previous inequality and using (5.49) we obtain

$$\lim_{k \to \infty} \|\nabla \hat{f}_{N_{max}}(x_k)\| \leq \lim_{\tilde{k} \to \infty, \tilde{k} \in K} \|\nabla \hat{f}_{N_{max}}(x_{\tilde{k}})\| = 0,$$

i.e. $\lim_{k \to \infty} \|\nabla \hat{f}_{N_{max}}(x_k)\| = 0$. Finally, if x^* is an arbitrary accumulation point of the sequence $\{x_k\}_{k \in \mathbb{N}}$, i.e. if $\lim_{k \in K_1} x_k = x^*$ for some subset $K_1 \in \mathbb{N}$, then the assumption A1 implies

$$\|\nabla \hat{f}_{N_{max}}(x^*)\| = \lim_{k \in K_1} \|\nabla \hat{f}_{N_{max}}(x_k)\| = \lim_{k \to \infty} \|\nabla \hat{f}_{N_{max}}(x_k)\| = 0.$$

Therefore, every accumulation point of the sequence $\{x_k\}_{k \in \mathbb{N}}$ is stationary for the function $\hat{f}_{N_{max}}$. ■

At the end of this section, we will show that R-linear rate of convergence is also attainable for the line search (5.43). In order to abbreviate the proof, we will use the parts of the previously stated proofs.

Theorem 5.3.5 *Suppose that the assumptions A1, A3, A5, A6, B3 and C4 are satisfied. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$ and that the sequence $\{x_k\}_{k\in\mathbb{N}}$ is generated by Algorithm 4 with the line search (5.43). Then there exist constants $\theta_4 \in (0,1)$, $M_m > 0$, finite number $\tilde{n}$ and an unique minimizer x^* of the function $\hat{f}_{N_{max}}$ such that for every $s \geq M$*

$$\|x_{\tilde{n}+s} - x^*\| \leq \theta_4^s M_m.$$

Proof. Again, we will start by noticing that Lemma 5.2.4 implies the existence of a finite number $\tilde{n}$ such that $N_k = N_{max}$ for every $k \geq \tilde{n}$. Lemma 5.3.4 implies that for every $k > \tilde{n}$ the iterate x_k belongs to the level set (5.44). Strong convexity of the function $\hat{f}_{N_{max}}$ implies the boundedness and convexity of that level set and the existence of an unique minimizer of $\hat{f}_{N_{max}}$. Therefore, there exists at least one accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$. Moreover, Theorem 5.3.4 implies that every accumulation point of that sequence is stationary for the function $\hat{f}_{N_{max}}$ and therefore $\lim_{k\to\infty} x_k = x^*$ where x^* is the unique stationary point of $\hat{f}_{N_{max}}$.

Since the assumptions of Lemma 5.2.3 are satisfied, (5.18) holds and we obtain that for every $k \in \mathbb{N}$

$$\tilde{C}_{s(k+1)} \leq \tilde{C}_{s(k)} + \sum_{i=0}^{M-1} \varepsilon_{s(k)+i} - \eta dm_{v(k+1)-1} \tag{5.50}$$

where $s(k) = \tilde{n} + kM$ and $\hat{f}_{N_{max}}(x_{v(k)}) = \tilde{C}_{s(k)}$. Moreover, as in the proof of Theorem 5.3.4, we conclude that there are constants $c_3 = 1 + c_2 L$ and $\bar{\beta}_0$ such that for every $k > \tilde{n}$

$$\|\nabla \hat{f}_{N_{max}}(x_{k+1})\| \leq c_3 \|\nabla \hat{f}_{N_{max}}(x_k)\|$$

and

$$dm_k(\alpha_k) \geq \bar{\beta}_0 \|\nabla \hat{f}_{N_{max}}(x_k)\|^2.$$

From the previous inequality and (5.50) we obtain

$$\begin{aligned}\tilde{C}_{s(k+1)} - \hat{f}_{N_{max}}(x^*) &\leq \tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*) - \eta\bar{\beta}_0\|\nabla\hat{f}_{N_{max}}(x_{v(k+1)-1})\|^2 \\ &+ \sum_{i=0}^{M-1}\varepsilon_{s(k)+i}.\end{aligned}$$

Define a constant

$$b = \frac{1}{\bar{\beta}_0 + \gamma c_3^2}.$$

If $\|\nabla\hat{f}_{N_{max}}(x_{v(k+1)-1})\|^2 \geq b(\tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*))$ then we have

$$\begin{aligned}\tilde{C}_{s(k+1)} - \hat{f}_{N_{max}}(x^*) &\leq \tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*) - \eta\bar{\beta}_0\|\nabla\hat{f}_{N_{max}}(x_{v(k+1)-1})\|^2 \\ &+ \sum_{i=0}^{M-1}\varepsilon_{s(k)+i} \\ &\leq \tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*) - \eta\bar{\beta}_0 b(\tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*)) \\ &+ \sum_{i=0}^{M-1}\varepsilon_{s(k)+i} \\ &= \theta_1(\tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*)) \\ &+ \sum_{i=0}^{M-1}\varepsilon_{s(k)+i}\end{aligned}$$

where $\theta_1 = 1 - \eta\bar{\beta}_0 b$ and therefore $\theta_1 \in (0,1)$ by definition of b and η. On the other hand, if $\|\nabla\hat{f}_{N_{max}}(x_{v(k+1)-1})\|^2 < b(\tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*))$, using the result of Lemma 5.3.1 we obtain

$$\begin{aligned}\tilde{C}_{s(k+1)} - \hat{f}_{N_{max}}(x^*) &= \hat{f}_{N_{max}}(x_{v(k+1)}) - \hat{f}_{N_{max}}(x^*) \\ &\leq \gamma\|\nabla\hat{f}_{N_{max}}(x_{v(k+1)})\|^2 \\ &\leq \gamma c_3^2\|\nabla\hat{f}_{N_{max}}(x_{v(k+1)-1})\|^2 \\ &< \theta_2(\tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*))\end{aligned}$$

where $\theta_2 = \gamma c_3^2 b$ and $\theta_2 \in (0,1)$ by definition of b. Therefore, for $\theta = \max\{\theta_1, \theta_2\} \in (0,1)$ and for every $k \in \mathbb{N}$

$$\tilde{C}_{s(k+1)} - \hat{f}_{N_{max}}(x^*) \leq \theta(\tilde{C}_{s(k)} - \hat{f}_{N_{max}}(x^*)) + \sum_{i=0}^{M-1} \varepsilon_{s(k)+i}.$$

Using the induction argument, we obtain

$$\tilde{C}_{s(k+1)} - \hat{f}_{N_{max}}(x^*) \leq \theta^k(\tilde{C}_{s(1)} - \hat{f}_{N_{max}}(x^*)) + \sum_{j=1}^{k} \sum_{i=0}^{M-1} \theta^{j-1} \varepsilon_{s(k+1-j)+i}$$

Moreover, for every $j \in \{1, \ldots, M\}$ and every $k \in \mathbb{N}$

$$\hat{f}_{N_{max}}(x_{s(k)+j}) \leq \tilde{C}_{s(k+1)}$$

and therefore

$$\hat{f}_{N_{max}}(x_{s(k)+j}) - \hat{f}_{N_{max}}(x^*) \leq \theta^k V + r_k \tag{5.51}$$

where $V = \tilde{C}_{s(1)} - \hat{f}_{N_{max}}(x^*) \geq 0$ and

$$r_k = \sum_{j=1}^{k} \sum_{i=0}^{M-1} \theta^{j-1} \varepsilon_{s(k+1-j)+i}.$$

Now, assumption C4 implies the existence of $\rho \in (0,1)$ and $C > 0$ such that $\varepsilon_k \leq C\rho^k$ for every k. Defining $C_1 = MC\rho^{\tilde{n}}$ and $\gamma_1 = \max\{\rho^M, \theta\}$

we obtain $\gamma_1 < 1$ and

$$\begin{aligned} r_k &\leq \sum_{j=1}^{k}\sum_{i=0}^{M-1} \theta^{j-1} C \rho^{s(k+1-j)+i} \leq C \sum_{j=1}^{k}\sum_{i=0}^{M-1} \theta^{j-1} \rho^{s(k+1-j)} \\ &= C \sum_{j=1}^{k} \theta^{j-1} M \rho^{(k+1-j)M+\tilde{n}} = MC \sum_{j=1}^{k} \theta^{j-1} \left(\rho^M\right)^{(k+1-j)} \rho^{\tilde{n}} \\ &\leq MC\rho^{\tilde{n}} \sum_{j=1}^{k} \gamma_1^{j-1} \gamma_1^{k+1-j} = C_1 \sum_{j=1}^{k} \gamma_1^k \\ &= C_1 k \gamma_1^k. \end{aligned}$$

Following the ideas from the proof of Lemma 5.3.3, we conclude that there exist $t \in (0,1)$ and $S > 0$ such that $r_k \leq St^k$. Furthermore, defining $D = V + S$ and $\bar{\theta} = \max\{\theta, t\} < 1$ and using (5.51) we obtain

$$\hat{f}_{N_{max}}(x_{s(k)+j}) - \hat{f}_{N_{max}}(x^*) \leq \bar{\theta}^k D.$$

The previous inequality and Lemma 5.3.1 imply the existence of the constants $\theta_3 = \left(\bar{\theta}\right)^{1/2} \in (0,1)$ and $M_h = \sqrt{2\gamma D} > 0$ such that for every $j \in \{1, \ldots, M\}$ and every $k \in \mathbb{N}$

$$\|x_{\tilde{n}+kM+j} - x^*\| \leq \theta_3^k M_h$$

or equivalently for every $j \in \{1, \ldots, M\}$ and every $s \in \mathbb{N}, s \geq M$

$$\|x_{\tilde{n}+s} - x^*\| \leq \theta_3^{\frac{s-j}{M}} M_h.$$

Since $\theta_3 \in (0,1)$ and $j \leq M$ we obtain

$$\|x_{\tilde{n}+s} - x^*\| \leq \theta_3^{\frac{s-j}{M}} M_h \leq \theta_3^{\frac{s}{M}-1} M_h = \theta_4^s M_m$$

where $\theta_4 = \theta_3^{\frac{1}{M}}$ and $M_m = \frac{M_h}{\theta_3}$. ■

Chapter 6

Numerical results

In the previous two chapters we established the convergence theory for the proposed algorithms. Assuming that the lack of precision is bounded away from zero and imposing the standard assumptions that come from the well known deterministic optimization theory yielded the convergence results. However, equally important issue in numerical optimization is practical implementation of the considered algorithms. This chapter is devoted to the performance evaluation of the proposed methods.

The chapter is divided into two parts. In the first section we consider the variable sample size method proposed in Chapter 4 where the monotone line search rule is used. The goal of this testing was to see whether the variable sample size scheme does have a positive effect on the performance of the algorithm. The method proposed in Chapter 4 is compared with the other variable sample size techniques. Also, the role of the safeguard parameter ρ_k is examined. Recall that the idea of imposing this safeguard check is to prohibit the potentially unproductive decrease in the sample size. The second section is primary devoted to the comparison of the different line search rules in the variable sample size framework. Therefore, the algorithm proposed in

Chapter 5 is considered.

All the proposed methods have the goal of decreasing the number of function evaluations needed for obtaining reasonably good approximation of a solution. Therefore, the number of function evaluations represents the main criterion for comparing the algorithms within this chapter.

