Global Convergence Rate Analysis of a Generic Line Search Algorithm with Noise

A. S. Berahas∗ L. Cao∗ K. Scheinberg†‡

October 10, 2019

Abstract

In this paper, we develop convergence analysis of a modified line search method for objective functions whose value is computed with noise and whose gradient estimates are inexact and possibly random. The noise is assumed to be bounded in absolute value without any additional assumptions. We extend the framework based on stochastic methods from [5] which was developed to provide analysis of a standard line search method with exact function values and random gradients to the case of noisy function. We introduce a condition on the gradient which when satisfied with some sufficiently large probability at each iteration, guarantees convergence properties of the line search method. We derive expected complexity bounds for convex, strongly convex and nonconvex functions.

1 Introduction

We consider an unconstrained optimization problem of the form

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

(1.1)

where $f(x) = \phi(x) + \epsilon(x)$ is computable, while $\phi(x)$ is not. In other words $f : \mathbb{R}^n \to \mathbb{R}$ is a possibly noisy approximation of a smooth function $\phi : \mathbb{R}^n \to \mathbb{R}$, and the goal is to minimize $\phi$. Alternatively, $f(x)$ may be a nonsmooth function and $\phi(x)$ - its smooth approximation, see for instance [11, 14]. Such problems arise in a plethora of fields such as Derivative-Free Optimization (DFO) [6, 10], Simulation Optimization [17] and Machine Learning. There has been a lot of work analyzing the case when $\epsilon(x)$ is a random function with zero mean. Here, we take a different research direction, allowing $\epsilon(x)$ to be stochastic, deterministic, or adversarial, but assuming that $|\epsilon(x)| \leq \epsilon_f$ for all $x$. While this is a strong assumption, it is often satisfied in practice when $f(x)$ is a result of a computer code aimed at computing $\phi(x)$, but has inaccuracies due to internal discretization [12]. It will be evident from our analysis, that the modified line search method makes progress as long as $\|\nabla \phi(x)\|$ is sufficiently large compared to the noise.

Line searches are classical and well-known techniques for improving the performance of optimization algorithms [15]. They allow algorithms to be more robust, less dependent on the choices of hyper-parameters and typically ensure faster practical convergence rates. However, in their original form they rely on exact function and gradient information. Many modern applications give rise to functions for which computing accurate function values and/or gradients is either impossible or prohibitively expensive. Thus, it is desirable to extend the line search paradigm and its analysis to such functions. In [5] a general line search algorithm was analyzed under the conditions that the function values are exact but the gradient estimates are inexact and random. The resulting optimization method is a stochastic process. It is shown that under certain

∗Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA; E-mails: albertberahas@gmail.com, liyuan@lehigh.edu
∗School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA; katyas@cornell.edu
†Corresponding author.
(realizable) probabilistic conditions on the accuracy of the gradient estimates, the resulting line search has the same expected complexity (up to constants) as the line search based on exact gradients. In [16] a line search applied to stochastic functions was analyzed and it was shown that the expected complexity is the same (up to constants) as that of regular deterministic gradient descent, under some probabilistic, and realizable, conditions on stochastic function values and gradient estimates.

In this paper, we extend the analysis in [5] to apply to (1.1). Since the function values are noisy the line search is modified to accept steps that may potentially increase the current estimated value. This modification causes significant changes in the analysis of the expected complexity rates, as the analysis in [5] heavily relies on the fact that objective function can never be increased by the algorithm. Nevertheless, we are able to extend the results in [5] recovering expected complexity for the cases of convex, strongly convex and nonconvex objective functions. The conditions we impose on the line search algorithm are essentially the same as in [5], while the conditions required for the analysis of the stochastic line search [16], where the noise is unbounded, are more restrictive and thus that analysis does not apply to the case we consider here. Moreover, the resulting bounds in [16] have worse dependence on constants than those in [5] and the bounds we derive in this paper.

Our main motivation for this analysis is the recent popularity of smoothing methods for gradient estimates of black-box functions. Stochastic gradient approximations can be computed at relatively low costs, e.g., via Gaussian smoothing and smoothing on a unit sphere, and used within a gradient descent algorithm. This approach has been analyzed in [14] and more recently used in several papers for the specific cases of policy optimization in reinforcement learning and online learning [7, 8, 18]. All of these papers employ specific fixed step length gradient descent schemes within limited settings (e.g., convex functions). Our goal is to develop convergence rate analyses for convex, strongly convex and nonconvex functions, for a generic line search algorithm based on gradient approximations, that can apply not only to gradient descent, but also to quasi-Newton methods such as L-BFGS.

It turns out that the variance of the stochastic gradients computed via Gaussian and unit sphere smoothing can be bounded from above by the squared norm of the expectation, that is \( \|\nabla \phi(x)\|^2 \), when \( \phi(x) \) is the smoothing function [2]. This allows us to use a different probabilistic condition on the accuracy of the gradient estimates in this paper which is simpler than the one used in [5]. However, our analysis also can be easily extended to the condition used in [5].

**Assumptions** Throughout the paper we make the following assumptions.

**Assumption 1.1. (Lipschitz continuity of the gradients of \( \phi \))** The function \( \phi \) is continuously differentiable, and the gradient of \( \phi \) is \( L \)-Lipschitz continuous for all \( x \in \mathbb{R}^n \).

**Assumption 1.2. (Lower bound on \( \phi \))** The function \( \phi \) is bounded below by a scalar \( \hat{\phi} \).

**Assumption 1.3. (Boundedness of Noise in the Function)** There is a constant \( \epsilon_f \geq 0 \) such that \( |f(x) - \phi(x)| = |e(x)| \leq \epsilon_f \) for all \( x \in \mathbb{R}^n \).

Assumption 1.3 may seem very strong, however, we will show that under this assumption the modified line search algorithm converges to a neighborhood of the optimal solution whose size is defined by \( \epsilon_f \). Thus, if it is possible to control the value of \( \epsilon_f \), then one can tighten the convergence neighborhood. This is possible in many applications, where for example, values of \( \phi(x) \) are obtained as a limit to some discretized computation and the error is controlled by the fineness of the discrete grid [12] or if \( \phi(x) \) is a smoothed approximation of \( f(x) \) where the smoothing parameter controls the error between \( f(x) \) and \( \phi(x) \) [11].

**Summary of Results** While we are motivated by some specific methods of computing gradient estimates, in the remainder of the paper, we simply aim to establish complexity bounds on a generic modified line
search algorithm applied to the minimization of convex, strongly convex and nonconvex functions, under the condition that the gradient estimate \(g(x)\) satisfies

\[
\|g(x) - \nabla \phi(x)\| \leq \theta \|\nabla \phi(x)\|, \tag{1.2}
\]

for sufficiently small \(\theta\)\(^1\) with some probability \(1 - \delta\). We establish expected complexity bounds similar to those in [5], where the line search is analyzed under a more complicated bound on \(\|g(x) - \nabla \phi(x)\|\) using exact evaluations of \(\phi(x)\). The bound (1.2), known as the norm condition, was first introduced in [4] and consequently used in a variety of works (see e.g., [3]). This bound is generally not realizable for generic stochastic gradients estimates, however, it can be made to hold for several deterministic and stochastic gradient estimates such as those used in [1, 6, 7, 14]. It is important to note that the analysis in this paper can be easily extended to the more general condition used in [5].

The expected complexity bounds are established in terms of desired accuracy \(\epsilon\), under the assumption that \(\epsilon\) is sufficiently big compared to the error level \(\epsilon_f\). We derive specific bounds on \(\epsilon\) with respect to \(\epsilon_f\) for convex, strongly convex and nonconvex cases.

**Organization** The paper is organized as follows. In Section 2 we describe a general line search algorithm that uses gradient approximations in lieu of the true gradient, and noisy function evaluations of the objective function. We present the stochastic analysis that allows us to bound the expected number of steps required by our generic scheme to reach a desired accuracy in Section 3. This analysis is an extension of the results in [5] that accounts for noise in the objective function. In Section 4, we apply the results of Section 3 to derive global convergence rates and bounds on \(\epsilon\) in terms of \(\epsilon_f\) when the generic line search method is applied to convex, strongly convex and nonconvex functions. Finally, in Section 5 we make some concluding remarks and discuss avenues for future research.

## 2 A Generic Modified Line Search Algorithm

In this section, we describe a generic line search algorithm that uses gradient approximations in lieu of the true gradient and that operates in the noisy regime. In general, line search algorithms construct a possibly noisy approximation of the gradient, \(g(x_k)\), and compute a search direction using this gradient estimate and possibly additional information, e.g., a quasi-Newton search direction. The step size parameter is then chosen; this could be constant, selected from a predetermined sequence of step lengths (e.g., diminishing) or adaptive (e.g., via a back-tracking Armijo line search [15, Chapter 3]). The framework of the generic line search method we analyze here is given in Algorithm 1. As is clear from Algorithm 1, the key components of this method are: (i) the construction of the gradient approximation (Step 1); (ii) the choice of the search direction (Step 2); and (iii) the choice of the step size parameter and the iterate update (Step 3).

**Algorithm 1: Generic Line Search Algorithm**

```
 Algorithm 1: Generic Line Search Algorithm

Inputs: Starting point \(x_0\), initial step size parameter \(\alpha_0\).

for \(k = 0, 1, 2, \ldots\) do

1 Gradient approximation \(g(x_k)\):
   Compute an approximation \(g(x_k)\) of \(\nabla f(x_k)\).

2 Construct a search direction \(d_k\):
   Construct a search direction \(d_k\), e.g., \(d_k = -g(x_k)\) or \(d_k = -H_k g(x_k)\).

3 Compute step size \(\alpha_k\) and update the iterate:
```

Algorithm 1 is a generic line search algorithm. We will perform the analysis in Section 4 for the case \(d_k = -g(x_k)\). We will then outline how this analysis can be easily modified to the case of more general \(d_k\) under additional assumptions on \(d_k\). In order to prove theoretical convergence guarantees, we need to fully

\(^1\)The norms used in this paper are Euclidean norms.
specify the manner in which the step size parameter is selected at every iteration and how a new iterate is computed (Line 3). We consider Algorithm 1 for which the step size parameter \( \alpha_k \) varies under the condition that \( \alpha_k \) is chosen to satisfy a modified version of the sufficient decrease Armijo condition,

\[
f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k d_k^T g(x_k) + 2\epsilon f,
\]

where \( c_1 \in (0, 1) \) is the Armijo parameter, and \( \epsilon f \) is an approximation of the noise in the objective function. If a trial value \( \alpha_k \) does not satisfy (2.1), then the iteration is called unsuccessful; the new iterate is set to the previous iterate, i.e., \( x_{k+1} = x_k \), and the step size parameter is set to a (fixed) fraction \( \tau \leq 1 \) of the previous value, i.e., \( \alpha_{k+1} \leftarrow \tau \alpha_k \). This step makes sense particularly when \( g_k \) (and thus \( d_k \)) are random vectors and thus can be different even for the same \( x_k \). If the trial value satisfies (2.1), then the iteration is called successful, the new iterate is updated based on the search direction \( d_k \), i.e., \( x_{k+1} = x_k + \alpha_k d_k \), and the step size parameter is set to \( \alpha_{k+1} \leftarrow \tau^{-1} \alpha_k \). Algorithm 2, fully specifies a subroutine for computing the step size parameter and taking a step. Note that if \( \tau = 1 \), Algorithm 1 is a constant step size parameter line search algorithm.

**Algorithm 2: Line Search Subroutine**

**Inputs:** Current iterate \( x_k \), current gradient estimate \( g(x_k) \), backtracking factor \( \tau \in (0, 1] \), Armijo parameter \( c_1 \in (0, 1) \).
1. **Check sufficient decrease:**
   - Check if (2.1) is satisfied
2. **if Condition Satisfied (successful step) then**
   - \( x_{k+1} = x_k - \alpha_k d_k \) and \( \alpha_{k+1} \leftarrow \tau^{-1} \alpha_k \)
3. **else**
   - \( x_{k+1} = x_k \) and \( \alpha_{k+1} \leftarrow \tau \alpha_k \)
**Outputs:** New iterate \( x_{k+1} \), new step size parameter \( \alpha_{k+1} \)

The modified Armijo condition has been used in [1]. The addition of the term \( 2\epsilon f \) ensures that a step is possible if \( \alpha_k \) is small enough and \( d_k^T g(x_k) \) is large enough. In [1] the case of functions with arbitrary but bounded noise, such as the one considered here, has been considered. But unlike this paper the error of the gradient estimates is also assumed to be bounded by a constant. Convergence rates are derived for strongly convex objectives.

### 3 Analysis of the Underlying Stochastic Process

In this section, we describe the general mechanism that is used to provide the theoretical results of the paper. This analysis is an extension of the analysis provided in [5] that accounts for possible noise in the function evaluations, i.e., \( \epsilon(x) \neq 0 \).

We begin by introducing several definitions, key assumptions and theoretical results (similar to those in [5]) but suitably modified as required for the analysis in this paper. In particular, as in [5], we view Algorithm 1 as a stochastic process, generated from a sequence of random gradient estimates \( G_k \). \( G_k \) depends on the current iterate \( X_k \), while the step size parameter \( A_k \) and the search direction \( D_k \) depend on \( X_k \) and \( G_k \). Note that \( X_k \) is fully determined by \( G_{k-1} \). Realizations of these random quantities are denoted by \( g_k = G_k(\omega_k) \), \( x_k = X_k(\omega_k) \), \( \alpha_k = A_k(\omega) \) and \( d_k = D_k(\omega_k) \), respectively. For brevity, we will omit the \( \omega_k \) in the notation. We use \( \mathcal{F}^{G}_{k-1} = \sigma(G_0, \ldots, G_{k-1}) \) to denote the the \( \sigma \)-algebra generated by \( G_0, \ldots, G_{k-1} \), that is to say, generated by Algorithm 1 up to the start of iteration \( k \).

**Sufficiently accurate gradients** We assume that the random gradient approximations \( G_k \) satisfy some notion of good quality with probability \( 1 - \delta \). We use the following general notion of sufficiently accurate gradients, similar to that presented in [5].
Definition 3.1. A sequence of random gradients \( \{G_k\} \) is \((1 - \delta)\)-probabilistically “sufficiently accurate” for Algorithm 1, if the indicator variables

\[ I_k = 1\{G_k \text{ is a sufficiently accurate gradient of } \phi \text{ for the given } A_k, X_k \text{ and } D_k\} \]

satisfy the following submartingale condition

\[ \mathbb{P}(I_k = 1|F_{k-1}^c) \geq 1 - \delta, \]  

(3.1)

where \( F_{k-1}^c = \sigma(G_0, \ldots, G_{k-1}) \) is the \( \sigma \)-algebra generated by \( G_0, \ldots, G_{k-1} \). Moreover, we say that iteration \( k \) is a true iteration if the event \( I_k = 1 \) occurs, otherwise the iteration is called false.

Definition 3.1 is generic, but somewhat less so than that in [5], where second order models are also considered, hence Definition 3.1 is not restricted to gradients. The reason Definition 3.1 is generic is because it can be particularized differently depending on the way the gradient estimates are generated. Specifically, in Section 4 we will define sufficiently accurate gradients differently than in [5], however, the results in this paper can be easily modified to apply to that definition as well. We will discuss this in detail in Section 4.

Number of iterations \( N_\epsilon \) to reach \( \epsilon \) accuracy The main goal of this section is to derive bounds on the expected number of iterations \( \mathbb{E}[N_\epsilon] \) required to reach a desired level of accuracy \( \epsilon \). Specifically,

- If \( f \) is convex or strongly convex: \( N_\epsilon \) is the number of iterations required until \( \phi(X_k) - \phi^* \leq \epsilon \) occurs for the first time. Note, \( \phi^* = \phi(x^*) \), where \( x^* \) is a global minimizer of \( f \).
- If \( f \) is nonconvex: \( N_\epsilon \) is the number of iterations required until \( \|\nabla \phi(X_k)\| \leq \epsilon \) occurs for the first time.

Thus \( N_\epsilon \) is a random variable with the property \( \sigma(1\{N_\epsilon > k\}) \subset F_{k-1}^c \), thus it is a stopping time for our stochastic process; see [5, Section 2]. To bound \( \mathbb{E}[N_\epsilon] \) we assume that while \( k < N_\epsilon \), the stochastic process induced by Algorithm 1 behaves in a certain way, in particular it tends to make certain amount of progress towards optimality.

Measure of progress towards optimality and upper bound As is done in [5, Section 2], let \( Z_k \) denote a measure of progress towards optimality, and let \( Z_\epsilon \) be an upper bound for \( Z_k \), under condition \( k < N_\epsilon \). In particular, in our analysis of line search, we will use the definitions of \( Z_k \) and \( Z_\epsilon \) as described in Table 1.

Table 1: Definitions of \( Z_k \) and \( Z_\epsilon \) for convex, strongly convex and nonconvex functions.

| Function      | \( Z_k \)                      | \( Z_\epsilon \) |
|---------------|---------------------------------|------------------|
| convex        | \frac{1}{\epsilon} (\phi(X_k) - \phi^*) | \frac{1}{\epsilon} |
| strongly convex | \log(1/\epsilon) (\phi(X_k) - \phi^*)  | \frac{1}{\epsilon} |
| nonconvex     | \phi(X_0) - \phi(X_k)          | \phi(X_0) - \phi |

Here we introduce the key assumption of the behavior of the stochastic process \( \{A_k, Z_k\} \) generated by Algorithm 1 under which we can derive a bound on \( \mathbb{E}[N_\epsilon] \). In Section 4, we will show that this assumption holds for our generic line search algorithm, under a particular definition of sufficiently accurate gradient estimates, and thus we will be able to derive the expected complexity bound.

Recall that when the gradient estimate \( g_k \) is sufficiently accurate, the iteration is called true, and this is assumed to happen with probability at least \( 1 - \delta \), conditioned on the past. The following assumption is a modification of the assumption in [5, Section 2.4]. Let \( z_k = Z_k(\omega_k) \) be a realization of the random quantity \( Z_k \). Note, \( z_k = Z_k(\omega_k) \) is a related measure of progress towards optimality.

\(^2F_k\) and \( F_\epsilon \) is the notation used in [5].
Assumption 3.2. There exist a constant \( \bar{\alpha} > 0 \), a nondecreasing function \( h(\alpha) : \mathbb{R} \to \mathbb{R} \), which satisfies \( h(\alpha) > 0 \) for any \( \alpha > 0 \), and a nondecreasing function \( r(\epsilon_f) : \mathbb{R} \to \mathbb{R} \), which satisfies \( r(\epsilon_f) \geq 0 \) for any \( \epsilon_f \geq 0 \), such that for any realization of Algorithm 1 the following hold for all \( k < N_e \):

(i) If iteration \( k \) is true (i.e. \( I_k = 1 \)) and successful, then \( z_{k+1} \geq z_k + h(\alpha_k) - r(\epsilon_f) \).

(ii) If \( \alpha_k \leq \bar{\alpha} \) and iteration \( k \) is true then iteration \( k \) is also successful, which implies \( \alpha_{k+1} = \tau^{-1}\alpha_k \).

(iii) \( z_{k+1} \geq z_k - r(\epsilon_f) \) for all successful iterations \( k \) and \( z_{k+1} \geq z_k \) for all unsuccessful iteration \( k \).

(iv) The ratio \( r(\epsilon_f)/h(\bar{\alpha}) \) is bounded from above by some \( \gamma \in (0, 1) \).

The key difference between Assumption 3.2 and the corresponding assumption in [5, Assumption 2.1] is that on each successful iteration \( Z_k \) may be decreased by up to \( r(\epsilon_f) \). When \( r(\epsilon_f) = 0 \), Assumption 3.2 reduces to the assumption in [5].

As in [5] we define additional indicator random variables:

\[
\Lambda_k = \mathbb{I}\{A_k > \bar{\alpha}\}, \quad \bar{\Lambda}_k = \mathbb{I}\{A_k \geq \bar{\alpha}\},
\]

\[
\Theta_k = \mathbb{I}\{\text{Iteration } k \text{ is successful i.e., } A_{k+1} = \tau^{-1}A_k\},
\]

Note that \( \sigma(\Lambda_k) \subset \mathcal{F}_{k-1}^G \), \( \sigma(\bar{\Lambda}_k) \subset \mathcal{F}_{k-1}^G \) and \( \sigma(\Theta_k) \subset \mathcal{F}_k^G \), that is the random variables \( \Lambda_k \) and \( \bar{\Lambda}_k \) are fully determined by the first \( k-1 \) steps of the algorithm, while \( \Theta_k \) is fully determined by the first \( k \) steps.

Without loss of generality, we assume that \( \bar{\alpha} = \tau^c\alpha_0 \) for some positive integer \( c \). In other words, \( \bar{\alpha} \) is the largest value that the step size \( A_k \) actually achieves for which part (ii) of Assumption 3.2 holds. Note that if \( \tau = 1 \), the algorithm uses constant step size and hence has to start with the value for which Assumption 3.2 holds in order to converge.