6.1 Variable sample size methods

In this section we present some numerical results obtained by Algorithm 1 and compare them with the results obtained by two other methods. The first subsection contains the results obtained on a set of academic test examples while the second subsection deals with the discrete choice problem that is relevant in many applications. The test examples presented in 6.1.1 consist of two different sets. The first one includes Aluffi-Pentini's problem (Montaz Ali et al. [44]) and Rosenbrock problem [22] in noisy environments. Both of them are convenient for initial testing purposes as one can solve them analytically and thus we can actually compute some quality indicators of the approximate solutions obtained by the presented variable sample size line search methods. The second set of examples consists of five larger dimension problems in noisy environments taken from [44]. The Mixed Logit problem is slightly different than the problem (4.4). Given the practical importance of this problem we introduce some minor adjustments of Algorithm 1 and report the results in 6.1.2. This problem is solved by all considered methods.

As common in numerical testing of noisy problems we are measuring the cost by the number of function evaluations needed for achieving the exit criteria. In all presented examples we say that the exit criteria is satisfied if we reach the point x_k such that

$$\|\nabla \hat{f}_{N_{\max}}(x_k)\| < 10^{-2}. \tag{6.1}$$

As the exit criteria implies that the approximate solutions obtained by all methods are of the same quality, the number of function evaluations is a relevant measure for comparison of the considered methods.

Except for the Rosenbrock function, all problems are solved by four different implementations of Algorithm 1, two different heuristic methods and two different implementations of the SAA. Let us state the details of their implementation. We start by defining the search directions.

Algorithm 1 uses an unspecified descent direction p_k at step S4. We report the results for two possible directions, the negative gradient

$$p_k = -\nabla \hat{f}_{N_k}(x_k), \tag{6.2}$$

and the second order direction obtained by

$$p_k = -H_k \nabla \hat{f}_{N_k}(x_k), \tag{6.3}$$

where H_k is a positive definite matrix that approximates the inverse Hessian matrix $(\nabla^2 \hat{f}_{N_k}(x_k))^{-1}$. Among many options for H_k we have chosen the BFGS approach with $H_0 = I$ where I denotes the identity matrix. The inverse Hessian approximation is updated by the BFGS formula

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

where $s_k = x_{k+1} - x_k$ and

$$y_k = \nabla \hat{f}_{N_{k+1}}(x_{k+1}) - \nabla \hat{f}_{N_k}(x_k).$$

The condition $y_k^T s_k > 0$ ensures positive definiteness of the next BFGS update. We enforced this condition or otherwise take $H_{k+1} = H_k$. This way the approximation matrix remains positive definite and provides the decreasing search direction (6.3).

Notice also that the assumption B1 is satisfied for both direction (6.2) or (6.3), but in the case of (6.3) we need to assume that $F(\cdot, \xi) \in$

C^2 instead of A1. Furthermore, some kind of boundedness for H_k is also necessary.

We implemented the safeguard rule presented in section 4.2 where the decrease of a sample size is declined if $\rho_k < \eta_0$ where

$$\rho_k = \frac{\hat{f}_{N_k^+}(x_k) - \hat{f}_{N_k^+}(x_{k+1})}{\hat{f}_{N_k}(x_k) - \hat{f}_{N_k}(x_{k+1})}.$$

Therefore, if we choose to apply the safeguard rule we set the input parameter η_0 to be some finite number. On the other hand, if we set $\eta_0 = -\infty$ the safeguard rule is not applied and thus the algorithm accepts the candidate sample size for the next iteration. In other words, for every iteration k we have that $N_{k+1} = N_k^+$.

Based on the descent direction choice and the safeguard rule application, four different implementations of Algorithm 1 are tested here. As all considered methods are implemented with both descent directions, NG and BFGS are used to denote the negative gradient search direction and BFGS search direction in general. The implementations of Algorithm 1 that do not use the safeguard rule i.e. with $\eta_0 = -\infty$ are denoted by $\rho = -\infty$, while $\rho = \eta_0$ stands for the implementations that use the safeguard rule with the value η_0. The input parameters of Algorithm 2 is $\nu_1 = 1/\sqrt{N_{max}}$.

In this implementation, the step S3 of Algorithm 1 is slightly altered. Namely, the condition $\|\nabla \hat{f}_{N_k}(x_k)\| = 0$ is replaced by

$$\|\nabla \hat{f}_{N_k}(x_k)\| \leq \max\{0, 10^{-2} - \tilde{\varepsilon}_\delta^{N_k}(x_k)\}$$

where $\tilde{\varepsilon}_\delta^{N_k}(x_k)$ is the measure of the confidence interval for $\|\nabla \hat{f}_{N_k}(x_k)\|$ around $\|\nabla f(x_k)\|$. Recall that we are interested in finding the stationary point of the function f which is assumed to be well approximated by the function $\hat{f}_{N_{max}}$. Moreover, we are assuming that the interchange of the gradient and the expectation is allowed and therefore

$\nabla f_{N_{max}}$ is a relevant estimator of ∇f. Suppose that $\|\nabla \hat{f}_{N_k}(x_k)\| \leq 10^{-2} - \tilde{\varepsilon}_{\delta}^{N_k}(x_k)$. This means that $\|\nabla f(x_k)\| \leq 10^{-2}$ with some high probability which further implies that we are probably close to the stationary point of the original objective function. The parameter $\tilde{\varepsilon}_{\delta}^{N_k}(x_k)$ is set to be of the form of previously defined lack of precision, but with $\hat{\sigma}^2(x_k)$ being the sample variance of $\|\nabla F(x_k, \xi)\|$. As the gradient $\nabla \hat{f}_{N_k}(x_k)$ is already available, this measure for the confidence interval is obtained without additional costs in terms of function evaluations.

The heuristic is motivated by the following scheme: conduct first 10% of iterations with the sample size $0.1N_{\max}$, then the following 10% with the sample size $0.2N_{\max}$ and so on. We implemented this scheme for both descent directions as for Algorithm 1 - the negative gradient and the BFGS direction. The scheme suggested is slightly adjusted to allow us to compare the results with other methods i.e. to ensure that we get the approximate solution with the same exit criteria as in all other tested methods. We consider the number of iterations used by the corresponding Algorithm 1 (negative gradient or BFGS) with $\rho = \eta_0$ as the reference number of iterations, say K. Then we perform $0.1K$ iterations (rounded if necessary) with the sample size $0.1N_{\max}$, another $0.1K$ iterations with the sample size $0.2N_{\max}$ and so on until (6.1) is reached. This way we ensured that the solutions obtained by this scheme are comparable with those obtained by other methods.

Sample Average Approximation method works directly with the function $\hat{f}_{N_{max}}$. We tested SAA methods here with both negative gradient and BFGS direction. The line search used for all of the above-described methods is the one defined in step S5 of Algorithm 1 with the value for the Armijo parameter $\eta = 10^{-4}$. The backtracking is performed with $\beta = 0.5$.

σ^2	global minimizer - x^*	local minimizer	maximizer	$f(x^*)$
0.01	$(-1.02217, 0)$	$(0.922107, 0)$	$(0.100062, 0)$	-0.340482
0.1	$(-0.863645, 0)$	$(0.771579, 0)$	$(0.092065, 0)$	-0.269891
1	$(-0.470382, 0)$	$(0.419732, 0)$	$(0.05065, 0)$	-0.145908

Table 6.1: Stationary points for Aluffi-Pentini's problem. Stacionarne tačke za Aluffi-Pentini problem.

6.1.1 Noisy problems

First, we present the numerical results obtained for Aluffi-Pentini's problem which can be found in [44]. Originally, this is a deterministic problem with box constraints. Following the ideas from [22], some noise is added to the first component of the decision variable and the constraints are removed, so the objective function becomes

$$f(x) = E(0.25(x_1\xi)^4 - 0.5(x_1\xi)^2 + 0.1\xi x_1 + 0.5x_2^2),$$

where ξ represents a random variable with the normal distribution

$$\xi : \mathcal{N}(1, \sigma^2). \tag{6.4}$$

This problem is solved with three different levels of variance. As we are able to calculate the real objective function and its gradient, we can actually see how close are the approximate and the true stationary points. Table 6.1 contains the stationary points for various levels of noise and the global minimums of the relevant objective functions.

We conducted 50 independent runs of each algorithm with $x_0 = (1, 1)^T$ and $N_0^{\min} = 3$. The sample of size N_{max} is generated for each run and all algorithms are tested with that same sample realization. The results in the following tables are the average values obtained from these 50 runs. Columns $\|\nabla \hat{f}_{N_{max}}\|$ and $\|\nabla f\|$ give, respectively, the average values of the gradient norm at the final iterations for

the approximate problem and for the original problem while ϕ represents the average number of function evaluations with one gradient evaluation being counted as n function evaluations. The last column is added to facilitate comparison and represents the percentage increase/decrease in the number of function evaluations for different methods with $\rho = 0.7$ being the benchmark method. So if the number of function evaluations is ϕ_ρ for the benchmark method and ϕ_i is the number of function evaluations for any other method then the reported number is $(\phi_i - \phi_\rho)/\phi_\rho$.

The methods generated by Algorithm 1 clearly outperform the straightforward SAA method as expected. The heuristic approach is fairly competitive in this example, in particular for problems with smaller variance. The safeguard rule with $\eta_0 = 0.7$ is beneficial in all cases, except for the BFGS direction and $\sigma = 0.1$ where it does not make significant difference in comparison to $\rho = -\infty$. The decrease in the sample size is proposed in approximately 20% of iterations and the safeguard rule is active in approximately half of these iterations.

Given that the considered problems have more than one stationary point we report the distribution of the approximate stationary points in Table 6.3. Columns *global*, *local* and *max* count the numbers of replicants converging to the global minimizer, local minimizer and maximizer respectively. Columns fgm and flm represent the average values of function f in the runs that converged to the global minimizer and local minimizer, respectively.

All methods behave more or less similarly. Notice that as the variance increases, the number of replications that are converging towards the global minimizers increases as well. However, we also registered convergence towards maximizers when the variance is increased. The only exception from this relatively similar behavior of all methods appears to happen for $\sigma = 0.1$ where SAA strongly favors the local minimizers while all other methods converge to the global minimizers

NG				
$\sigma^2 = 0.01$, $N_{max} = 100$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.00747	0.01501	1308	9.01
$\rho = 0.7$	0.00767	0.01496	1200	0.00
Heur	0.00618	0.01480	1250	4.24
SAA	0.00844	0.01378	1832	52.73
$\sigma^2 = 0.1$, $N_{max} = 200$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.00722	0.03499	3452	7.84
$\rho = 0.7$	0.00718	0.03435	3201	0.00
Heur	0.00658	0.03531	3556	11.09
SAA	0.00793	0.03005	4264	33.23 5
$\sigma^2 = 1$, $N_{max} = 600$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.00540	0.06061	13401	17.78
$\rho = 0.7$	0.00528	0.06071	11378	0.00
Heur	0.00492	0.05843	13775	21.07
SAA	0.00593	0.05734	15852	39.32

BFGS				
$\sigma^2 = 0.01$, $N_{max} = 100$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.00389	0.01208	811	6.64
$\rho = 0.7$	0.00365	0.01279	761	0.00
Heur	0.00407	0.01383	852	12.04
SAA	0.00527	0.01398	940	23.55
$\sigma^2 = 0.1$, $N_{max} = 200$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.00363	0.03530	1948	-0.38
$\rho = 0.7$	0.00341	0.03466	1955	0.00
Heur	0.00414	0.03460	2284	16.81
SAA	0.00392	0.03051	2928	49.75
$\sigma^2 = 1$, $N_{max} = 600$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.00303	0.06110	8478	15.53
$\rho = 0.7$	0.00358	0.06116	7338	0.00
Heur	0.00344	0.05656	8719	18.81
SAA	0.00336	0.06444	14784	101.46

Table 6.2: Aluffi-Pentini's problem. Aluffi-Pentini problem.