Under Assumption 3.2, recalling the update rules for \( \alpha_k \) in Algorithm 1 and the assumption that true iterations occur with probability at least \( 1 - \delta \), we can write the stochastic process \( \{A_k, Z_k\} \) as obeying the expressions below:

\[
A_{k+1} = \begin{cases} 
\tau^{-1}A_k & \text{if } I_k = 1 \text{ and } \Lambda_k = 0, \\
\tau^{-1}A_k & \text{if } \Theta_k = 1, \ I_k = 0 \text{ and } \Lambda_k = 0, \\
\tau A_k & \text{if } \Theta_k = 0, \ I_k = 0 \text{ and } \Lambda_k = 0, \\
\tau^{-1}A_k & \text{if } \Theta_k = 1 \text{ and } \Lambda_k = 1, \\
\tau A_k & \text{if } \Theta_k = 0 \text{ and } \Lambda_k = 1,
\end{cases}
\]

\[
Z_{k+1} = \begin{cases} 
Z_k + h(A_k) - r(\epsilon_f) & \text{if } \Theta_k = 1 \text{ and } I_k = 1, \\
Z_k - r(\epsilon_f) & \text{if } \Theta_k = 1, \ I_k = 0 \\
Z_k & \text{if } \Theta_k = 0.
\end{cases}
\]

3.1 Analysis of the stochastic processes

We will now present the derivation of the bounds on \( \mathbb{E}(N_e) \) under Assumption 3.2, by modifying the analysis in [5]. We start by introducing a useful lemma from [5].

Lemma 3.3. Let \( N_e \) denote the stopping time. For all \( k < N_e \), let \( I_k \) be the sequence of random variables in Definition 3.1 so that (3.1) holds. Let \( W_k \) be a nonnegative stochastic process such that \( \sigma(W_k) \subset \mathcal{F}_k^G \), for any \( k \geq 0 \). Then,

\[
\mathbb{E}\left(\sum_{k=0}^{N_e-1} W_k I_k\right) \geq (1 - \delta)\mathbb{E}\left(\sum_{k=0}^{N_e-1} W_k\right).
\]

Similarly,

\[
\mathbb{E}\left(\sum_{k=0}^{N_e-1} W_k(1 - I_k)\right) \leq \delta\mathbb{E}\left(\sum_{k=0}^{N_e-1} W_k\right).
\]


The following lemma from [5] bounds the number of steps for which \( \alpha_k \leq \bar{\alpha} \). The proof depends only on the probabilities of different outcomes and not on the changes in \( Z_k \), thus the proof from [5] applies directly.

**Lemma 3.4.** The expected number of steps for which \( \alpha_k \leq \bar{\alpha} \) can be bounded as,

\[
E\left( \sum_{k=0}^{N-1} (1 - \Lambda_k) \right) \leq \frac{1}{2(1 - \delta)} E(N_c).
\]

We now turn to the derivation of the bound on \( E\left( \sum_{i=0}^{N-1} \Lambda_i \right) \), which requires a substantially more elaborate analysis than that in [5] but is similar in spirit. The key difference is that, while in [5] \( Z_k \) never decreases, here we have to account for all iterations where \( Z_k \) may decrease, and bound their expected number. For brevity of notation, we define the following quantities:

- \( N_{FS} = \sum_{k=0}^{N-1} \bar{\Lambda}_k (1 - I_k) \Theta_k \) - the number of false successful iterations with \( \mathcal{A}_k \geq \bar{\alpha} \).
- \( N_{TS} = \sum_{k=0}^{N-1} \bar{\Lambda}_k I_k \Theta_k \) - the number of true successful iterations with \( \mathcal{A}_k \geq \bar{\alpha} \).
- \( N_F = \sum_{k=0}^{N-1} \bar{\Lambda}_k (1 - I_k) \) - the number of false iterations with \( \mathcal{A}_k \geq \bar{\alpha} \).
- \( N_T = \sum_{k=0}^{N-1} \bar{\Lambda}_k I_k \) - the number of true iterations with \( \mathcal{A}_k \geq \bar{\alpha} \).
- \( N_{TU} = \sum_{k=0}^{N-1} \Lambda_k I_k (1 - \Theta_k) \) - the number of true unsuccessful iterations with \( \mathcal{A}_k > \bar{\alpha} \).
- \( N_U = \sum_{k=0}^{N-1} \Lambda_k (1 - \Theta_k) \) - the number of unsuccessful iterations with \( \mathcal{A}_k > \bar{\alpha} \).
- \( M_S = \sum_{k=0}^{N-1} (1 - \bar{\Lambda}_k) \Theta_k \) - the number of successful iterations with \( \mathcal{A}_k < \bar{\alpha} \).

Since \( E\left( \sum_{k=0}^{N-1} \Lambda_k \right) = E\left( \sum_{k=0}^{N-1} \bar{\Lambda}_k (1 - I_k) \right) + E\left( \sum_{k=0}^{N-1} \Lambda_k I_k \right) \leq E(N_F) + E(N_T) \), our goal is to bound \( E(N_F) + E(N_T) \).

We now establish several inequalities relating the quantities we just defined. We begin with,

\[
N_T = N_{FS} + N_{TU} \leq N_{TS} + N_U. \tag{3.4}
\]

The equality above holds because by Assumption 3.2(ii) there are no true unsuccessful iterations when \( \mathcal{A}_k = \bar{\alpha} \).

**Lemma 3.5.** For any \( l \in \{0, \ldots, N_c - 1\} \) and for all realizations of Algorithm 1, we have

\[
\sum_{k=0}^{l} \Lambda_k (1 - \Theta_k) \leq \sum_{k=0}^{l} \bar{\Lambda}_k \Theta_k + \log_\tau\left( \frac{\bar{\alpha}}{\alpha_0} \right),
\]

hence when \( l = N_c - 1 \),

\[
N_T \leq N_{FS} + 2N_{TS} + \log_\tau(\bar{\alpha}/\alpha_0). \tag{3.5}
\]

**Proof.** \( \mathcal{A}_k \) is increased on successful iterations and decreased on unsuccessful ones. Hence the total number of steps when \( \mathcal{A}_k > \bar{\alpha} \) and \( \mathcal{A}_k \) is decreased, is bounded by the total number of steps when \( \mathcal{A}_k \geq \bar{\alpha} \) is increased plus the number of steps it is required to reduce \( \mathcal{A}_k \) from its initial value \( \alpha_0 \) to \( \bar{\alpha} \). Inequality (3.5) is a simple consequence of this observation combined with (3.4). \qed

**Lemma 3.6.** The expected number of false iterations with \( \mathcal{A}_k \geq \bar{\alpha} \) can be bounded as,

\[
E(N_F) \leq \frac{\delta}{1 - \delta} E(N_T). \tag{3.6}
\]

**Proof.** The proof uses Lemma 3.3 and is the same as in [5]. \qed
Hence by (3.4) and Lemmas 3.5 and 3.6, we have
\[ \mathbb{E}(N_F) + \mathbb{E}(N_T) \leq \frac{1}{1 - \delta} \mathbb{E}(N_T) \leq \frac{1}{1 - \delta} (\mathbb{E}(N_{TS}) + \mathbb{E}(N_U)) \leq \frac{1}{1 - \delta} \left( \mathbb{E}(N_{FS}) + 2\mathbb{E}(N_{TS}) + \log_F \left( \frac{\bar{\alpha}}{\alpha_0} \right) \right). \] (3.7)

We now bound \( \mathbb{E}(M_S) \).

**Lemma 3.7.** The expected number of successful iterations with \( A_k < \bar{\alpha} \) can be bounded as,
\[ \mathbb{E}(M_S) = \mathbb{E} \left( \sum_{k=0}^{N_s-1} (1 - \bar{\Lambda}_k)\Theta_k \right) \leq \frac{\delta}{2(1 - \delta)} \mathbb{E}(N_s). \]

**Proof.** We want to bound the expected number of successful iterations for which \( \alpha_k < \bar{\alpha} \). Since on all successful iterations \( \alpha_k \) is increased, and \( \alpha_0 \geq \lambda \), then for each such successful iteration there has to be an unsuccessful iteration with \( \alpha_k \leq \lambda \). Hence
\[ \sum_{k=0}^{N_s-1} (1 - \bar{\Lambda}_k)\Theta_k \leq \sum_{k=0}^{N_s-1} (1 - \Lambda_k)(1 - \Theta_k) \leq \sum_{k=0}^{N_s-1} (1 - \Lambda_k)(1 - I_k). \]

The last inequality follows from the fact that when \( \alpha_k \leq \lambda \), all true iterations are successful, which implies \((1 - \Lambda_k)I_k \leq (1 - \Lambda_k)\Theta_k \). Now applying Lemmas 3.3 and 3.4 we have
\[ \mathbb{E} \left( \sum_{k=0}^{N_s-1} (1 - \Lambda_k)(1 - I_k) \right) \leq \delta \mathbb{E} \left( \sum_{k=0}^{N_s-1} (1 - \Lambda_k) \right) \leq \frac{\delta}{2(1 - \delta)} \mathbb{E}(N_s), \]
and the result of the lemma follows. \( \square \)

Our next observation is central in our analysis. It reflects the fact that the total gain minus the total loss in \( Z_k \) is bounded from above by \( Z_s \). We observe that when \( A_k \geq \bar{\alpha} \) on true successful iterations this gain is bounded from below away from zero by \( h(\bar{\alpha}) - r(\epsilon_f) \geq (1 - \gamma)h(\bar{\alpha}) \) and at other successful iterations the loss is bounded above by \( r(\epsilon_f) \). This will allow us to bound \( \mathbb{E}(N_{TS}) \).

**Lemma 3.8.** The number of true successful iterations with \( A_k \geq \bar{\alpha} \) can be bounded as,
\[ N_{TS} \leq \frac{Z_s}{(1 - \gamma)h(\bar{\alpha})} + \frac{\gamma}{1 - \gamma} (N_{FS} + M_S) \] (3.8)
and, hence,
\[ \mathbb{E}(N_{TS}) \leq \frac{Z_s}{(1 - \gamma)h(\bar{\alpha})} + \frac{\gamma}{1 - \gamma} \mathbb{E}(N_{FS}) + \frac{\gamma}{1 - \gamma} \delta \frac{\delta}{2(1 - \delta)} \mathbb{E}(N_s). \] (3.9)

**Proof.** The proof follows directly from (3.3) and Assumption 3.2. \( F_k \) is increased by at least \( h(\bar{\alpha}) - r(\epsilon_f) \) at each true successful iteration when \( \alpha_k \geq \bar{\alpha} \) and it may be decreased at most \( r(\epsilon_f) \) at each false successful iteration when \( \alpha_k \geq \bar{\alpha} \) and at each successful iteration when \( \alpha_k < \bar{\alpha} \). Thus, we have
\[ Z_s \geq Z_k \geq N_{TS}(h(\bar{\alpha}) - r(\epsilon_f)) - r(\epsilon_f)(N_{FS} + M_S). \] (3.10)
Recalling that by Assumption 3.2, \( r(\epsilon_f) \leq \gamma h(\bar{\alpha}) \) and \( \gamma \in (0, 1) \) we obtain (3.8), while (3.9) follows further from Lemma 3.7. \( \square \)

**Lemma 3.9.** Under the condition that \( \delta < \frac{1}{2} - \frac{\gamma}{2} \), the number of false successful iterations with \( A_k \geq \bar{\alpha} \) can be bounded as,
\[ \mathbb{E}(N_{FS}) \leq \frac{2\delta}{1 - 2\delta - \gamma} F_s \frac{h(\bar{\alpha})}{(1 - \gamma)h(\bar{\alpha})} + \frac{(1 - \gamma)\delta}{1 - 2\delta - \gamma} \log_r \left( \frac{\bar{\alpha}}{\alpha_0} \right) + \frac{\delta^2 \gamma}{(1 - \delta)(1 - 2\delta - \gamma)} \mathbb{E}(N_s) \] (3.11)
\[\mathbb{E}(N_{FS}) \leq \frac{\delta}{1 - \delta} \left[ \mathbb{E}(N_F) + 2\mathbb{E}(N_{TS}) + \log_\tau \left( \frac{\bar{\alpha}}{\alpha_0} \right) \right].\]

Then, from Lemma 3.8
\[\mathbb{E}(N_{FS}) \leq \frac{\delta}{1 - \delta} \left[ \frac{1 + \gamma}{1 - \gamma} \mathbb{E}(N_{FS}) + \frac{2F_e}{(1 - \gamma)h(\bar{\alpha})} + \frac{\gamma\delta}{1 - \gamma} \mathbb{E}(N_e) + \log_\tau \left( \frac{\bar{\alpha}}{\alpha_0} \right) \right].\]

Collecting the terms involving \(\mathbb{E}(N_{FS})\) on the left and observing that \(1 - \frac{1 + \gamma}{1 - \gamma} \frac{\delta}{1 - \delta} = \frac{1 - 2\delta - \gamma}{(1 - \gamma)(1 - \delta)}\) we can derive the bound
\[\mathbb{E}(N_{FS}) \leq \frac{(1 - \gamma)\delta}{1 - 2\delta - \gamma} \left[ \frac{2F_e}{(1 - \gamma)h(\bar{\alpha})} + \frac{\gamma\delta}{1 - \gamma} \mathbb{E}(N_e) + \log_\tau \left( \frac{\bar{\alpha}}{\alpha_0} \right) \right].\]

We can now derive the bound for \(\mathbb{E}(N_{TS})\) using Lemmas 3.8 and 3.9 and collecting the appropriate terms.

**Lemma 3.10.** Under the condition that \(\delta < \frac{1}{2} - \frac{\gamma}{2}\), the number of true successful iterations with \(\mathcal{A}_k \geq \bar{\alpha}\) can be bounded as,
\[\mathbb{E}(N_{TS}) \leq \frac{1 - 2\delta}{1 - 2\delta - \gamma} \frac{F_e}{(1 - \gamma)h(\bar{\alpha})} + \frac{\gamma\delta}{1 - \gamma} \mathbb{E}(N_{FS}) + \frac{\gamma(1 - 2\delta)\delta}{2(1 - \delta)(1 - 2\delta - \gamma)} \mathbb{E}(N_e) \tag{3.12}\]

**Proof.** From Lemma 3.8
\[\mathbb{E}(N_{TS}) \leq \frac{F_e}{(1 - \gamma)h(\bar{\alpha})} + \frac{\gamma}{1 - \gamma} \mathbb{E}(N_{FS}) + \frac{\gamma\delta}{1 - \gamma} \mathbb{E}(N_e).\]

Using the result from Lemma 3.9
\[\mathbb{E}(N_{TS}) \leq \frac{F_e}{(1 - \gamma)h(\bar{\alpha})} + \frac{\gamma}{1 - \gamma} \left[ \frac{2\delta}{1 - 2\delta - \gamma} \frac{F_e}{h(\bar{\alpha})} + \frac{(1 - \gamma)\delta}{1 - 2\delta - \gamma} \log_\tau \left( \frac{\bar{\alpha}}{\alpha_0} \right) + \frac{\delta^2\gamma}{(1 - \delta)(1 - 2\delta - \gamma)} \mathbb{E}(N_e) \right] + \frac{\gamma(1 - 2\delta)\delta}{2(1 - \delta)(1 - 2\delta - \gamma)} \mathbb{E}(N_e)\]

**Lemma 3.11.** Under the condition that \(\delta < \frac{1}{2} - \frac{\gamma}{2}\), the number of iterations with \(\mathcal{A}_k > \bar{\alpha}\) can be bounded as,
\[\mathbb{E} \left( \sum_{k=0}^{N-1} \Lambda_k \right) \leq \frac{2}{1 - 2\delta - \gamma} \frac{F_e}{h(\bar{\alpha})} + \frac{(1 - \gamma)}{1 - 2\delta - \gamma} \log_\tau \left( \frac{\bar{\alpha}}{\alpha_0} \right) + \frac{\gamma\delta}{(1 - \delta)(1 - 2\delta - \gamma)} \mathbb{E}(N_e).\]
Proof. Recall that $\mathbb{E}\left(\sum_{k=0}^{N_\epsilon-1} \Lambda_k\right) \leq \mathbb{E}(N_F + N_T)$. Using (3.7) and Lemmas 3.9 and 3.10 we have

\[
\mathbb{E}\left(\sum_{k=0}^{N_\epsilon-1} \Lambda_k\right) \leq \frac{1}{1-\delta} \left[\mathbb{E}(N_F S) + 2\mathbb{E}(N_T S) + \log_\tau(\bar{\alpha}/\alpha_0)\right] \\
\leq \frac{1}{1-\delta} \left[\frac{2\delta}{1-2\delta - \gamma} \frac{F_\epsilon}{h(\bar{\alpha})} + \frac{(1-\gamma)\delta}{1-2\delta - \gamma} \log_\tau\left(\frac{\bar{\alpha}}{\alpha_0}\right) + \frac{\delta^2 \gamma}{(1-\delta)(1-2\delta - \gamma)} \mathbb{E}(N_\epsilon)\right] \\
+ \frac{2}{1-\delta} \left[\frac{1-2\delta - \gamma}{1-2\delta - \gamma} \frac{F_\epsilon}{h(\bar{\alpha})} + \frac{\gamma\delta}{1-2\delta - \gamma} \log_\tau\left(\frac{\bar{\alpha}}{\alpha_0}\right) + \frac{(2p-1)\delta}{2(1-\delta)(1-2\delta - \gamma)} \mathbb{E}(N_\epsilon)\right] \\
+ \frac{1}{1-\delta} \log_\tau\left(\frac{\bar{\alpha}}{\alpha_0}\right)
\]

Combining Lemmas 3.4 and 3.11 we have the key bound

\[
\mathbb{E}(N_\epsilon) \leq \mathbb{E}\left(\sum_{k=0}^{N_\epsilon-1} \Lambda_k\right) + \mathbb{E}\left(\sum_{k=0}^{N_\epsilon-1} (1 - \Lambda_k)\right) \\
\leq \frac{2}{1-2\delta - \gamma} \frac{F_\epsilon}{h(\bar{\alpha})} + \frac{(1-\gamma)\delta}{1-2\delta - \gamma} \log_\tau\left(\frac{\bar{\alpha}}{\alpha_0}\right) + \frac{\gamma\delta}{(1-\delta)(1-2\delta - \gamma)} \mathbb{E}(N_\epsilon) + \frac{1}{2(1-\delta)} \mathbb{E}(N_\epsilon)
\]

Collecting the terms with $\mathbb{E}(N_\epsilon)$ on the left-hand-side and multiplying both sides by $1-2\delta - \gamma$ we obtain

\[
\left[1 - 2\delta - \gamma - \frac{\gamma\delta}{1-\delta} - \frac{1-2\delta - \gamma}{2(1-\delta)}\right] \mathbb{E}(N_\epsilon) \leq \frac{2F_\epsilon}{\bar{h}(\bar{\alpha})} + (1-\gamma) \log_\tau\left(\frac{\bar{\alpha}}{\alpha_0}\right)
\]

If the coefficient in front of $\mathbb{E}(N_\epsilon)$ is positive, that immediately gives us a bound on the expected stopping time $\mathbb{E}(N_\epsilon)$. This coefficient is

\[
1 - 2\delta - \gamma - \frac{\gamma\delta}{1-\delta} - \frac{1-2\delta - \gamma}{2(1-\delta)} = \frac{4\delta^2 - 4\delta + 1 - \gamma}{2(1-\delta)} = \frac{(1-2\delta)^2 - \gamma}{2(1-\delta)}.
\]

The smaller of the two roots of $4\delta^2 - 4\delta + 1 - \gamma$ is $\frac{1}{2} - \frac{\sqrt{\gamma}}{2} \leq \frac{1}{2} - \frac{\gamma}{2}$. Hence, we have the following final bound.

**Theorem 3.12.** Under the condition that $\delta < \frac{1}{2} - \frac{\sqrt{\gamma}}{2}$, the stopping time $N_\epsilon$ is bounded in expectation as follows

\[
\mathbb{E}(N_\epsilon) \leq \frac{2(1-\delta)}{(1-2\delta)^2 - \gamma} \left[\frac{2F_\epsilon}{\bar{h}(\bar{\alpha})} + (1-\gamma) \log_\tau\left(\frac{\bar{\alpha}}{\alpha_0}\right)\right]
\]

(3.14)

**Remark 3.13.** The result of Theorem 3.12 is a generalization of the result in [5] to the case where the function is computed with some noise. Specifically, when $\epsilon_\ell = 0$, and as a result $\tau(\epsilon_\ell) = 0$, then $\gamma = 0$ and (3.14) reduces to the bounds in [5]. If, on the other hand $\delta = 0$, then we recover deterministic complexity bound.

### 4 Convergence Analysis of the Modified Line Search

We begin by stating the specific condition on the gradient estimates which we use in our analysis.

\[
\|g(x_k) - \nabla \phi(x_k)\| \leq \theta \|\nabla \phi(x_k)\|, \quad \text{for all} \quad k = 0, 1, 2, \ldots
\]

(4.1)

for some $\theta \in [0, 1)$. This condition is referred to as a *norm condition* and was introduced and studied in [4] in the context of trust-region methods with inaccurate gradients. Note, this condition implies that $g(x_k)$ is
a descent direction for the function $\phi$. When unbiased stochastic estimators of $\nabla \phi(x)$ are available, $g_k$ can be computed by averaging these estimators. If the variance of these estimators is bounded by $\|\nabla \phi(x_k)\|^2$, then condition (4.1) can be satisfied, with probability $1 - \delta$, by using a sufficiently large number of the estimators (batch size) to compute $g_k$. This happens to be the case when $\phi(x)$ is a Gaussian smoothing [14].