NG					
$\sigma^2 = 0.01$, $N_{max} = 100$					
Algorithm	g	l	max	fgm	flm
$\rho = -\infty$	0	50	0	-	-0.14524
$\rho = 0.7$	0	50	0	-	-0.14542
Heur	0	50	0	-	-0.14542
SAA	0	50	0	-	-0.14542
$\sigma^2 = 0.1$, $N_{max} = 200$					
Algorithm	g	l	max	fgm	flm
$\rho = -\infty$	14	35	1	-0.11712	-0.12887
$\rho = 0.7$	17	32	1	-0.11507	-0.13104
Heur	20	30	0	-0.11364	-0.13275
SAA	1	49	0	-0.10523	-0.12551
$\sigma^2 = 1$, $N_{max} = 600$					
Algorithm	g	l	max	fgm	flm
$\rho = -\infty$	35	15	0	-0.12674	-0.097026
$\rho = 0.7$	36	14	0	-0.11956	-0.11337
Heur	34	16	0	-0.12114	-0.11079
SAA	33	17	0	-0.11745	-0.11857

BFGS					
$\sigma^2 = 0.01$, $N_{max} = 100$					
Algorithm	g	l	max	fgm	flm
$\rho = -\infty$	0	50	0	-	-0.14543
$\rho = 0.7$	0	50	0	-	-0.14543
Heur	0	50	0	-	-0.14543
SAA	0	50	0	-	-0.14543
$\sigma^2 = 0.1$, $N_{max} = 200$					
Algorithm	g	l	max	fgm	flm
$\rho = -\infty$	14	36	0	-0.11710	-0.12818
$\rho = 0.7$	14	36	0	-0.11710	-0.12818
Heur	15	35	0	-0.11635	-0.12882
SAA	1	49	0	-0.10533	-0.12548
$\sigma^2 = 1$, $N_{max} = 600$					
Algorithm	g	l	max	fgm	flm
$\rho = -\infty$	37	13	0	-0.12047	-0.13036
$\rho = 0.7$	36	14	0	-0.11982	-0.13133
Heur	28	22	0	-0.11887	-0.12835
SAA	50	0	0	-0.11230	-

Table 6.3: The approximate stationary points for Aluffi-Pentini's problem. Aproksimativne stacionarne tačke za Aluffi-Pentini problem.

σ^2	global minimizer - x^*	$f(x^*)$
0.001	$(0.711273, 0.506415)$	0.186298
0.01	$(0.416199, 0.174953)$	0.463179
0.1	$(0.209267, 0.048172)$	0.634960

Table 6.4: Rosenbrock problem - the global minimizers. Rosenbrock problem - tačke globalnog minimuma.

more frequently.

The next example is based on the Rosenbrock function. Following the example from [22], the noise is added to the first component in order to make it random. The following objective function is thus obtained

$$f(x) = E(100(x_2 - (x_1\xi)^2)^2 + (x_1\xi - 1)^2), \tag{6.5}$$

where ξ is the random variable defined with (6.4). This kind of function has only one stationary point which is global minimizer, but it depends on the level of noise. The algorithms are tested with the dispersion parameter σ^2 equal to 0.001, 0.01 and 0.1. An interesting observation regarding this problem is that the objective function (6.5) becomes more and more "optimization friendly" when the variance increases. Therefore, we put the same maximal sample size for all levels of noise. The stationary points and the minimal values of the objective function are given in Table 6.4 while the graphics below represent the shape of the objective function f for variances 0.001 and 1 respectively.

Minimization of the Rosenbrock function is a well known problem and in general the second-order directions are necessary to solve it. The same appears to be true in a noisy environment. As almost all runs with the negative gradient failed, only BFGS type results are presented in Table 6.5. All the parameters are the same as in the previous example except that the initial approximation is $x_0 = (-1, 1.2)^T$.

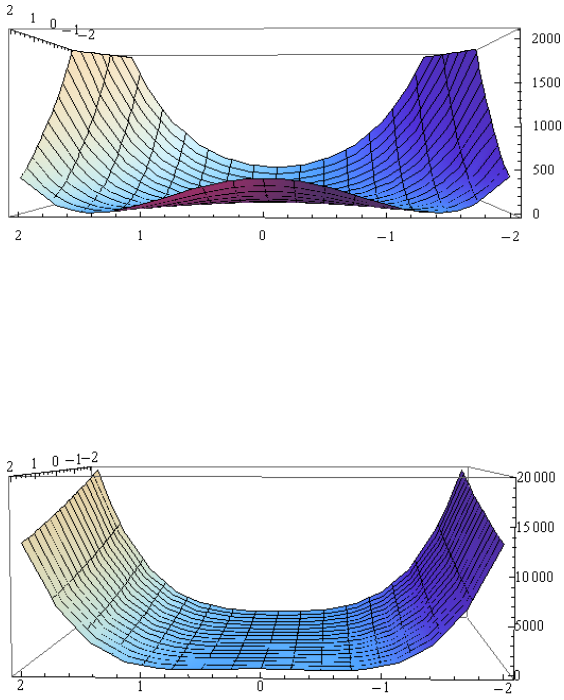

Figure 6.1: Rosenbrock function with different levels of variance. Rosenbrock funkcija sa različitim nivoima varijanse.

BFGS				
$\sigma^2 = 0.001$, $N_{max} = 3500$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.003939	0.208515	44445	7.51
$\rho = 0.7$	0.003595	0.208355	41338	0.00
Heur	0.002521	0.206415	127980	209.59
SAA	0.003241	0.208450	247625	499.03
$\sigma^2 = 0.01$, $N_{max} = 3500$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.003064	0.132830	58944	7.74
$\rho = 0.7$	0.003170	0.132185	54711	0.00
Heur	0.001968	0.132730	114070	108.5
SAA	0.003156	0.132155	216825	296.3
$\sigma^2 = 0.1$, $N_{max} = 3500$				
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	$\|\nabla f\|$	ϕ	%
$\rho = -\infty$	0.003387	0.091843	70958	3.49
$\rho = 0.7$	0.003359	0.091778	68566	0.00
Heur	0.002259	0.091167	106031	54.64
SAA	0.003279	0.092130	161525	135.58

Table 6.5: Rosenbrock problem. Rosenbrock problem.

The same conclusion is valid for this example as for Aluffi-Pentini's problem - the variable sample size strategy reduces the number of function evaluations. Moreover, as far as this example is concerned, a clear advantage is assigned to the algorithm that uses the safeguard rule. The heuristic sample size scheme does not appear to be well suited for this example although the performance improves significantly as the variance increases. The percentage of iterations where the decrease of a sample size is considered increases with the noise and varies from 7% for $\sigma = 0.001$ to 30% for $\sigma = 1$. The rejection due to the safeguard rule from Algorithm 3 also differs, from 15% in the first case to 33% in the case with the largest variance.

Let us now present the results for larger dimension problems. We consider the set of five problems, each one of the dimension 10. The problems from [44] are stated below together with their initial approximations x_0.

- Exponential problem

$$f(x) = E\left(-e^{-0.5\|\xi x\|^2}\right),\ x_0 = (0.5, \ldots, 0.5)^T$$

- Griewank problem

$$f(x) = E\left(1 + \frac{1}{4000} \parallel \xi x \parallel^2 - \prod_{i=1}^{10} \cos\left(\frac{x_i \xi}{\sqrt{i}}\right)\right),$$

$$x_0 = (10, \ldots, 10)^T.$$

- Neumaier 3 problem

$$f(x) = E\left(\sum_{i=1}^{10}(\xi x_i - 1)^2 - \sum_{i=2}^{10} \xi^2 x_i x_{i-1}\right),\ x_0 = (1, \ldots, 1)^T.$$

- Salomon problem

$$f(x) = E\left(1 - \cos(2\pi \parallel \xi x \parallel^2) + 0.1 \parallel \xi x \parallel^2\right),\ x_0 = (2, \ldots, 2)^T.$$

- Sinusoidal problem

$$f(x) = E\left(-A\prod_{i=1}^{10}\sin(\xi x_i - z) - \prod_{i=1}^{10}\sin(B(\xi x_i - z))\right),$$

$$A = 2.5, B = 5, z = 30,\ x_0 = (1, \ldots, 1)^T.$$

The noise component ξ represents normally distributed random variable $\mathcal{N}(1, \sigma^2)$ with different values of σ as specified in Tables 6.6-6.10. All results are obtained taking $N_0^{\min} = 3$ with exit criteria (6.1). The considered methods are again the same - four variants of Algorithm 1 (the negative gradient with $\rho = -\infty$ and $\rho = 0.7$ and the BFGS methods with $\rho = -\infty$ and $\rho = 0.7$), the heuristic sample size scheme and the SAA method, in total 8 methods. Two levels of noise $\sigma^2 = 0.1$ and $\sigma^2 = 1$ are considered for each of the five problems resulting in the set of 10 problems.

When the number of test problems is that big, it is not that easy to compare the algorithms by observing the numbers in tables. Therefore, alternative ways for presenting the overall results are developed. One of them is the performance profile (Dolan, Moré [24]) which is presented in Figure 6.2. Roughly speaking, the performance profile gives the probability that the considered method will be close enough to the best one. Here, the criterion is the number of function evaluations and the best method is the one that has the smallest ϕ. The probability is given by the relative frequency and the term "close enough" is determined by the tolerance level α. Specially for $\alpha = 1$, performance profile represents the probability that the method is going to be the best. For example, Figure 6.2 implies that BFGS $\rho = 0.7$ performed the best

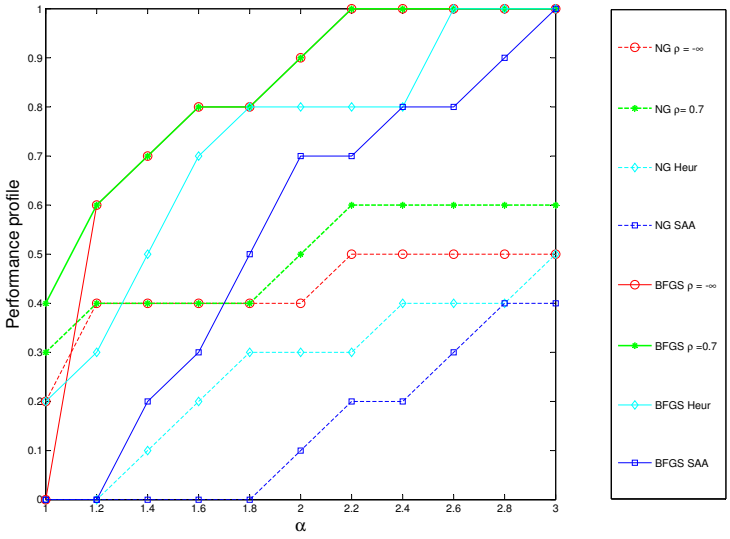

Figure 6.2: Performance profile. Profil učinka.

in 40% of the considered problems, i.e. 4 of 10. Furthermore, if we take a look at the tolerance level $\alpha = 1.2$, then the same method has the performance profile equal to 0.5. This means that in 50% of problems this method took no more than 20% function evaluations more than the best method. In other words, if we denote the minimum number of function evaluations among the considered methods by ϕ_{min}, then in 5 of 10 problems inequality $\phi(BFGS\rho = 0.7) \leq 1.2\phi_{min}$ was satisfied. As the performance profile clearly indicates, all implementations of Algorithm 1 clearly outperformed both the heuristic and SAA corresponding methods.