In a more general stochastic setting, unless one knows $\|\nabla \phi(x_k)\|$, this condition is hard or impossible to verify or guarantee. There is significant amount of work that attempts to circumvent this difficulty in case of general stochastic gradient estimates; see e.g., [3, 5, 16]. In [3] a practical approach to estimate $\|\nabla \phi(x_k)\|$ is proposed and used to ensure some approximation of (4.1) holds. In [5] and [16], (4.1) is replaced with a condition that for some $\kappa > 0$ and for each iteration $k = 0, 1, 2, \ldots$

$$\|g(x_k) - \nabla \phi(x_k)\| \leq \kappa \alpha_k \|g(x_k)\|, \quad (4.2)$$

holds with probability $1 - \delta$. Under this condition, expected complexity bounds are derived for a line search method that has access to deterministic function values in $[Sufficiently accurate gradients]$. Assumption 4.2.]

A simple way of making condition (4.1) realizable is to replace $\|\nabla \phi(x_k)\|$ with $\epsilon$, where $\epsilon$ is the desired convergence accuracy. However, if the cost of obtaining $g(x_k)$ that satisfies $\|g(x_k) - \nabla \phi(x_k)\| \leq \epsilon$ increases as $\epsilon$ decreases, replacing $\|\nabla \phi(x_k)\|$ by its global lower bound $\epsilon$ can lead to inefficient algorithms.

While we can apply analysis to the condition (4.2), in this paper we choose to use condition (4.1) because we are motivated by the specific setting where estimates $g_k$ are computed via finite differences, interpolation or smoothing [2].

In the remainder of this section, we present a convergence analysis for the generic line search algorithm (Algorithms 1-2). The analysis is an extension of the analysis presented in [5] under condition (4.1). We use the following notion of sufficiently accurate gradients.

**Definition 4.1.** A sequence of random gradients $\{G_k\}$ is $(1 - \delta)$-probabilistically “sufficiently accurate” for Algorithm 1 if there exists a constant $\theta \in [0, \frac{1 - \kappa \alpha_k}{\kappa \alpha_k})$, such that the indicator variables

$$I_k = \mathbb{1}\{\|G_k - \nabla \phi(X_k)\| \leq \theta \|\nabla \phi(X_k)\|\}$$

satisfy the following submartingale condition

$$\mathbb{P}(I_k = 1 | \mathcal{F}_{k-1}^G) \geq 1 - \delta,$$

where $\mathcal{F}_{k-1}^G = \sigma(G_0, \ldots, G_{k-1})$ is the $\sigma$-algebra generated by $G_0, \ldots, G_{k-1}$. Moreover, we say that iteration $k$ is a true iteration is the event $I_k = 1$ occurs, otherwise the iteration is called false.

For the remainder of this section, we make the following additional assumption.

**Assumption 4.2. (Sufficiently accurate gradients)** The sequence of random gradients $\{G_k\}$ generated by Algorithm 1 are $(1 - \delta)$-probabilistically “sufficiently accurate” with $\delta < \frac{1}{2} - \frac{\gamma \sqrt{2}}{2}$, for some $\gamma \in (0, 1)$.

Equipped with the above definitions, assumptions and theorems, we now provide convergence guarantees for the generic line search algorithm (Algorithm 1), where the step size parameter is chosen using Algorithm 2, for convex, strongly convex and nonconvex objective functions.

For each true iteration (i.e., $I_k = 1$), we have

$$\|g(x_k) - \nabla \phi(x_k)\| \leq \theta \|\nabla \phi(x_k)\|, \quad (4.3)$$

which implies, using the triangle inequality that

$$\|g(x_k)\| \geq (1 - \theta) \|\nabla \phi(x_k)\|. \quad (4.4)$$

We now show that Assumption 3.2 is satisfied. To this end, for the three classes of functions, we show that there exists an upper bound $\bar{\alpha}$ on the step length parameter, and functions $h(\alpha)$ and $r(\epsilon_f)$ such that the assumption is true. First, we derive an expression for the constant $\bar{\alpha}$.
Lemma 4.3. Let Assumption 1.1 hold. For every realization of Algorithm 1, if iteration $k$ is true (i.e., $I_k = 1$), and if
\[
\alpha_k \leq \bar{\alpha} = \frac{2(1 - 2\theta - c_1(1 - \theta))}{L(1 - \theta)},
\] (4.5)
then (2.1) holds. In other words, when (4.5) holds, any true iteration is also a successful iteration. Moreover, for every true and successful iteration,
\[
\phi(x_{k+1}) \leq \phi(x_k) - c_1\alpha_k(1 - \theta)^2\|\nabla \phi(x_k)\|^2 + 4\epsilon_f.
\] (4.6)

Proof. By Assumption 1.1, we have
\[
\phi(x_k - \alpha_k g(x_k)) \leq \phi(x_k) - \alpha_k g(x_k)^T \nabla \phi(x_k) + \frac{\alpha_k^2 L}{2} \|g(x_k)\|^2,
\] (4.7)
Applying the Cauchy-Schwarz inequality, (4.3) and (4.4), for every true iteration
\[
\begin{align*}
\phi(x_k - \alpha_k g(x_k)) &\leq \phi(x_k) - \alpha_k g(x_k)^T \nabla \phi(x_k) + \frac{\alpha_k^2 L}{2} \|g(x_k)\|^2 \\
&= \phi(x_k) - \alpha_k g(x_k)^T \nabla \phi(x_k) - g(x_k) - \alpha_k \left[1 - \frac{\alpha_k L}{2}\right] \|g(x_k)\|^2 \\
&\leq \phi(x_k) + \alpha_k \|g(x_k)\| \|\nabla \phi(x_k) - g(x_k)\| - \alpha_k \left[1 - \frac{\alpha_k L}{2}\right] \|g(x_k)\|^2 \\
&\leq \phi(x_k) + \alpha_k \|g(x_k)\| \left[\frac{1 - 2\theta}{1 - \theta} - \frac{\alpha_k L}{2}\right] - \alpha_k \left[1 - \frac{\alpha_k L}{2}\right] \|g(x_k)\|^2 \\
&= \phi(x_k) - \alpha_k \left[\frac{1 - 2\theta}{1 - \theta} - \frac{\alpha_k L}{2}\right] \|g(x_k)\|^2.
\end{align*}
\]
By Assumption 1.3, we have
\[
f(x_k - \alpha_k g(x_k)) \leq f(x_k) - \alpha_k \left[1 - \frac{2\theta}{1 - \theta} - \frac{\alpha_k L}{2}\right] \|g(x_k)\|^2 + 2\epsilon_f.
\]
From this we conclude that (2.1) holds whenever
\[
f(x_k) - \alpha_k \left[1 - \frac{2\theta}{1 - \theta} - \frac{\alpha_k L}{2}\right] \|g(x_k)\|^2 + 2\epsilon_f \leq f(x_k) - c_1\alpha_k \|g(x_k)\|^2 + 2\epsilon_f,
\]
which is equivalent to (4.5). Therefore, using Assumption 1.3 and (4.4), for every true and successful iteration we have
\[
\phi(x_{k+1}) \leq \phi(x_k) - c_1\alpha_k(1 - \theta)^2\|\nabla \phi(x_k)\|^2 + 4\epsilon_f,
\]
which completes the proof.

We should mention that when the error in the gradient approximation is zero, i.e., $\theta = 0$, we recover the step size parameter condition for the deterministic setting. Moreover, when there is no noise in the function, i.e., $\epsilon_f = 0$, we recover the sufficient decrease condition of the deterministic gradient descent algorithm with an Armijo backtracking line search.

Next, we state and prove a result for the case of false and successful iterations.

Lemma 4.4. For every false and successful iteration of Algorithm 1, we have
\[
\phi(x_{k+1}) \leq \phi(x_k) - c_1\alpha_k \|g(x_k)\|^2 + 4\epsilon_f.
\] (4.8)

12
Proof. The proof of this lemma is straightforward. For every successful iteration we have

\[ f(x_{k+1}) \leq f(x_k) - c_1 \alpha_k \|g(x_k)\|^2 + 2\epsilon_f. \]

Thus, by Assumption 1.3,

\[ \phi(x_{k+1}) \leq \phi(x_k) - c_1 \alpha_k \|g(x_k)\|^2 + 4\epsilon_f, \]

which completes the proof.

The result of Lemma 4.4 shows the amount of decrease on false and successful iterations. Note, the error term 4\epsilon_f illustrates that on false iterations the function value may increase and that the increase is related to the noise in the function values.

### 4.1 Convex Functions

In this section, we analyze the expected complexity of Algorithm 1 in the case when \( \phi \) is a convex function.

**Assumption 4.5.** (Convexity and boundedness of iterates) The function \( \phi \) is convex and there exists a constant \( D > 0 \) such that

\[ \|x - x^*\| \leq D \quad \text{for all} \quad x \in \mathcal{U}, \quad (4.9) \]

where \( x^* \) is some global minimizer of \( f \) and the set \( \mathcal{U} \) contains all iteration realizations. Let \( \phi^* = \phi(x^*) \).

This assumption may seem strong since it requires all iterates of the algorithm to remain in a bounded region. When the objective function is not allowed to increase, this assumption is simply ensured by assuming bounded level sets of \( \phi(x) \). In the case of noisy function values in principle, iterates can wander out of a bounded region with some small probability (as this will require a large sequence of false successful iterations). Thus, ideally, we need to modify the algorithm to prevent it from going outside of some predefined bounded region, that is known to contain \( x^* \). Such modification is simple and our analysis will still apply, but with some notational complications. Therefore, we choose not to impose this modification explicitly. Note, we only use this assumption in the convex case and drop it in the strongly convex and nonconvex cases, and, thus, the nonconvex case convergence rates apply to the convex case without (4.9).

We bound the number of iterations taken by Algorithm 1 until \( \phi(X_k) - \phi^* \leq \epsilon \) occurs. Let

\[ \Delta_k^\phi = \phi(X_k) - \phi^*, \quad \text{and} \quad Z_k = \frac{1}{\Delta_k^\phi}. \quad (4.10) \]

By this definition, \( N_\epsilon \) is the number of iterations taken until \( Z_k \geq \frac{1}{\epsilon} = Z_\epsilon \). Note, that due to the noise in the function evaluations, \( \epsilon \) cannot be chosen to be arbitrarily small. We make an assumption on \( \epsilon \) that explicitly defines the neighborhood of convergence.

**Assumption 4.6.** (Neighborhood of convergence, convex case)

\[ \epsilon^2 > \max \left\{ \frac{8\epsilon_f LD^2}{\gamma c_1 (1 - \theta)(1 - 2\theta - c_1(1 - \theta))}, \frac{16\epsilon_f^2}{4D^2} \right\}, \]

with some \( \gamma \in (0, 1) \).