A natural question here is the dynamics of the variable sample scheme and the actual influence of the decrease as well as the safeguard

rule. The number of iterations where $N_k^+ < N_k$ varies very much through the set of examples and variances. The Griewank test function is solved by both NG methods without any decrease at all. A number of BFGS iterations where $N_k^+ < N_k$ occurred was also rather small and the average number of safeguard rule calls varies from 11% to 20% for this example and none of the decreases is beneficial in terms of function evaluations. This is the only example where the heuristic scheme is the best method for both directions. In all other examples a decrease in the sample size occurs and the safeguard is applied. However the numbers are rather different, ranging from a couple of percent to almost one half of the iterations. The same range is valid for the rejection of the decrease according to the safeguard rule. The average number of iterations where $N_k^+ < N_k$ for all tested examples and both NG and BFGS methods is 14.87% . The decrease is judged as unproductive and it is rejected in 20.57% of cases on average. It is quite clear that the safeguard rule i.e. the appropriate value of the parameter which determines the acceptance or rejection of the decrease is problem dependent. We report results for the same value of that parameter for all examples and methods to make the comparison fair as all other parameters have same values for all problems.

To conclude this discussion the plot of the sample scheme dynamic for the Sinusoidal problem and one noise realization with $\sigma = 1$ and NG direction is shown in Figure 6.3. The NG $\rho = 0.7$ method requested 26219 function evaluations, while NG with $\rho = -\infty$ took 40385 function evaluations, and NG Heur 39983 function evaluations. As in almost all examples SAA NG is the worst requiring 86500 function evaluations. One can see in Figure 6.2 that the safeguard rule rejects the decrease at the 6th iteration and keeps the maximal sample size until the end of the process, while the method with $\rho = -\infty$ performed a number of sample decreases which are in fact unproductive in this example.

A more detailed account of these tests is available in Tables 6.6-

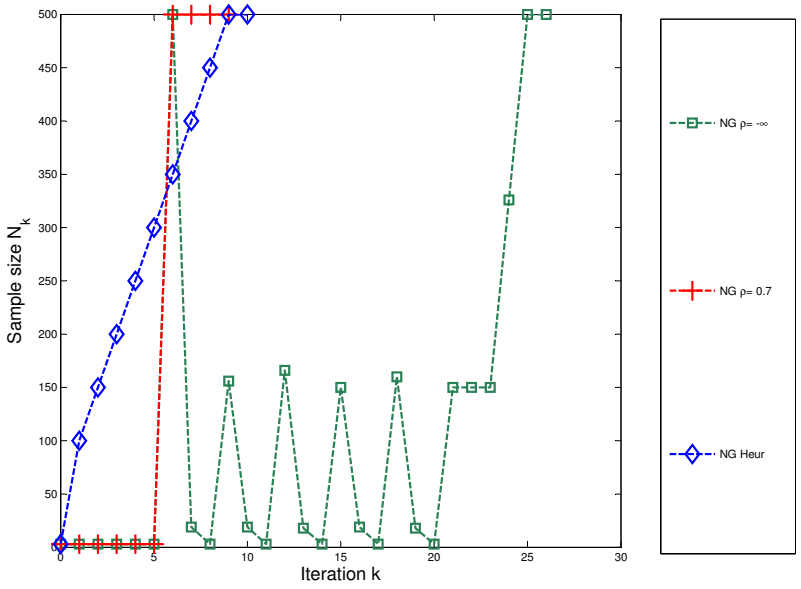

Figure 6.3: Sample size versus iteration. Veličina uzorka u odnosu na iteracije.

	NG			BFGS		
$\sigma^2 = 0.1$, $N_{max} = 200$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00087	4591	0.00	0.00154	4604	0.00
$\rho = 0.7$	0.00087	4591	0.00	0.00154	4604	0.00
Heur	0.00111	7033	53.18	0.00131	7018	52.42
SAA	0.00314	11600	152.64	0.00081	12200	164.97
$\sigma^2 = 1$, $N_{max} = 500$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00313	47454	68.71	0.00088	15752	2.53
$\rho = 0.7$	0.00217	28128	0.00	0.00138	15364	0.00
Heur	0.00400	575270	1945.22	0.00054	21668	41.04
SAA	0.00474	668025	2274.99	0.00268	36250	135.95

Table 6.6: Exponential problem. Eksponencijalni problem.

6.10. The structure of the tables is the same as before - the columns are the value of the sample gradient at the last iteration, the cost measured as the number of function evaluations and the column showing the relative increase/decrease for different methods. The cost of Algorithm 1 with the safeguard is taken as the benchmark. All algorithms are tested in 20 independent runs and the reported numbers are the average values of these 20 runs. The same sample realizations are used for all methods.

6.1.2 Application to the Mixed logit models

In this subsection we present numerical results obtained by applying slightly modified algorithms on simulated data. Discrete choice problems are the subject of various disciplines such as econometrics, transportation, psychology etc. The problem that we considered is an unconstrained parameter estimation problem. We briefly describe the problem here while the more detailed description with further refer-

	NG			BFGS		
$\sigma^2 = 0.1$, $N_{max} = 500$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00997	1796250	0.00	0.00795	312840	-0.86
$\rho = 0.7$	0.00997	1796250	0.00	0.00822	315550	0.00
Heur	0.00988	1160300	-35.40	0.00505	172490	-45.34
SAA	0.00996	1800750	0.25	0.00794	504425	59.86
$\sigma^2 = 1$, $N_{max} = 1000$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00993	6343500	0.00	0.00758	408585	1.98
$\rho = 0.7$	0.00993	6343500	0.00	0.00759	400670	0.00
Heur	0.00995	3790300	-40.25	0.00537	264070	-34.09
SAA	0.00994	6355500	0.19	0.00698	340150	-15.10

Table 6.7: Griewank problem. Griewank problem.

	NG			BFGS		
$\sigma^2 = 0.1$, $N_{max} = 500$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00732	798625	-1.85	0.00305	30223	0.80
$\rho = 0.7$	0.00714	813685	0.00	0.00306	29984	0.00
Heur	0.00598	725680	-10.82	0.00384	40338	34.53
SAA	0.00663	1052025	29.29	0.00278	54825	82.85
$\sigma^2 = 1$, $N_{max} = 2000$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00949	3050850	0.17	0.00421	138195	2.71
$\rho = 0.7$	0.00948	3045650	0.00	0.00354	134555	0.00
Heur	0.00945	2199650	-27.78	0.00503	161140	19.76
SAA	0.00937	3496200	14.79	0.00128	190000	41.21

Table 6.8: Neumaier 3 problem. Neumaier 3 problem.

	NG			BFGS		
$\sigma^2 = 0.1$, $N_{max} = 500$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00411	26590	8.55	0.00376	30814	-7.26
$\rho = 0.7$	0.00396	24495	0.00	0.00297	33226	0.00
Heur	0.00569	54620	122.99	0.00243	59057	77.74
SAA	0.00497	44750	82.69	0.00452	30250	-8.96
$\sigma^2 = 1$, $N_{max} = 2000$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00164	75078	-16.20	0.00234	154245	0.00
$\rho = 0.7$	0.00157	89595	0.00	0.00235	154245	0.00
Heur	0.00153	127920	42.78	0.00214	182650	18.42
SAA	0.00272	196100	118.87	0.00349	143100	-7.23

Table 6.9: Salomon problem. Salomon problem.

	NG			BFGS		
$\sigma^2 = 0.1$, $N_{max} = 200$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00525	22578	2.99	0.00169	10518	0.54
$\rho = 0.7$	0.00520	21923	0.00	0.00125	10461	0.00
Heur	0.00457	29512	34.61	0.00202	18450	76.36
SAA	0.00575	32860	49.89	0.00326	18470	76.56
$\sigma^2 = 1$, $N_{max} = 500$						
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00473	30968	2.14	0.00349	30550	-0.60
$\rho = 0.7$	0.00449	30320	0.00	0.00339	30735	0.00
Heur	0.00385	40453	33.42	0.00338	37588	22.30
SAA	0.00527	65475	115.95	0.00473	48525	57.88

Table 6.10: Sinusoidal problem. Sinusoidalni problem.

ences can be found in [3, 4, 5].

Let us consider a set of r_a agents and r_m alternatives. Suppose that every agent chooses one of finitely many alternatives. The choice is made according to r_k characteristics that each alternative has. Suppose that they are all numerical. Further, each agent chooses the alternative that maximizes his utility. Utility of agent i for alternative j is given by

$$U_{i,j} = V_{i,j} + \varepsilon_{i,j},$$

where $V_{i,j}$ depends on the vector of characteristics of alternative j defined by $m_j = (k_1^j, ..., k_{r_k}^j)^T$ and $\varepsilon_{i,j}$ is the error term. We consider probably the most popular model in practice where $V_{i,j}$ is a linear function, that is

$$V_{i,j} = V_{i,j}(\beta^i) = m_j^T \beta^i.$$

The vector β^i, $i = 1, 2, ..., r_a$ has r_k components, all of them normally distributed. More precisely,

$$\beta^i = (\beta_1^i, ..., \beta_{r_k}^i)^T = (\mu_1 + \xi_1^i \sigma_1, ..., \mu_{r_k} + \xi_{r_k}^i \sigma_{r_k})^T,$$

where ξ_j^i, $i = 1, 2, ..., r_a$, $j = 1, 2, ..., r_k$ are i.i.d. random variables with the standard normal distribution. In other words, $\beta_k^i : \mathcal{N}(\mu_k, \sigma_k^2)$ for every i. The parameters μ_k and σ_k, $k = 1, 2, ..., r_k$ are the ones that should be estimated. Therefore, the vector of unknowns is

$$x = (\mu_1, \ldots, \mu_{r_k}, \sigma_1, \ldots, \sigma_{r_k})^T$$

and the dimension of our problem is $n = 2r_k$. Thus $V_{i,j}$ is a function of x and the random vector ξ^i,

$$V_{i,j} = m_j^T \beta^i(x, \xi^i) = \sum_{s=1}^{r_k} k_s^j (x_s + \xi_s^i x_{r_k+s}) = V_{i,j}(x, \xi^i).$$

The term $\varepsilon_{i,j}$ is a random variable whose role is to collect all factors which are not included in the function $V_{i,j}$. It can also be viewed as the

taste of each agent. Different assumptions about these terms lead to different models. We assume that for every i and every j the random variable $\varepsilon_{i,j}$ follows the Gumbel distribution with location parameter 0 and scale parameter 1. The Gumbel distribution is also known as the type 1 extreme value distribution.

Now, suppose that every agent makes his own choice among these alternatives. The problem is to maximize the likelihood function. Under the assumptions that are stated above, if the realization $\bar{\xi}^i$ of $\xi^i = (\xi^i_1, ..., \xi^i_{r_k})^T$ is known, the probability that agent i chooses alternative j becomes

$$L_{i,j}(x, \bar{\xi}^i) = \frac{e^{V_{i,j}(x,\bar{\xi}^i)}}{\sum_{s=1}^{r_m} e^{V_{i,s}(x,\bar{\xi}^i)}}.$$

Moreover, the unconditional probability is given by

$$P_{i,j}(x) = E(L_{i,j}(x, \xi^i)).$$

Now, if we denote by $j(i)$ the choice of agent i, the problem becomes

$$\max_{x\in\mathbb{R}^n} \prod_{i=1}^{r_a} P_{i,j(i)}(x). \tag{6.6}$$

The equivalent form of (6.6) is given by

$$\min_{x\in\mathbb{R}^n} -\frac{1}{r_a} \sum_{i=1}^{r_a} \ln E(L_{i,j(i)}(x, \xi^i)).$$

Notice that this problem is similar, but not exactly the same as (4.1). The objective function is now

$$f(x) = -\frac{1}{r_a} \sum_{i=1}^{r_a} \ln E(L_{i,j(i)}(x, \xi^i)),$$

so the approximating function is

$$\hat{f}_N(x) = -\frac{1}{r_a}\sum_{i=1}^{r_a}\ln(\frac{1}{N}\sum_{s=1}^{N}L_{i,j(i)}(x,\xi_s^i)).$$

Here $\xi_1^i, ..., \xi_N^i$ are independent realizations of the random vector ξ^i. The realizations are independent across the agents as well. Calculating the exact gradient of $\hat{f}_N$ is affordable and the derivative based approach is suitable.