By Lemma 4.3, whenever \( A_k \leq \bar{\alpha} \), then every true iteration is also successful. We now show that on true and successful iterations, \( Z_k \) increases by at least some function \( h(A_k) - r(\epsilon_f) \), for all \( k < N_\epsilon \).

**Lemma 4.7.** Let Assumptions 1.3 and 4.5 hold, and consider any realization of Algorithm 1. For every iteration that is true and successful, we have

\[ \frac{1}{\Delta_{k+1}^\phi} \geq \frac{1}{\Delta_k^\phi} + \frac{c_1 \alpha_k (1 - \theta)^2}{4D^2} - \frac{4\epsilon_f}{\epsilon^2} \quad (4.11) \]

where \( \Delta_k^\phi \) is defined in (4.10).
Proof. By Assumption 4.5, for all \( x, y \in \mathbb{R}^n \), we have
\[
\phi(x) - \phi(y) \geq \nabla \phi(y)^T (x - y).
\]
Thus, if \( x = x^* \) and \( y = x_k \), we have
\[
-\Delta_k^\phi = \phi^* - \phi(x_k) \geq \nabla \phi(x_k)^T (x^* - x_k) \geq -D \| \nabla \phi(x_k) \|,
\]
where we used the Cauchy-Schwarz inequality and (4.9). Thus, when \( k \) is a true iteration, by (4.4) we have
\[
\|g(x_k)\|^2 \geq (1 - \theta)^2 \| \nabla \phi(x_k) \|^2 \geq \frac{(1 - \theta)^2 (\Delta_k^\phi)^2}{D^2}.
\]
If \( k \) is also a successful iteration, then
\[
\Delta_k^\phi - \Delta_{k+1}^\phi = \phi(x_k) - \phi(x_{k+1}) \geq c_1 \alpha_k \|g(x_k)\|^2 - 4\epsilon_f \geq \frac{c_1 \alpha_k (1 - \theta)^2 (\Delta_k^\phi)^2}{D^2} - 4\epsilon_f,
\]
and thus,
\[
\Delta_k^\phi + 4\epsilon_f - \Delta_{k+1}^\phi \geq \frac{c_1 \alpha_k (1 - \theta)^2 (\Delta_k^\phi)^2}{D^2}.
\]
Dividing by \( (\Delta_{k+1}^\phi) (\Delta_k^\phi + 4\epsilon_f) \)
\[
\frac{1}{\Delta_{k+1}^\phi} - \frac{1}{\Delta_k^\phi + 4\epsilon_f} \geq \frac{c_1 \alpha_k (1 - \theta)^2 (\Delta_k^\phi)^2}{D^2 (\Delta_{k+1}^\phi) (\Delta_k^\phi + 4\epsilon_f)}.
\]
The left-hand-side of (4.13) can be bounded by
\[
\frac{1}{\Delta_{k+1}^\phi} - \frac{1}{\Delta_k^\phi + 4\epsilon_f} = \frac{1}{\Delta_{k+1}^\phi} + \frac{1}{\Delta_k^\phi} - \frac{1}{\Delta_k^\phi + 4\epsilon_f} - \frac{1}{\Delta_k^\phi + 4\epsilon_f}
\]
\[
\leq \frac{1}{\Delta_{k+1}^\phi} - \frac{1}{\Delta_k^\phi} \frac{4\epsilon_f}{\Delta_k^\phi + 4\epsilon_f}
\]
\[
\leq \frac{1}{\Delta_{k+1}^\phi} - \frac{1}{\Delta_k^\phi} \frac{4\epsilon_f}{\epsilon^2},
\]
where the last inequality holds since \( \Delta_k^\phi + 4\epsilon_f \geq \Delta_k^\phi \geq \epsilon \). The right-hand-side of (4.13) can be bounded by
\[
\frac{c_1 \alpha_k (1 - \theta)^2 (\Delta_k^\phi)^2}{D^2 (\Delta_{k+1}^\phi) (\Delta_k^\phi + 4\epsilon_f)} \geq \frac{c_1 \alpha_k (1 - \theta)^2 (\Delta_k^\phi)^2}{D^2 (\Delta_k^\phi + 4\epsilon_f)^2}
\]
\[
\geq \frac{c_1 \alpha_k (1 - \theta)^2}{4D^2}
\]
where the first inequality holds since \( \Delta_{k+1}^\phi \leq \Delta_k^\phi + 4\epsilon_f \), and the second due to the fact that \( \Delta_k^\phi \geq \epsilon > 4\epsilon_f \) (due to Assumption 4.6) and thus \( \frac{\Delta_k^\phi}{\Delta_k^\phi + 4\epsilon_f} \geq \frac{1}{2} \).

Therefore, we have,
\[
\frac{1}{\Delta_{k+1}^\phi} - \frac{1}{\Delta_k^\phi} \geq \frac{c_1 \alpha_k (1 - \theta)^2}{4D^2} - \frac{4\epsilon_f}{\epsilon^2},
\]
which completes the proof. \( \square \)
We now bound the amount of increase on false and successful iterations.

**Lemma 4.8.** Let Assumptions 1.3 and 4.5 hold, and consider any realization of Algorithm 1. For every iteration that is false and successful, we have

\[
\frac{1}{\Delta_k^{\phi}} \geq \frac{1}{\Delta_k^\phi} - \frac{4\epsilon_f}{\epsilon^2},
\]

where \(\Delta_k^\phi\) is defined in (4.10).

**Proof.** For every false and successful iteration, by Lemma 4.4 we have

\[
\phi(x_{k+1}) \leq \phi(x_k) - c_1\alpha_k\|g(x_k)\|^2 + 4\epsilon_f \leq \phi(x_k) + 4\epsilon_f.
\]

The rest of the proof is essentially a simplified version of the proof of Lemma 4.7, where the right hand side in (4.13) is simply replaced with 0. \(\square\)

Let,

\[
h(\alpha) = \frac{c_1\alpha(1-\theta)^2}{4D^2}, \quad \text{and} \quad r(\epsilon_f) = \frac{4\epsilon_f}{\epsilon^2}.
\]

By Lemmas 4.3, 4.7 and 4.8 and Assumption 4.6, for any realization of Algorithm 1 (which specifies the sequence \(\{\alpha_k, z_k\}\)) and \(k < N_e\), we have:

1. (Lemma 4.7) If \(k\) is a true and successful iteration, then

\[
z_{k+1} \geq z_k + h(\alpha_k) - r(\epsilon_f), \quad \text{and} \quad \alpha_{k+1} = \tau^{-1}\alpha_k.
\]

2. (Lemma 4.3) If \(\alpha_k \leq \bar{\alpha}\) and iteration \(k\) is true, then it is also successful.

3. (Lemma 4.8) If \(k\) is a false and successful iteration, then

\[
z_{k+1} \geq z_k - r(\epsilon_f).
\]

4. (Assumption 4.6) \(\frac{r(\epsilon_f)}{h(\alpha)} < \gamma\) for \(\gamma \in (0, 1)\).

Hence, Assumption 3.2 holds, with \(\bar{\alpha} > 0\) defined in (4.5), and \(h(\bar{\alpha})\) and \(r(\epsilon_f)\) defined in (4.16).

We now use Theorem 3.12 and the definitions of \(\bar{\alpha}\), \(h(\bar{\alpha})\), \(r(\epsilon_f)\) and \(Z\) to bound \(\mathbb{E}[N_e]\).

**Theorem 4.9.** Let Assumptions 1.1, 1.3, 4.2 and 4.5 hold. Moreover, let Assumption 4.6 hold, i.e., for some \(\gamma \in (0, 1)\),

\[
\mathbb{E}[N_e] \leq \frac{2(1-\delta)}{(1-2\delta)^2-\gamma} \left[ M \epsilon + (1-\gamma)\log_\epsilon \left( \frac{\bar{\alpha}}{\alpha_0} \right) \right],
\]

where \(M = \frac{4LD^2}{c_1(1-\delta)c_1(1-\theta)}\).

**Remark 4.10.** If \(\delta = \theta = \epsilon_f = 0\) our algorithm reduces to a deterministic line search algorithm with exact function evaluations and gradients. When \(\epsilon_f = 0\), \(\gamma\) can be chosen arbitrarily small, and the lower bound on \(\epsilon\) is 0. Notice that the complexity bound has two components, the first component \(\frac{8D^2L}{c_1(1-c_1)\epsilon}\) achieves its minimum value, \(\frac{2\alpha^2L^2}{c_1(1-c_1)}\), for \(c_1 = 1/2\) and is similar to the complexity bounds of the fixed step gradient descent method for convex functions, and the second term \(\log_\epsilon \left( \frac{2(1-c_1)}{\alpha_0L} \right)\) bounds the total number of unsuccessful iterations, on which \(\alpha_k\) is reduced.
4.2 Strongly Convex Functions

In this section, we analyze the expected complexity of Algorithm 1 in the case when \( \phi \) is a strongly convex function.

**Assumption 4.11. (Strong convexity of \( \phi \))** There exists a positive constant \( \mu \) such that

\[
\phi(x) \geq \phi(y) + \nabla \phi(y)^T (x - y) + \frac{\mu}{2} \|x - y\|^2, \quad \text{for all } x, y \in \mathbb{R}^n.
\]

Under Assumption 4.11, let \( \phi^* = \phi(x^*) \), where \( x^* \) is the minimizer of \( \phi \).

Recall the definition of \( \Delta_k^\phi \) (4.10). In this setting, we bound the number of iterations taken by Algorithm 1 until \( \Delta_k^\phi \leq \epsilon \) occurs. However, in this setting the bound is logarithmic in \( \frac{1}{\epsilon} \). Note, similar to the convex case, due to the noise in the function evaluations, \( \epsilon \) cannot be chosen to be arbitrarily small. We give a precise lower bound on \( \epsilon \), and thus explicitly derive a bound for the neighborhood of convergence.

**Assumption 4.12. (Neighborhood of convergence, strongly convex case)**

\[
\epsilon > \frac{4\epsilon_f}{\left(1 - \frac{2\mu c_1(1 - \theta)(1 - 2\theta - c_1(1 - \theta))}{L}\right)^{\gamma}} - 1
\]

for some \( \gamma \in (0, 1) \).

By Lemma 4.3, whenever \( A_k \leq \bar{A} \), then every true iteration is also successful. We now show that on true and successful iterations, \( Z_k \) increases by at least some function \( h(A_k) - r(\epsilon_f) \), for all \( k < N_{\epsilon} \).

**Lemma 4.13.** Let Assumptions 1.3 and 4.11 hold, and consider any realization of Algorithm 1. For every iteration that is true and successful, we have

\[
\log \left( \frac{1}{\Delta_{k+1}^\phi} \right) \geq \log \left( \frac{1}{\Delta_k^\phi} \right) - \log \left( 1 - \frac{\mu c_1 \alpha_k (1 - \theta)^2}{1 + 4\epsilon_f} \right), \quad (4.18)
\]

where \( \Delta_k^\phi \) is defined in (4.10).

**Proof.** Assumption 4.11 implies that \( (x = x_k \text{ and } y = x^*) \)

\[
\phi(x_k) - \phi^* \leq \frac{1}{2\mu} \|\nabla \phi(x_k)\|^2, \quad (4.19)
\]

see [13, Theorem 2.1.10]. Equivalently, using (4.4)

\[
\|g(x_k)\|^2 \geq (1 - \theta)^2 \|\nabla \phi(x_k)\|^2 \geq 2\mu (1 - \theta)^2 (\phi(x_k) - \phi^*). \quad (4.20)
\]

By equation (4.6), for every true and successful iteration we have

\[
\phi(x_{k+1}) \leq \phi(x_k) - c_1 \alpha_k (1 - \theta)^2 \|\nabla \phi(x_k)\|^2 + 4\epsilon_f
\]

\[
\leq \phi(x_k) - 2\mu c_1 \alpha_k (1 - \theta)^2 (\phi(x_k) - \phi^*) + 4\epsilon_f, \quad (4.21)
\]

and, thus,

\[
\phi(x_{k+1}) - \phi^* \leq (1 - 2\mu c_1 \alpha_k (1 - \theta)^2) (\phi(x_k) - \phi^*) + 4\epsilon_f.
\]

Since we have that \( \phi(x_k) - \phi^* \geq \epsilon \),

\[
\phi(x_{k+1}) - \phi^* \leq (1 - 2\mu c_1 \alpha_k (1 - \theta)^2) (\phi(x_k) - \phi^*) + 4\epsilon_f
\]

\[
\leq (1 - 2\mu c_1 \alpha_k (1 - \theta)^2) (\phi(x_k) - \phi^*) + \frac{4\epsilon_f}{\epsilon} (\phi(x_k) - \phi^*)
\]

\[
= \left(1 - 2\mu c_1 \alpha_k (1 - \theta)^2 + \frac{4\epsilon_f}{\epsilon}\right) (\phi(x_k) - \phi^*).
\]

16
Thus, using the definition of $\Delta^\phi_k$, we have
\[
\Delta^\phi_{k+1} \leq \left(1 - 2\mu c_1 \alpha_k (1 - \theta)^2 + \frac{4\epsilon f}{\epsilon}\right) \Delta^\phi_k.
\]
Since $\epsilon > 4\epsilon f$ (due to Assumption 4.12), we have
\[
\Delta^\phi_{k+1} \leq \left(1 - \mu c_1 \alpha_k (1 - \theta)^2 - 4\epsilon f \mu c_1 \alpha_k (1 - \theta)^2 + \frac{4\epsilon f}{\epsilon}\right) \Delta^\phi_k
\]
\[
= (1 - \mu c_1 \alpha_k (1 - \theta)^2) \left(1 + \frac{4\epsilon f}{\epsilon}\right) \Delta^\phi_k.
\]
(4.22)

Notice that since $\left(1 + \frac{4\epsilon f}{\epsilon}\right) > 0$, $\Delta^\phi_k > 0$ and $\Delta^\phi_{k+1} \geq 0$, this implies that $1 - \mu c_1 \alpha_k (1 - \theta)^2 \geq 0$. Now taking the inverse and then the log of both sides, we have
\[
\log \left(\frac{1}{\Delta^\phi_{k+1}}\right) \geq \log \left(\frac{1}{\Delta^\phi_k}\right) - \log \left(1 - \mu c_1 \alpha_k (1 - \theta)^2\right) - \log \left(1 + \frac{4\epsilon f}{\epsilon}\right),
\]
which completes the proof.

We note here that $1 - \mu c_1 \alpha_k (1 - \theta)^2 \geq 0$ holds for all $\alpha_k \leq \bar{\alpha}$ due to the constrants $\theta \in (0, \frac{\sqrt{c_1}}{2})$.

We now bound the amount of increase on false and successful iterations.

**Lemma 4.14.** Let Assumptions 1.3 and 4.11 hold, and consider any realization of Algorithm 1. For every iteration that is false and successful, we have
\[
\log \left(\frac{1}{\Delta^\phi_{k+1}}\right) \geq \log \left(\frac{1}{\Delta^\phi_k}\right) - \log \left(1 - \mu c_1 \alpha_k (1 - \theta)^2\right) - \log \left(1 + \frac{4\epsilon f}{\epsilon}\right),
\]
(4.23)
where $\Delta^\phi_k$ is defined in (4.10) and $\epsilon \geq 4\epsilon f$.

**Proof.** For every false and successful iteration, by Lemma 4.4 we have
\[
\phi(x_{k+1}) \leq \phi(x_k) - c_1 \alpha_k \|g(x_k)\|^2 + 4\epsilon f.
\]

The rest of the proof is essentially a simplification of the proof of Lemma 4.13 with the middle term of the right hand side of (4.21) replaced by 0.

Let
\[
h(\alpha) = -\log(1 - \mu c_1 (1 - \theta)^2 \alpha), \quad \text{and} \quad r(\epsilon_f) = \log \left(1 + \frac{4\epsilon f}{\epsilon}\right)
\]
(4.24)

By Lemmas 4.3, 4.13 and 4.14 and Assumption 4.12, for any realization of Algorithm 1 (which specifies the sequence $\{\alpha_k, z_k\}$) and $k < N_\epsilon$, we have:

1. (Lemma 4.13) If $k$ is a true and successful iteration, then
\[
z_{k+1} \geq z_k + h(\alpha_k) - r(\epsilon_f), \quad \text{and} \quad \alpha_{k+1} = \tau^{-1} \alpha_k.
\]
2. (Lemma 4.3) If $\alpha_k \leq \bar{\alpha}$ and iteration $k$ is true, then it is also successful.
3. (Lemma 4.14) If $k$ is a false and successful iteration, then
$$z_{k+1} \geq z_k - \log \left( 1 + \frac{4\epsilon_f}{\epsilon} \right).$$

4. (Assumption 4.12) $\frac{r(\epsilon_f)}{h(\bar{\alpha})} < \gamma$ for some $\gamma \in (0, 1)$.

Hence, Assumption 3.2 holds, with $\bar{\alpha} > 0$ defined in (4.5), and $h(A_k)$ and $r(\epsilon_f)$ defined in (4.16).

We now use Theorem 3.12 and the definitions of $\bar{\alpha}, h(\bar{\alpha}), r(\epsilon_f)$ and $Z_{\epsilon_f}$ to bound $E[N_\epsilon]$.

**Theorem 4.15.** Let Assumptions 1.1, 1.3, 4.2 and 4.11 hold. Moreover, let Assumption 4.12 hold, i.e.,
$$\epsilon > \frac{4\epsilon_f}{(1 - 2\theta)(1 - 2\theta - c_1(1 - \theta))^{\gamma} - 1}.$$ Then the expected number of iterations that Algorithm 1 takes until $\phi(X_k) - \phi^* \leq \epsilon$ occurs is bounded as follows
$$E[N_\epsilon] \leq \frac{2(1 - \delta)}{(1 - 2\delta)^2 \gamma_{c_1(1 - \theta)}} \left[ 2 \log_1/M \left( \frac{1}{\epsilon} \right) + (1 - \gamma) \log_{\gamma} \left( \frac{\bar{\alpha}}{\alpha_0} \right) \right],$$
where $M = 1 - \frac{4\epsilon_f(1 - 2\theta - c_1(1 - \theta))(1 - \theta)}{L}$.

**Remark 4.16.** Again, if $\delta = \theta = \epsilon_f = 0$ our algorithm reduces to a deterministic line search algorithm with exact function evaluations and gradients. The complexity bound has two components, $4 \log_1/M \left( \frac{1}{\epsilon} \right)$ where $M = 1 - \frac{4\epsilon_f(1 - 2\theta - c_1(1 - \theta))(1 - \theta)}{L}$ achieves its minimum value, $1 - \frac{4\epsilon_f}{L}$, for $c_1 = 1/2$ and is similar to complexity bounds of the fixed step gradient descent method for strongly convex functions, and the second term again is the bound on the total number of unsuccessful iterations.

### 4.3 Nonconvex Functions

In this section, we analyze the expected complexity of Algorithm 1 in the case when $\phi$ is a nonconvex function. Again, we first specify the neighborhood of convergence.

**Assumption 4.17. (Neighborhood of convergence, nonconvex case)**
$$\epsilon^2 > \frac{2\epsilon_f L}{\gamma_c(1 - \theta)(1 - 2\theta - c_1(1 - \theta))},$$
for some $\gamma \in (0, 1)$.

Let
$$h(\alpha) = c_1\alpha(1 - \theta)^2\|\nabla \phi(x_k)\|^2,$$
and
$$r(\epsilon_f) = 4\epsilon_f.$$ (4.26)

By Lemmas 4.3 and 4.4 and Assumption 4.17, for any realization of Algorithm 1 (which specifies the sequence $\{\alpha, z_k\}$) and $k < N_\epsilon$, we have:

1. (Lemma 4.3) If $k$ is a true and successful iteration, then
$$z_{k+1} \geq z_k + h(\alpha_k) - r(\epsilon_f) \quad \text{and} \quad \alpha_{k+1} = \tau^{-1} \alpha_k.$$

2. (Lemma 4.3) If $\alpha_k \leq \bar{\alpha}$ and iteration $k$ is true, then it is also successful.

3. (Lemma 4.4) If $k$ is a false and successful iteration, then
$$z_{k+1} \geq z_k - 4\epsilon_f.$$
4. (Assumption 4.17) \( \frac{r(\epsilon)}{r(\epsilon_f)} < \gamma \) for some \( \gamma \in (0, 1) \).

Hence, Assumption 3.2 holds, with \( \bar{\alpha} > 0 \) defined in (4.5), and \( h(A_k) \) and \( r(\epsilon_f) \) defined in (4.26).

We now use Theorem 3.12 and the definitions of \( \bar{\alpha} \), \( h(\bar{\alpha}) \), \( r(\epsilon_f) \) and \( Z_\epsilon \) to bound \( \mathbb{E}[N_\epsilon] \).