One of the main differences between algorithms presented in previous sections and the ones that are used for Mixed Logit problem is the way that we calculate the lack of precision, $\varepsilon_\delta^N(x)$. We define the approximation of the confidence interval radius as it is proposed in Bastin et al. [5],

$$\varepsilon_\delta^N(x) = \frac{\alpha_\delta}{r_a}\sqrt{\sum_{i=1}^{r_a}\frac{\hat{\sigma}^2_{N,i,j(i)}(x)}{NP^2_{i,j(i)}(x)}}. \tag{6.7}$$

Here, α_δ represents the same parameter as in (4.8) and $\hat{\sigma}^2_{N,i,j(i)}(x)$ is the sample variance estimator, i.e.

$$\hat{\sigma}^2_{N,i,j(i)}(x) = \frac{1}{N-1}\sum_{s=1}^{N}(L_{i,j(i)}(x,\xi_s^i) - \frac{1}{N}\sum_{k=1}^{N}(L_{i,j(i)}(x,\xi_k^i))^2.$$

The confidence level that is used for numerical testing is retained at 0.95, therefore $\alpha_\delta \approx 1.96$. The reason for taking (6.7) is the fact that it can be shown, by using the Delta method [55, 59], that $\sqrt{N}(f(x) - \hat{f}_N(x))$ converges in distribution towards the random variable with the normal distribution with mean zero and variance equal to $\frac{1}{N^2}\sum_{i=1}^{r_a}\frac{\sigma^2_{i,j(i)}(x)}{P^2_{i,j(i)}(x)}$.

Let us briefly analyze the convergence conditions for the adjusted algorithm. First of all, notice that for every N, function $\hat{f}_N$ is nonnegative and thus the lower bound in Lemma 4.1.2 is zero. Assumption A1 can be reformulated in the following way

A1' For every N, $\quad \hat{f}_N \in C^1(\mathbb{R}^n)$.

The following result holds.

Theorem 6.1.1 *Suppose that the assumptions A1' and B1 are satisfied. Furthermore, suppose that there exist a positive constant κ and number $n_1 \in \mathbb{N}$ such that $\varepsilon_\delta^{N_k}(x_k) \geq \kappa$ for every $k \geq n_1$ and that the sequence $\{x_k\}_{k\in\mathbb{N}}$ generated by the adjusted Algorithm 1 is bounded. Then, either the adjusted Algorithm 1 terminates after a finite number of iterations at a stationary point of $\hat{f}_{N_{max}}$ or every accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$ is a stationary point of $\hat{f}_{N_{max}}$.*

The test problem is generated as follows. We consider five alternatives with five characteristics for each alternative. Thus we generate the matrix M from $\mathbb{R}^{5\times5}$ using the standard normal distribution such that each column of M represents the characteristics of one of the alternatives. The number of agents is assumed to be 500. So the matrix $B \in \mathbb{R}^{5\times500}$ is generated with $B_{ij} : \mathcal{N}(0.5, 1)$ and each column of that matrix represents one realization of the random vector β^i. Finally, the matrix of random terms ε_{ij} from $\mathbb{R}^{5\times500}$ is formed such that each component is a realization of a random variable with the Gumbel distribution with parameters 0 and 1. These three matrices are used to find the vector of choices for 500 agents.

The results presented in Table 6.11 are obtained after 10 independent runs of each algorithm. At each run, the initial approximation is set to be $x_0 = (0.1, \ldots, 0.1)^T$. The maximal sample size for each agent is $N_{max} = 500$. Since we use independent samples across the agents,

	NG			BFGS		
Algorithm	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%	$\|\nabla \hat{f}_{N_{max}}\|$	ϕ	%
$\rho = -\infty$	0.00887	3.98E+07	-8.50	0.00550	5.65E+07	25.16
$\rho = 0.7$	0.00896	4.36E+07	0.00	0.00523	4.52E+06	0.00
Heur	0.00842	1.09E+08	151.68	0.00674	1.53E+07	237.94
SAA	0.00929	8.07E+07	85.41	0.00810	1.82E+07	303.98

Table 6.11: Mixed Logit problem. Mixed Logit problem.

the total maximal sample size is 250 000. Thus, this is the number of realizations of random vector ξ which are generated at the beginning of the optimization process. In algorithms with variable sample size, the starting sample size for each agent is $N_0^{min} = 3$. The other parameters are set as in the previous subsection.

According to ϕ columns, the algorithms with variable sample size strategy once again perform better than their fixed-size counterparts. The heuristic method does not perform well in this example. Notice also that the safeguard rule implies the decrease of the average number of function evaluations significantly in the case of the BFGS method but it produces a relatively small negative effect for the NG method, increasing the number of function evaluations. In the case of the BFGS method the decrease in the sample size is implied in 16.67% of iterations but the safeguard rule declines the decrease in 58.33% of these iterations. For the NG method the corresponding numbers are 26.84% and 20.88%.

6.2 Nonmonotone line search rules

The results from the previous section suggest that the variable sample size and the safeguard rule have positive effect on the algorithm performance. In this section we apply Algorithm 4 with the safeguard

proposed in Algorithm 3 and compare six different line search methods with different search directions. In the first subsection, we consider the problems from [73] which are transformed to include the noise. The second subsection is devoted to a problem which includes real data. The data is collected from a survey that examines the influence of the various factors on the metacognition and the feeling of knowing of the students (Ivana Rančić, project "Quality of Educational System in Serbia in the European perspective", OI 179010, supported by the Ministry of Education, Science and Technological Development, Republic of Serbia). The number of the examined students is 746. The linear regression is used as the model and the least squares problem is considered. This is the form of the objective function which is considered in [26] and therefore we compare Algorithm 4 with the scheme proposed in that paper.

Algorithm 4 is implemented with the stopping criterion $\|g_k^{N_{max}}\| \leq 0.1$ where $g_k^{N_{max}}$ is an approximation or the true gradient of the function $\hat{f}_{N_{max}}$. The maximal sample size for the first set of test problems is $N_{max} = 100$ and the initial sample size is $N_0 = 3$. Alternatively, the algorithm terminates if 10^7 function evaluations is exceeded. When the true gradient is used every component is counted as one function evaluation. In the first subsection, the results are obtained from eight replications of each algorithm and the average values are reported. In the second subsection, the problem does not have that kind of noise included and therefore one replication is sufficient. All the algorithms use the backtracking technique where the decreasing factor of the step size is $\beta = 0.5$. The parameters from Algorithm 2 are $\nu_1 = 0.1$ and $d = 0.5$. The confidence level is $\delta = 0.95$ which leads us to the lack of precision parameter $\alpha_\delta = 1.96$.

We list the line search rules as follows. The rules where the parameter $\tilde{\eta}_k = 0.85$ is given refer to $\tilde{C}_k$ defined by (5.4), while $M = 10$ determines the rule with $\tilde{C}_k$ defined by (5.8). The choice for this pa-

rameters is motivated by the work of [71] and [17]. We denote the approximation of the gradient $\nabla \hat{f}_{N_k}(x_k)$ by g_k. When the true gradient is available, $g_k = \nabla \hat{f}_{N_k}(x_k)$.

(B1) $\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \hat{f}_{N_k}(x_k) + \eta \alpha_k p_k^T g_k$

(B2) $\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \hat{f}_{N_k}(x_k) + \varepsilon_k - \alpha_k^2 \beta_k$

(B3) $\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \varepsilon_k - \alpha_k^2 \beta_k, \quad \tilde{\eta}_k = 0.85$

(B4) $\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \eta \alpha_k p_k^T g_k, \quad M = 10$

(B5) $\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \varepsilon_k - \alpha_k^2 \beta_k, \quad M = 10$

(B6) $\hat{f}_{N_k}(x_k + \alpha_k p_k) \leq \tilde{C}_k + \eta \alpha_k p_k^T g_k, \quad \tilde{\eta}_k = 0.85$

The rules B1, B4 and B6 assume the descent search directions and the parameter η is set to 10^{-4}. The initial member of the sequence which makes the nondescent directions acceptable is defined by $\varepsilon_0 = \max\{1, |\hat{f}_{N_0}(x_0)|\}$ while the rest of it is updated by $\varepsilon_k = \varepsilon_0 k^{-1.1}$ but only if the sample size does not change, i.e. if $N_{k-1} = N_k$. Otherwise, $\varepsilon_k = \varepsilon_{k-1}$. Furthermore, we define $\beta_k = |g_k^T H_k g_k|$ where H_k is the approximation of the inverse Hessian of the function $\hat{f}_{N_k}$ at the point x_k.

The search directions are of the form

$$p_k = -H_k g_k.$$

We make 4 different choices for the matrix H_k and obtain the following directions.

(NG) The negative gradient direction is obtained by setting $H_k = I$ where I represents the identity matrix.

(BFGS) This direction is obtained by using the BFGS formula for updating the inverse Hessian

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

where $y_k = g_{k+1} - g_k$, $s_k = x_{k+1} - x_k$ and $H_0 = I$.

(SG) The spectral gradient direction is defined by setting $H_k = \gamma_k I$ where

$$\gamma_k = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}.$$

(SR1) The symmetric rank-one direction is defined by $H_0 = I$ and

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

If the true gradient is available, the negative gradient is the descent search direction. Moreover, the BFGS and the SG implementations also ensure descent search direction. This issue is addressed in subsection 2.2.1 while a more detailed explanation is available at [46] and [63] for example.

We also tested the algorithm with the following gradient approximations. FD stands for the centered finite difference estimator while FuN represents the simultaneous perturbations approximation that allows the standard normal distribution for the perturbation sequence [27].

(FD) For $i = 1, 2, \ldots, n$

$$(g_k)_i = \frac{\hat{f}_{N_k}(x_k + he_i) - \hat{f}_{N_k}(x_k - he_i)}{2h},$$

where e_i is the ith column of the identity matrix and $h = 10^{-4}$.

(FuN) For $i = 1, 2, \ldots, n$

$$(g_k)_i = \frac{\hat{f}_{N_k}(x_k + h\Delta_k) - \hat{f}_{N_k}(x_k - h\Delta_k)}{2h}\Delta_{k,i},$$

where $h = 10^{-4}$ and random vector $\Delta_k = (\Delta_{k,1}, ..., \Delta_{k,n})^T$ follows the multivariate standard normal distribution.

The criterion for comparing the algorithms is the number of function evaluations as in the previous section.

6.2.1 Noisy problems

We use 7 test functions from the test collection [24] available at the web page [73]: Freudenstein and Roth, Jennrich and Sampson, Biggs EXP6, Osborne II, Trigonometric, Broyden Tridiagonal and Broyden Banded. They are converted into noisy problems in two ways. The first one is by adding the noise, and the second one involves multiplication by a random vector which then affects the gradient as well. The noise is represented by the random vector ξ with the normal distribution $\mathcal{N}(0, 1)$. If we denote the deterministic test function by $q(x)$, we obtain the objective functions $f(x) = E(F(x, \xi))$ in the following two manners:

(N1) $F(x, \xi) = q(x) + \xi$

(N2) $F(x, \xi) = q(x) + \|\xi x\|^2$.