**Theorem 4.18.** Let Assumptions 1.1, 1.2, 1.3 and 4.2. Moreover, let Assumption 4.17 hold, i.e.,

\[
\epsilon^2 > \frac{2\epsilon_f L}{\gamma c_1 (1 - \theta)(1 - 2\theta - c_1 (1 - \theta))},
\]

(4.27)

Then, the expected number of iterations that Algorithm 1 takes until \( \|\nabla \phi(X_k)\| \leq \epsilon \) occurs is bounded as follows

\[
\mathbb{E}[N_\epsilon] \leq \frac{2(1 - \delta)}{(1 - 2\delta)^2 - \gamma} \left[ \frac{M}{\epsilon^2} + (1 - \gamma) \log_\tau \left( \frac{\bar{\alpha}}{\alpha_0} \right) \right],
\]

(4.28)

where \( M = \frac{(\bar{\phi}(x_0) - \hat{\phi})L_{\bar{\gamma}}}{c_1 (1 - c_1)} \).

**Remark 4.19.** Again, if \( \delta = \theta = \epsilon_f = 0 \) our algorithm reduces to a deterministic line search with the exact gradients. The complexity bound has two components, \( \frac{2M}{\epsilon^2} \), where \( M = \frac{(\phi(x_0) - \phi)L_{\bar{\gamma}}}{c_1 (1 - c_1)} \) achieves its minimum value, \( 4(f(x_0) - \hat{f})L \), for \( c_1 = 1/2 \) and is similar to complexity bounds of the fixed step gradient descent for nonconvex functions, and the second term, as before, is the bound on the total number of unsuccessful iterations.

### 4.4 General descent

For simplicity, in the analysis of the previous sections we assumed that the search direction at every iteration was defined as \( d_k = -g(x_k) \). Here, we show how our analysis can be extended to account for more general search direction, e.g., quasi-Newton search direction where \( d_k = -H_k g(x_k) \), provided the search directions satisfy:

- There exists a constant \( \beta \), such that:
  \[
  \frac{d_k^T g(x_k)}{\|d_k\| \|g(x_k)\|} \leq -\beta, \quad \text{for all } k,
  \]
  (4.29)

- There exist constants \( \kappa_1, \kappa_2 > 0 \), such that:
  \[
  \kappa_1 \|g(x_k)\| \leq \|d_k\| \leq \kappa_2 \|g(x_k)\|, \quad \text{for all } k,
  \]
  (4.30)

Together with (4.3). Of course, in this setting, the modified line search would be given by (2.1), and the convergence analysis would have dependence on \( \beta, \kappa_1 \) and \( \kappa_2 \).

All we need to do is to derive an expression for \( \bar{\alpha} \) for the general search direction case, and prove analogues of Lemmas 4.3 and 4.4. First, we change the bound on \( \theta \) in the Definition 4.1. In particular we will require that \( \theta \in \left[ 0, \frac{(1-c_1)\beta}{1 + (1-c_1)\beta} \right] \). Now we can prove the following lemma.

**Lemma 4.20.** Let Assumption 1.1 hold. For every realization of Algorithm 1, if iteration \( k \) is true (i.e., \( I_k = 1 \)), and if

\[
\alpha_k \leq \bar{\alpha} = \frac{2}{L \kappa_2} \left[ \frac{(1-c_1)(1-\theta)\beta - \theta}{1-\theta} \right],
\]

(4.31)

then (2.1) holds. In other words, when (4.31) holds, any true iteration is also a successful iteration. Moreover, for every true and successful iteration,

\[
\phi(x_{k+1}) \leq \phi(x_k) - c_1 \alpha_k \beta \kappa_1 (1-\theta)^2 \|\nabla \phi(x_k)\|^2 + 4\epsilon_f.
\]

(4.32)
Proof. The proof is very similar to that of Lemma 4.3. First from Assumption 1.1 we have

\[ \phi(x_{k+1}) \leq \phi(x_k) + \alpha_k d_k^T \nabla \phi(x_k) + \frac{L}{2} \| \alpha_k d_k \|^2. \]  

(4.33)

Applying the Cauchy-Schwarz inequality, (4.3) and (4.4), for every true iteration

\[ \phi(x_k - \alpha_k d_k) \leq \phi(x_k) + \alpha_k d_k^T \nabla \phi(x_k) + \frac{\alpha_k^2 L}{2} \| d_k \|^2 \]

\[ = \phi(x_k) + \alpha_k d_k^T (\nabla \phi(x_k) - g(x_k)) + \alpha_k d_k^T g(x_k) + \frac{\alpha_k^2 L}{2} \| d_k \|^2 \]

\[ \leq \phi(x_k) + \alpha_k \| d_k \| \| \nabla \phi(x_k) - g(x_k) \| + \alpha_k d_k^T g(x_k) + \frac{\alpha_k^2 L}{2} \| d_k \| \| g(x_k) \| \]

\[ \leq \phi(x_k) + \alpha_k d_k^T g(x_k) + \alpha_k \left[ \frac{\theta}{1 - \theta} + \frac{\alpha_k L \kappa_2}{2} \right] \| d_k \| \| g(x_k) \|. \]

Now, using Assumption 1.3, we have

\[ f(x_k - \alpha_k g(x_k)) \leq f(x_k) + \alpha_k d_k^T g(x_k) + \alpha_k \left[ \frac{\theta}{1 - \theta} + \frac{\alpha_k L \kappa_2}{2} \right] \| d_k \| \| g(x_k) \| + 2 \epsilon_f. \]

From this we conclude that (2.1) holds whenever

\[ f(x_k) + \alpha_k d_k^T g(x_k) + \alpha_k \left[ \frac{\theta}{1 - \theta} + \frac{\alpha_k L \kappa_2}{2} \right] \| d_k \| \| g(x_k) \| + 2 \epsilon_f \leq f(x_k) + c_1 \alpha_k d_k^T g(x_k) + 2 \epsilon_f \]

or equivalently, since \( \alpha_k > 0 \),

\[ \left[ \frac{\theta}{1 - \theta} + \frac{\alpha_k L \kappa_2}{2} \right] \| d_k \| \| g(x_k) \| \leq -(1 - c_1) d_k^T g(x_k). \]

Using (4.29), the above expression holds whenever \( \alpha_k \) satisfies (4.31). Therefore, using Assumption 1.3, (4.30), and (4.4), for every true and successful iteration we have

\[ \phi(x_{k+1}) \leq \phi(x_k) - c_1 \alpha_k \beta \kappa_1 (1 - \theta)^2 \| \nabla \phi(x_k) \|^2 + 4 \epsilon_f, \]

which completes the proof. □

Next, we state and prove a result for the case of false and successful iterations.

Lemma 4.21. For every false and successful iteration of Algorithm 1, we have

\[ \phi(x_{k+1}) \leq \phi(x_k) - c_1 \alpha_k \beta \kappa_1 \| g(x_k) \|^2 + 4 \epsilon_f. \]  

(4.34)

Proof. For every successful iteration we have

\[ f(x_{k+1}) \leq f(x_k) + c_1 \alpha_k d_k^T g(x_k) + 2 \epsilon_f. \]

Thus, by Assumption 1.3, (4.30), and (4.4)

\[ \phi(x_{k+1}) \leq \phi(x_k) + c_1 \alpha_k d_k^T g(x_k) + 4 \epsilon_f \]

\[ \leq \phi(x_k) - c_1 \alpha_k \beta \| d_k \| \| g(x_k) \| + 4 \epsilon_f \]

\[ \leq \phi(x_k) - c_1 \alpha_k \beta \kappa_1 \| g(x_k) \|^2 + 4 \epsilon_f, \]

which is a repetition of the last part of the proof of Lemma 4.20. □

The rest of the analysis (deriving expected complexity bounds) applies almost without change, taking into account the influence of the constants \( \beta, \kappa_1 \) and \( \kappa_2 \).
5 Final Remarks

We presented the analysis of a modified line search method that can be applied to functions with bounded noise, and where the gradient approximations $g(x_k)$ are possibly random, e.g., Gaussian smoothed gradients [14, 18] or sphere smoothed gradients [7, 8]. However, as a special case, we recover results for gradient approximations that are not random ($\delta = 0$), e.g., finite difference approximations [1, 9] or linear interpolation gradient approximations [6].

Furthermore, we discuss the effect of the parameter $\gamma$, that plays a crucial role in the analysis presented. This parameter depends on the error in the function evaluations, and effectively controls the size of the neighborhood of convergence, i.e., the lower bound on the $\epsilon$. When there is zero error in the function evaluations, i.e., $\epsilon(x) = 0$ for all $x \in \mathbb{R}^n$, $\gamma$ can be chosen arbitrarily close to zero, in which case we recover the exact convergence results from [5].

Finally, while our analysis assumes that the step size parameter is chosen using an adaptive line search procedure (Algorithm 2), and thus varies at every iteration, it also holds for a constant step size parameter choice. Namely, if $\alpha_0 \leq \bar{\alpha}$ and $\tau = 1$, then $\alpha_k \leq \bar{\alpha}$ for all $k$, and all true iterations are also successful iterations. Thus, as a special case of the analysis presented in Section 4, we recover results for a fixed step size parameter procedure. We should note that the second term in the complexity bounds is zero in the case where $\tau = 1$ and $\alpha_0 = \bar{\alpha}$.

Acknowledgements

This work was partially supported by NSF Grants CCF 16-18717 and TRIPODS 17-40796, DARPA Lagrange award HR-001117S0039 and a Google Faculty Award.
References

[1] Albert S Berahas, Richard H Byrd, and Jorge Nocedal. Derivative-free optimization of noisy functions via quasi-newton methods. *SIAM Journal on Optimization*, 29(2):965–993, 2019.

[2] Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *arXiv preprint arXiv:1905.01332*, 2019.

[3] Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical programming*, 134(1):127–155, 2012.

[4] Richard G Carter. On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991.

[5] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, pages 1–39, 2018.

[6] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to Derivative-free Optimization*. MPS-SIAM Optimization series. SIAM, Philadelphia, USA, 2008.

[7] Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.

[8] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[9] C T Kelley. *Implicit filtering*, volume 23. SIAM, 2011.

[10] Jeffrey Larson, Matt Menickelly, and Stefan Wild. Derivative-free optimization methods. *Submitted*, 2019.

[11] Alvaro Maggiar, Andreas Wächter, Irina S Dolinskaya, and Jeremy Staum. A derivative-free trust-region algorithm for the optimization of functions smoothed via gaussian convolution using adaptive multiple importance sampling. *SIAM Journal on Optimization*, 28(2):1478–1507, 2018.

[12] J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. (20):172–191, 2009.

[13] Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, Boston, MA, 2004.

[14] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[15] Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, NY, USA, 2nd edition, 2006.

[16] Courtney Paquette and Katya Scheinberg. A stochastic line search method with convergence rate analysis. *arXiv preprint arXiv:1807.07994*, 2018.

[17] Raghu Pasupathy, Peter Glynn, Soumyadip Ghosh, and Fatemeh S Hashemi. On sampling rates in simulation-based recursions. *SIAM Journal on Optimization*, 28(1):45–73, 2018.

[18] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. Technical Report arXiv:1703.03864, 2016.