This provides us with 14 test problems. The average number of function evaluations in 8 replications is used as the main criterion. Let us denote it by ϕ_i^j where i represents the method determined by the line search and the search direction and j represents the problem. We define the efficiency index like in Krejić, Rapajić [39], i.e. for the method

	Efficiency index (ω)				Nonmonotonicity index (μ)			
	NG	SG	BFGS	SR1	NG	SG	BFGS	SR1
B1	0.2471	0.3975	0.5705	/	0.0000	0.0000	0.0000	/
B2	0.0774	0.4780	0.5474	0.4750	0.4835	0.2081	0.1616	0.2541
B3	0.0783	0.4927	0.5306	0.4401	0.4426	0.2083	0.1708	0.2810
B4	0.0620	0.6468	0.4200	/	0.4070	0.1049	0.0998	/
B5	0.0798	0.5157	0.5043	0.4725	0.4060	0.1998	0.1722	0.2593
B6	0.1064	0.6461	0.4690	/	0.3430	0.1050	0.0944	/

Table 6.12: The gradient-based methods. Gradijentni metodi.

i the efficiency index is

$$\omega_i = \frac{1}{14} \sum_{j=1}^{14} \frac{\min_i \phi_i^j}{\phi_i^j}.$$

We also report the level of nonmonotonicity. If the number of iterations is k and s is the number of iterations at which the accepted step size would not pass through if the line search rule has been B1, then we define the nonmonotonicity index by

$$\mu = \frac{s}{k}.$$

The numbers in the following two tables refer to the average values of 8 independent runs. Table 6.12 represents the results obtained by applying the methods with the true gradient, while the subsequent table refers to the gradient approximation approach. The SR1 method is not tested with the line search rules which assume descent search directions and therefore the efficiency index is omitted in that cases. The same is true for the nonmonotonicity. For the same reason we omit the line search rules B1, B4 and B6 in Table 6.13.

Among the 21 tested methods presented in Table 6.12, the efficiency index suggests that the best one is the spectral gradient method

	Efficiency index (ω)			
	SG-FD	SG-FuN	BFGS-FD	SR1-FD
B2	0.6832	0.4536	0.7316	0.6995
B3	0.6957	0.4164	0.7149	0.6576
B5	0.7255	0.4286	0.6808	0.7156
	Nonmonotonicity index (μ)			
	SG-FD	SG-FuN	BFGS-FD	SR1-FD
B2	0.1693	0.1008	0.1349	0.2277
B3	0.1682	0.1166	0.1449	0.2516
B5	0.1712	0.1248	0.1453	0.2410

Table 6.13: The gradient-free methods. Metodi bez gradijenata.

combined with the line search rule B4. However, we can see that the results also suggest that the negative gradient and the BFGS search direction should be combined with the monotone line search rule B1. The SR1 method works slightly better with the line search B2 than with B5 and we can say that it is more efficient with the lower level of nonmonotonicity. If we look at the SG method, we can conclude that large nonmonotonicity is not beneficial for that method either. In fact, B4 has the lowest nonmonotonicity if we exclude B1.

The results considering the spectral gradient method are consistent with the deterministic case because it is known that the monotone line search can inhibit the benefits of scaling the negative gradient direction. However, these testings suggest that allowing too much nonmonotonicity can be bad for the performance of the algorithms.

The results from Table 6.13 imply that B5 is the best choice if we consider the spectral gradient or SR1 method with the finite difference gradient approximation. Furthermore, this kind of approximation combined with the BFGS direction performs the best with the line search B2. This line search is the best choice for simultaneous perturbation approach as well. However, this approximation of the

gradient provided the least preferable results in general. This was expected because the simultaneous perturbation provided rather poor approximations of the gradient. Also, the number of iterations was not that large in general and the asymptotic features of that approach could not develop.

The formulation of the problem where we add the noise term was suitable for examining the convergence towards the local/global optimum. However, the numerical results have not yielded useful information. Moreover, if we consider the spectral gradient method, the results are not as it was expected: there is no clear evidence that the nonmonotone line search methods converge more frequently to a global solution than their monotone counterparts. In fact, in the Freudenstein and Roth problem for example, B1 method converged to the global minimum in all 8 replications, B6 converged to the global minimum only once while the other methods were trapped at the local solutions. Furthermore, in Broyden Banded problem, B4 and B6 were carried away from the global solution, while the other ones converged towards it.

The case where the noise affects the gradient was harder for tracking the global optimum. However, we captured that the SG method with the line searches that allow only the descent directions (B1, B4 and B6) converged to the point with the lower function value when the problem Broyden Tridiagonal is concerned. Furthermore, in problem Osborne II, SG with the Armijo line search B1 provided the lowest function value.

The efficiency index yields similar conclusions as the performance profile analysis. At the end of this subsection, we show the performance profile graphics for the methods that performed the best: SG in the gradient-based case (Figure 6.4) and BFGS-FD in the gradient-free case (Figure 6.5). The first graphic in both figures provides the results when the problems of the form (N1) are considered, the second one refers to the problems (N2) while the third one gathers all 14

problems together.

Figure 6.4 shows that B4 clearly outperforms all the other line search rules in (N1) case, while in (N2) case B6 is highly competitive. If we look at all the considered problems together, B4 is clearly the best choice. In the BFGS-FD case, B2 and B3 seem to work better than B5 and the advantage is on the side of B2. Moreover, the performance profile suggests that this advantage is gained in the case where the noise affects the search direction, i.e. when (N2) formulation is considered.

6.2.2 Application to the least squares problems

As we already mentioned, this subsection is devoted to the real data problem. The data comes from a survey that was conducted among 746 students in Serbia. The goal of this survey was to determine how do different factors affect the feeling of knowing (FOK) and metacognition (META) of the students. We will not go into further details of this survey since our aim is only to compare different algorithms. Our main concern is the number of function evaluations needed for solving the problem rather than the results of this survey. Therefore, we only present the number of function evaluations (ϕ) and nonmonotonicity index (μ) defined above.

Linear regression is used as the model and the parameters are searched for throughout the least squares problem. Therefore, we obtain two problems of the form $\min_{x\in\mathbb{R}^n} \hat{f}_N(x)$ where

$$\hat{f}_N(x) = \frac{1}{N}\sum_{i=1}^{N}(x^T a_i - y_i)^2.$$

The sample size is $N = N_{max} = 746$ and the number of factors examined is $n = 4$. Vectors a_i, $i = 1, 2, \ldots, 746$ represent the factors and

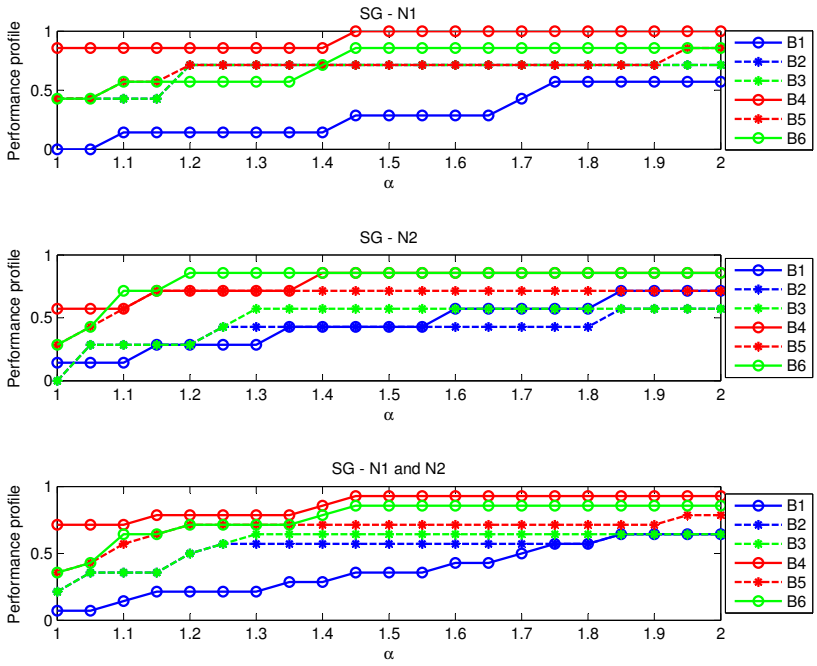

Figure 6.4: The SG methods in noisy environment. SG metodi u stohastičkom okruženju.

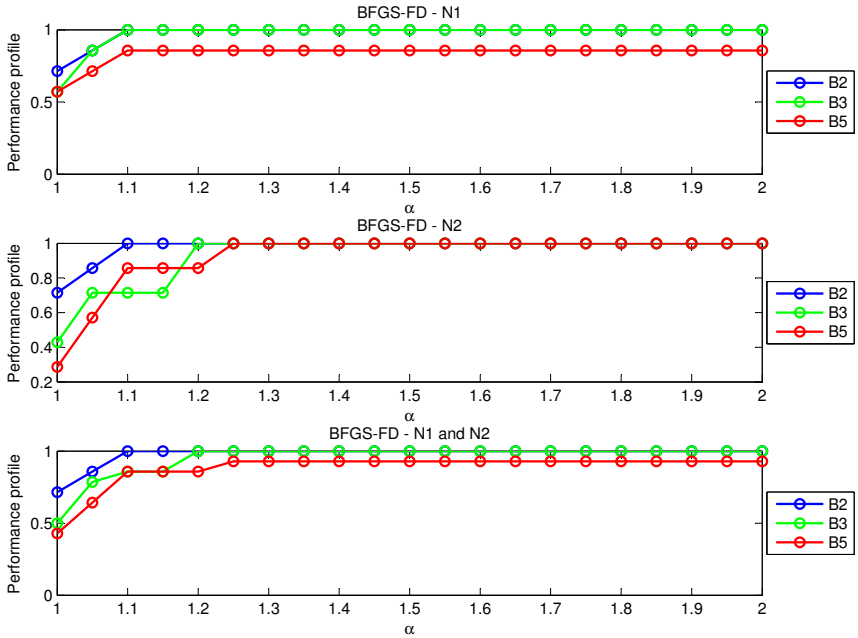

Figure 6.5: The BFGS-FD methods in noisy environment. BFGS metodi u stohastičkom okruženju.

	Algorithm 4		Heuristic	
SG	ϕ	μ	ϕ	μ
B1	9.4802E+04	0.0000	1.2525E+05	0.0000
B2	5.3009E+04	0.2105	6.0545E+04	0.2105
B3	5.3009E+04	0.2105	6.0545E+04	0.2105
B4	4.4841E+04	0.1765	9.4310E+04	0.2121
B5	5.3009E+04	0.2105	7.1844E+04	0.1967
B6	4.5587E+04	0.1176	1.1178E+05	0.1343

Table 6.14: The FOK analysis results. Rezultati FOK analize.

y_i, $i = 1, 2, \ldots, 746$ represent the FOK or the META results obtained from the survey.

The same type of problem is considered in [26]. Therefore, we wanted to compare the variable sample size scheme proposed in this thesis with the dynamics of increasing the sample size that is proposed in [26] (Heuristic). We state the results in Table 6.14 and Table 6.15. Heuristic assumes that the sample size increases in the following manner $N_{k+1} = \lceil \min\{1.1N_k, N_{max}\} \rceil$. Since the gradients are easy to obtain, we chose to work with the gradient-based approach and we use the spectral gradient method with the different line search rules to obtain the following results. The Algorithm 4 is used with the same parameters like in the previous subsection and the stopping criterion $\|g_k^{N_{max}}\| \leq 10^{-2}$.

First of all notice that the Algorithm 4 performs better than the Heuristic in all cases. Also, the monotone line search B1 performs the worst in both problems and both presented algorithms. When the FOK problem is considered, the best results are obtained with the line search B4 applied within the Algorithm 4, although B6 is highly competitive in that case. Both of the mentioned line search rules have modest nonmonotonicity coefficients. However, when Heuristic is applied, the additional term ε_k turns out to be useful since the best

	Algorithm 4		Heuristic	
SG	ϕ	μ	ϕ	μ
B1	1.6716E+05	0.0000	2.1777E+05	0.0000
B2	3.3606E+04	0.0909	6.2159E+04	0.2632
B3	3.3606E+04	0.0909	6.1408E+04	0.1897
B4	3.8852E+04	0.1538	6.6021E+04	0.1607
B5	3.3606E+04	0.0909	6.1408E+04	0.1897
B6	3.8852E+04	0.1538	1.4953E+05	0.1053

Table 6.15: The META analysis results. Rezultati META analize.

performance is obtained by B2 and B3.

While the analysis of FOK provided the results similar to the ones in the previous subsection, the META yielded rather different conclusions. In that case, the lowest number of function evaluations was achieved by the line search rules B2, B3 and B5. However, the results are not that different because the level of nonmonotonicity for those methods was not the highest detected among the line searches. Similar results are obtained for Heuristic where B3 and B5 are the best with the medium level of nonmonotonicity.

Bibliography

[1] S. Andradottir, A review of simulation optimization techniques, *Proceedings of the 1998 Winter Simulation Conference, (1998), pp. 151-158.*

[2] J. Barzilai, J. M. Borwein, Two-point step size gradient methods, *IMA Journal of Numerical Analysis, 8 (1988), pp. 141-148.*

[3] F. Bastin, Trust-Region Algorithms for Nonlinear Stochastic Programming and Mixed Logit Models, *PhD thesis, University of Namur, Belgium, 2004.*

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

[5] F. Bastin, C. Cirillo, P. L. Toint, Convergence theory for nonconvex stochastic programming with an application to mixed logit, *Mathematical Programming, Ser. B 108 (2006), pp. 207-234.*

[6] A. Benveniste, M. Metivier, P. Priouret, Adaptive Algorithms and Stochastic Approximations, *Springer-Verlag, New York, Vol. 22, 1990.*

[7] E. G. Birgin, N. Krejić, J. M. Martínez, Globaly convergent inexact quasi-Newton methods for solving nonlinear systems, *Numerical Algorithms, 32 (2003), pp. 249-260.*

[8] E. G. Birgin, J. M. Martínez, A Spectral Conjugate Gradient Method for Unconstrained Optimization, *Applied Mathematics and Optimization, Vol. 43, Issue 2 (2001), pp. 117-128.*

[9] E. G. Birgin, J. M. Martínez, M. Raydan, Nonmonotone Spectral Projected Gradient Methods on Convex Sets, *SIAM Journal on Optimization, Vol. 10, Issue 4 (2006), pp.11961211.*

[10] I. Bongartz, A. R. Conn, N. Gould, Ph. L. Toint, CUTE: Constrained and unconstrained testing environment, *ACM Transactions on Mathematical Software, 21 (1995), pp. 123-160.*

[11] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal, H. C. Pedersen, The watchdog technique for forcing convergence in algorithms for constrained optimization, *Mathematical Programming Studies, 16 (1982), pp. 1-17*

[12] W. Cheng, D.H. Li, A derivative-free nonmonotone line search and its applications to the spectral residual method, *IMA Journal of Numerical Analysis, 29 (2008), pp. 814-825*

[13] A. R. Conn, N. I. M. Gould, Ph. L. Toint, Trust-region Methods, *SIAM, 2000.*

[14] A. R. Conn, K. Scheinberg, Ph. L. Toint, Recent progress in unconstrained nonlinear optimization without derivatives, *Mathematical Programming, Vol. 79, Issue 1-3 (1997), pp. 397-414 .*

[15] A. R. Conn, K. Scheinberg, Ph. L. Toint, A derivative free optimization algorithm in practice, *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, (1998).*

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

[17] Y.H. Dai, On the nonmonotone line search, *Journal of Optimization Theory and Applications, 112 (2002), pp. 315-330.*

[18] B. Delyon, A. Juditsky, Accelerated stochastic approximation, *SIAM Journal on Optimization, Vol.3, No.4 (1993), pp. 868-881.*

[19] S. S. Drew, T. Homem-de-Mello, Quasi-Monte Carlo strategies for stochastic optimization, *Proceedings of the 38th Winter Conference on Simulation, (2006), pp. 774-782 .*

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

[21] V. Fabian, On Asymptotic Normality in Stochastic Optimization, *Annals of Mathematical Statistics, Vol. 39 (1968), pp. 1327-1332.*

[22] G. Deng, M. C. Ferris, Variable-Number Sample Path Optimization, *Mathematical Programming, Vol. 117, No. 1-2 (2009), pp. 81-109.*

[23] M.A. Diniz-Ehrhardt, J. M. Martínez, M. Raydan, A derivative-free nonmonotone line-search technique for uncon-

strained optimization, *Journal of Computational and Applied Mathematics, Vol. 219, Issue 2 (2008), pp. 383-397.*

[24] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming, Ser. A 91 (2002), pp. 201-213*

[25] G. E. Forsythe, On the asymptotic directions of the s-dimensional optimum gradient method, *Numerische Mathematik, 11 (1968), pp. 57-76.*

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

[27] M. C. Fu, Gradient Estimation, *S.G. Henderson and B.L. Nelson (Eds.), Handbook in OR & MS, Vol. 13 (2006), pp. 575-616.*

[28] M. C. Fu, Optimization via simulation: A review, *Annals of Operational Research 53 (1994), pp. 199-247.*

[29] A. Griewank, The globall convergence of Broyden-like methods with a suitable line search, *Journal of the Australian Mathematical Society, Ser. B 28 (1986), pp. 75-92.*

[30] L. Grippo, F. Lampariello, S. Lucidi, A nononotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis, Vol. 23, No. 4 (1986), pp. 707-716.*

[31] L. Grippo, F. Lampariello, S. Lucidi, A class of nonmonotone stabilization methods in unconstrained optimization, *Numerical Mathematics, 59 (1991), pp. 779-805.*

[32] T. Homem-de-Mello, On rates of convergence for stochastic optimization problems under non-independent and identically

distributed sampling, *SIAM Journal on Optimization, Vol. 19, No. 2 (2008), pp. 524551.*

[33] T. Homem-de-Mello, Variable-Sample Methods for Stochastic Optimization, *ACM Transactions on Modeling and Computer Simulation, Vol. 13, Issue 2 (2003), pp. 108-133.*

[34] C. Kao, S. Chen, A stochastic quasi-Newton method for simulation response optimization, *European Journal of Operational Research, 173 (2006), pp. 30-46.*

[35] C. Kao, W. T. Song, S. Chen, A modified Quasi-Newton Method for Optimization in Simulation, *International Transactions in Operational Research, Vol.4, No.3 (1997), pp. 223-233.*

[36] H. Kesten, Accelerated stochastic approximation, *The Annals of Mathematical Statistics, 29 (1958), pp. 41-59.*

[37] N. Krejić, N. Krklec, Line search methods with variable sample size for unconstrained optimization, *Journal of Computational and Applied Mathematics, Vol. 245 (2013), pp. 213-231.*

[38] N. Krejić, N. Krklec Jerinkić, Nonmonotone line search methods with variable sample size, *Technical report (2013).*

[39] N. Krejić, S. Rapajić, Globally convergent Jacobian smoothing inexact Newton methods for NCP, *Computational Optimization and Applications, Vol. 41, Issue 2 (2008), pp. 243-261.*

[40] W. La Cruz, J. M. Martínez, M. Raydan, Spectral residual method withouth gradient information for solving large-scale nonlinear systems of equations, *Mathematics of Computation, 75 (2006), pp. 1429-1448.*

[41] D. H. Li, M. Fukushima, A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, *Optimization Methods and Software, 13 (2000), pp. 181-201.*

[42] S. Lucidi, M. Sciandrone, On the global convergence of derivative-free methods for unconstrained optimization, *SIAM Journal on Optimization, 13 (2002), pp. 97-116.*

[43] K. Marti, Solving Stochastical Structural Optimization Problems by RSM-Based Stochastic Approximation Methods - Gradient Estimation in Case of Intermediate Variables, *Mathematical Methods of Operational Research 46 (1997), pp. 409-434.*

[44] M. Montaz Ali, C. Khompatraporn, Z. B. Zabinsky, A Numerical Evaluation of Several Stochastic Algorithms on Selected Continous Global Optimization Test Problems, *Journal of Global Optimization, Vol. 31, Issue 4 (2005), pp.635-672 .*

[45] J. J. Moré, S. M. Wild, Benchmarking derivative-free optimization algorithms, *SIAM Journal on Optimization, Vol. 20, No. 1 (2009), pp. 172-191.*

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

[47] R. Pasupathy, On choosing parameters in retrospective-approximation algorithms for simulation-optimization, *proceedings of the 2006 Winter Simulation Conference, L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol and R.M. Fujimoto, eds., pp. 208-215.*

[48] R. Pasupathy, On Choosing Parameters in Retrospective-Approximation Algorithms for Stochastic Root Finding and

Simulation Optimization, *Operations Research Vol. 58, No. 4 (2010), pp. 889-901.*

[49] S. Pilipović, D. Seleši, Mera i integral - fundamenti teorije verovatnoće, *Zavod za udžbenike, Beograd, 2012* .

[50] E. Polak, J. O. Royset, Eficient sample sizes in stochastic nonlinear programing, *Journal of Computational and Applied Mathematics, Vol. 217, Issue 2 (2008), pp. 301-310.*

[51] M. J. D. Powell, UOBYQA: Unconstrained optimization by quadratic approximation, *Mathematical Programming, 92 (2002), pp. 555-582.*

[52] D. Rajter-Ćirić, Verovatnoća, *Univerzitet u Novom Sadu, Prirodno-matematički fakultet, 2009.*

[53] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization, 7 (1997), pp. 26-33.*

[54] J. O. Royset, Optimality functions in stochastic programming, *Mathematical Programming, Vol. 135, Issue 1-2 (2012), pp. 293-321*

[55] R.Y. Rubinstein, A. Shapiro, Discrete Event Systems, *John Wiley & Sons, Chichester, England, 1993.*

[56] J. Sacks, Asymptotic distribution of stochastic approximation procedures, *Annals of Mathematical Statistics, Vol. 29 (1958), pp. 373-405.*

[57] A. Shapiro, Asymptotic Properties of Statistical Estimators in Stochastic Programming, *The Annals of Statistics, Vol. 17, No. 2 (1989), pp. 841-858*

[58] A. Shapiro, T. Homem-de-Mello, On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs, *SIAM Journal on Optimization, Vol. 11, No. 1 (2000), pp. 70-86.*

[59] A. Shapiro, A. Ruszczynski, Stochastic Programming, *Vol. 10 of Handbooks in Operational Research and Management science, Elsevier, 2003, pp. 353-425.*

[60] A. Shapiro, Y. Wardi, Convergence Analysis of Stochastic Algorithms, *Mathematics of Operations Research, INFORMS, Vol. 21, No. 3 (1996), pp. 615-628.*

[61] J. C. Spall, Adaptive stochastic approximation by the simultaneous perturbation method, *IEEE Transactions and Automatic Control, Vol. 45, No. 10 (2000), pp. 1839-1853.*

[62] J. C. Spall, Introduction to Stochastic Search and Optimization, *Wiley-Interscience Series in Discrete Mathematics, New Jersey, 2003.*

[63] R. Tavakoli, H. Zhang, A nonmonotone spectral projected gradient method for large-scale topology optimization problems, *Numerical Algebra, Control and Optimization Vol. 2, No. 2 (2012), pp. 395-412.*

[64] P. L. Toint, An assessment of nonmonotone line search techniques for unconstrained optimization, *SIAM Journal of Scientific Computing Vol. 17, No. 3 (1996), pp. 725-739.*

[65] Y. Wardi, A Stochastic Algorithm Using One Sample Point per Iteration and Diminishing Stepsizes , *Journal of Optimization Theory and Applications, Vol. 61, No. 3 (1989), pp. 473-485.*

[66] Y. Wardi, A Stochastic Steepest-Descent Algorithm, *Journal of Optimization Theory and Applications, Vol. 59, No. 2 (1988), pp. 307-323.*

[67] Y. Wardi, Stochastic Algorithms with Armijo Stepsizes for Minimization of Functions , *Journal of Optimization Theory and Applications, Vol. 64, No. 2 (1990), pp. 399-417.*

[68] D. H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation, Vol. 1 (1997), pp. 67-82.*

[69] D. Yan, H. Mukai,, Optimization Algorithm with Probabilistic Estimation, *Journal of Optimization Theory and Applications, Vol. 79, No. 2 (1993), pp. 345-371.*

[70] M. Zervos, On the Epiconvergence of Stochastic Optimization Problems, *Mathematics of Operations Research, INFORMS, Vol. 24, No. 2 (1999), pp. 495-508.*

[71] H. Zhang, W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM Journal on Optimization, 4 (2004), pp. 1043-1056.*

[72] H. Zhang, W. Yin, Gradient methods for convex minimization: better rates under weaker conditions, *Technical report, Optimization and Control, (2013).*

[73] http://www.uni-graz.at/imawww/kuntsevich/solvopt/results /moreset.html.

.

Biography

I was born on 9th of September 1984 in Novi Sad where I attended the elementary school "Žarko Zrenjanin" and the high school "Svetozar Marković". In 2003, I became a student of Mathematics of finance at the Faculty of Sciences, University of Novi Sad. I graduated in 2007 with the average grade 9.59.

After graduating, in November 2007 I became a PhD student on the same university in the field of Numerical mathematics. By September 2010 I passed all the exams with the average grade 10.00. Since 2008 I have been holding tutorials at the Department of Mathematics and Informatics at the University of Novi Sad. I held courses of Numerical Analysis, Software Practicum, Probability Theory, Partial Differential Equations and Financial Mathematics. I also held tutorials of Actuarial Mathematics at the National Bank of Serbia (2007- 2008) and at the Faculty of Sciences, University of Novi Sad (2009-2012).

I participated at the project "Numerical Methods for Nonlinear Mathematical Models" supported by the Serbian Ministry of Science and Environment Protection between 2008 and 2011. During that period I had been receiving the scholarship of the Serbian Ministry of Education and Technological Development. Since February 2011 I am a research assistant at the project "Numerical Methods, Simulations and Applications" supported by the Serbian Ministry of Education and Science.

Novi Sad, June 6, 2013 Nataša Krklec Jerinkić

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 thesis
CC
Author: Nataša Krklec Jerinkić
AU
Mentor: Prof. Dr. Nataša Krejić
MN

Title: Line search methods with variable sample size
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: 2013
PY
Publisher: Author's reprint
PU
Publication place: Novi Sad, Faculty of Sciences, Trg Dositeja Obradovića 4
PP
Physical description: 6/222/73/15/0/5/0
(chapters/pages/literature/tables/pictures/graphics/appendices)
PD
Scientific field: Mathematics
SF
Scientific discipline: Numerical mathematics
SD
Subject / Key words: Nonlinear optimization, line search methods, nonmonotone line search, sample average approximation, variable sample size, stochastic optimization.
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 optimization problem with the objective function in the form of mathematical expectation. The expectation is with respect to the random variable

that represents the uncertainty. Therefore, the objective function is in fact deterministic. However, finding the analytical form of that objective function can be very difficult or even impossible. This is the reason why the sample average approximation is often used. In order to obtain reasonable good approximation of the objective function, we have to use relatively large sample size. We assume that the sample is generated at the beginning of the optimization process and therefore we can consider this sample average objective function as the deterministic one. However, applying some deterministic method on that sample average function from the start can be very costly. The number of evaluations of the function under expectation is a common way of measuring the cost of an algorithm. Therefore, methods that vary the sample size throughout the optimization process are developed. Most of them are trying to determine the optimal dynamics of increasing the sample size.

The main goal of this thesis is to develop the clas of methods that can decrease the cost of an algorithm by decreasing the number of function evaluations. The idea is to decrease the sample size whenever it seems to be reasonable - roughly speaking, we do not want to impose a large precision, i.e. a large sample size when we are far away from the solution we search for. The detailed description of the new methods is presented in Chapter 4 together with the convergence analysis. It is shown that the approximate solution is of the same quality as the one obtained by dealing with the full sample from the start.

Another important characteristic of the methods that are proposed here is the line search technique which is used for obtaining the subsequent iterates. The idea is to find a suitable direction and to search along it until we obtain a sufficient decrease in the function value. The sufficient decrease is determined throughout the line search rule. In Chapter 4, that rule is supposed to be monotone, i.e. we are imposing strict decrease of the function value. In order to decrease the cost of the algorithm even more and to enlarge the set of suitable search di-

rections, we use nonmonotone line search rules in Chapter 5. Within that chapter, these rules are modified to fit the variable sample size framework. Moreover, the conditions for the global convergence and the R-linear rate are presented.

In Chapter 6, numerical results are presented. The test problems are various - some of them are academic and some of them are real world problems. The academic problems are here to give us more insight into the behavior of the algorithms. On the other hand, data that comes from the real world problems are here to test the real applicability of the proposed algorithms. In the first part of that chapter, the focus is on the variable sample size techniques. Different implementations of the proposed algorithm are compared to each other and to the other sample schemes as well. The second part is mostly devoted to the comparison of the various line search rules combined with different search directions in the variable sample size framework. The overall numerical results show that using the variable sample size can improve the performance of the algorithms significantly, especially when the nonmonotone line search rules are used.

The first chapter of this thesis provides the background material for the subsequent chapters. In Chapter 2, basics of the nonlinear optimization are presented and the focus is on the line search, while Chapter 3 deals with the stochastic framework. These chapters are here to provide the review of the relevant known results, while the rest of the thesis represents the original contribution.

AB

Accepted by Scientific Board on: November 17, 2011
ASB
Defended:
DE

Thesis defend board:
President: Zorana Lužanin, PhD, Full Professor, Faculty of Sciences, University of Novi Sad

Member: Nataša Krejić, PhD, Full Professor, Faculty of Sciences, University of Novi Sad

Member: Stefania Bellavia, PhD, Associate Professor, Department of Industrial Engineering, University of Florence

Member: Miodrag Spalević, PhD, Full Professor, Faculty of Mechanical Engineering, University of Belgrade

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 teza
VR
Autor: Nataša Krklec Jerinkić
AU
Mentor: Prof. dr Nataša Krejić
MN

Naslov rada: Metodi linijskog pretraživanja sa promenljivom veličinom uzorka
NR

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

Zemlja publikovanja: Republika Srbija
ZP
Uže geografsko područje: Vojvodina
UGP
Godina: 2013.
GO
Izdavač: Autorski reprint
IZ
Mesto i adresa: Novi Sad, Prirodno-matematički fakultet, Trg Dositeja Obradovića 4
MA
Fizički opis rada: 6/222/73/15/0/5/0
(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, metodi linijskog pretraživanja, nemonotono linijsko pretraživanje, uzoračko očekivanje, promenljiva veličina uzorka, stohastička optimizacija.
PO

UDK:
Čuva se: u biblioteci Departmana za matematiku i informatiku, Novi Sad
ČU
Važna napomena:
VN

Izvod:
U okviru ove teze posmatra se problem optimizacije bez ograničenja pri čemu je funkcija cilja u formi matematičkog očekivanja. Očekivanje se odnosi na slučajnu promenljivu koja predstavlja neizvesnost. Zbog toga je funkcija cilja, u stvari, deterministička veličina. Ipak, odredjivanje analitičkog oblika te funkcije cilja može biti vrlo komplikovano pa čak i nemoguće. Zbog toga se za aproksimaciju često koristi uzoračko očekivanje. Da bi se postigla dobra aproksimacija, obično je neophodan obiman uzorak. Ako pretpostavimo da se uzorak realizuje pre početka procesa optimizacije, možemo posmatrati uzoračko očekivanje kao determinističku funkciju. Medjutim, primena nekog od determinističkih metoda direktno na tu funkciju može biti veoma skupa jer evaluacija funkcije pod očekivanjem često predstavlja veliki trošak i uobičajeno je da se ukupan trošak optimizacije meri po broju izračunavanja funkcije pod očekivanjem. Zbog toga su razvijeni metodi sa promenljivom veličinom uzorka. Većina njih je bazirana na odredjivanju optimalne dinamike uvećanja uzorka.

Glavni cilj ove teze je razvoj algoritma koji, kroz smanjenje broja izračunavanja funkcije, smanjuje ukupne troškove optimizacije. Ideja je da se veličina uzorka smanji kad god je to moguće. Grubo rečeno, izbegava se korišćenje velike preciznosti (velikog uzorka) kada smo daleko od rešenja. U četvrtom poglavlju ove teze opisana je nova klasa metoda i predstavljena je analiza konvergencije. Dokazano je da je aproksimacija rešenja koju dobijamo bar toliko dobra koliko i za metod koji radi sa celim uzorkom sve vreme.

Još jedna bitna karakteristika metoda koji su ovde razmatrani je primena linijskog pretraživanja u cilju odredjivanja naredne iteracije. Osnovna ideja je da se nadje odgovarajući pravac i da se duž njega vrši pretraga za dužinom koraka koja će dovoljno smanjiti vrednost funkcije. Dovoljno smanjenje je odredjeno pravilom linijskog pretraživanja. U četvrtom poglavlju to pravilo je monotono što znači da zahtevamo striktno smanjenje vrednosti funkcije. U cilju još

većeg smanjenja troškova optimizacije kao i proširenja skupa pogodnih pravaca, u petom poglavlju koristimo nemonotona pravila linijskog pretraživanja koja su modifikovana zbog promenljive veličine uzorka. Takodje, razmatrani su uslovi za globalnu konvergenciju i R-linearnu brzinu konvergencije.

Numerički rezultati su predstavljeni u šestom poglavlju. Test problemi su različiti - neki od njih su akademski, a neki su realni. Akademski problemi su tu da nam daju bolji uvid u ponašanje algoritama. Sa druge strane, podaci koji potiču od stvarnih problema služe kao pravi test za primenljivost pomenutih algoritama. U prvom delu tog poglavlja akcenat je na načinu ažuriranja veličine uzorka. Različite varijante metoda koji su ovde predloženi porede se medjusobno kao i sa drugim šemama za ažuriranje veličine uzorka. Drugi deo poglavlja pretežno je posvećen poredjenu različitih pravila linijskog pretraživanja sa različitim pravcima pretraživanja u okviru promenljive veličine uzorka. Uzimajući sve postignute rezultate u obzir dolazi se do zaključka da variranje veličine uzorka može značajno popraviti učinak algoritma, posebno ako se koriste nemonotone metode linijskog pretraživanja.

U prvom poglavlju ove teze opisana je motivacija kao i osnovni pojmovi potrebni za praćenje preostalih poglavlja. U drugom poglavlju je iznet pregled osnova nelinearne optimizacije sa akcentom na metode linijskog pretraživanja, dok su u trećem poglavlju predstavljene osnove stohastičke optimizacije. Pomenuta poglavlja su tu radi pregleda dosadašnjih relevantnih rezultata dok je originalni doprinos ove teze predstavljen u poglavljima 4-6.
IZ

Datum prihvatanja teme od strane NN Veća: 17.11.2011.
DP

Datum odbrane:
DO
Članovi komisije:
Predsednik: dr Zorana Lužanin, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu

Član: dr Nataša Krejić, redovni profesor, Prirodno-matematički fakultet, Univerzitet u Novom Sadu

Član: dr Stefania Bellavia, vanredni profesor, Departman za industrijsko inženjerstvo, Univerzitet u Firenci

Član: dr Miodrag Spalević, redovni profesor, Mašinski fakultet, Univerzitet u Beogradu
KO