Global convergence rate analysis of unconstrained optimization methods based on probabilistic models

C. Cartis[*] K. Scheinberg[†]

May 22, 2015

Abstract

We present global convergence rates for a line-search method which is based on random first-order models and directions whose quality is ensured only with certain probability. We show that in terms of the order of the accuracy, the evaluation complexity of such a method is the same as its counterparts that use deterministic accurate models; the use of probabilistic models only increases the complexity by a constant, which depends on the probability of the models being good. We particularize and improve these results in the convex and strongly convex case.

We also analyse a probabilistic cubic regularization variant that allows approximate probabilistic second-order models and show improved complexity bounds compared to probabilistic first-order methods; again, as a function of the accuracy, the probabilistic cubic regularization bounds are of the same (optimal) order as for the deterministic case.

Keywords: line-search methods, cubic regularization methods, random models, global convergence analysis.

1 Introduction

We consider in this paper the unconstrained optimization problem

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

where the first (and second, when specified) derivatives of the objective function $f(x)$ are assumed to exist and be (globally) Lipschitz continuous.

Most unconstrained optimization methods rely on approximate local information to compute a local descent step in such a way that sufficient decrease of the objective function is achieved.

[*] Mathematical Institute, University of Oxford, Radcliffe Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, United Kingdom (`cartis@maths.ox.ac.uk`). This work was partially supported by the Oxford University EPSRC Platform Grant EP/I01893X/1.

[†] Department of Industrial and Systems Engineering, Lehigh University, Harold S. Mohler Laboratory, 200 West Packer Avenue, Bethlehem, PA 18015-1582, USA (`katyas@lehigh.edu`). The work of this author is partially supported by NSF Grants DMS 10-16571, DMS 13-19356, CCF-1320137, AFOSR Grant FA9550-11-1-0239, and DARPA grant FA 9550-12-1-0406 negotiated by AFOSR.

To ensure such sufficient decrease, the step has to satisfy certain requirements. Often in practical applications ensuring these requirements for each step is prohibitively expensive or impossible. This may be due to the fact that derivative information about the objective function is not available or because full gradient (and Hessian) are too expensive to compute, or a model of the objective function is too expensive to optimize accurately.

Recently, there has been a significant increase in interest in unconstrained optimization methods with inexact information. Some of these methods consider the case when gradient information is inaccurate. This error in the gradient computation may simply be bounded in the worst case (deterministically), see, for example, [10, 19], or the error is random and the estimated gradient is accurate in expectation, as in stochastic gradient algorithms, see for example, [11, 18, 21, 20]. These methods are typically applied in a convex setting and do not extend to non-convex cases. Complexity bounds are derived that bound the expected accuracy that is achieved after a given number of iterations.

In the nonlinear optimization setting, the complexity of various unconstrained methods has been derived under exact derivative information [6, 7, 16], and also under inexact information, where the errors are bounded in a deterministic fashion [2, 5, 10, 13, 19]. In all the cases of the deterministic inexact setting, traditional optimization algorithms such as line search, trust region or adaptive regularization algorithms are applied with little modification and work in practice as well as in theory, while the error is assumed to be bounded in some decaying manner *at each iteration*. In contrast, the methods based on stochastic estimates of the derivatives, do not assume deterministically bounded errors, however they are quite different from the "traditional" methods in their strategy for step size selection and averaging of the iterates. In other words, they are not simple counterparts of the deterministic methods.

Our purpose in this paper is to derive a class of methods which inherit the best properties of traditional deterministic algorithms, and yet relax the assumption that the derivative/model error is bounded in a deterministic manner. Moreover, we do not assume that the error is zero in expectation or that it has a bounded variance. Our results apply in the setting where at each iteration, with sufficiently high probability, the error is bounded in a decaying manner, while in the remaining cases, this error can be arbitrarily large. In this paper, we assume that the error may happen in the computation of the derivatives and search directions, but that there is no error in the function evaluations, when success of an iterate has to be validated.

Recently several methods for unconstrained black-box optimization have been proposed, which rely on random models or directions [1, 12, 15], but are applied to deterministic functions. In this paper we take this line of work one step further by establishing expected convergence rates for several schemes based on one generic analytical framework.

We consider four cases and derive four different complexity bounds. In particular, we analyze a line search method based on random models, for the cases of general non convex, convex and strongly convex functions. We also analyze a second order method - an adaptive regularization method with cubics [6, 7] - which is known to achieve the optimal convergence rate for the nonconvex smooth functions [4] and we show that the same convergence rate holds in expectation.

In summary, our results differ from existing literature using inexact, stochastic or random information in the following main points:

- Our models are assumed to be "good" with some probability, but there is no other assumptions on the expected values or variance of the model parameters.

- The methods that we analyze are essentially the exact counterparts of the deterministic methods, and do not require averaging of the iterates or any other significant changes. We believe that, amongst other things, our analysis helps to understand the convergence properties of practical algorithms, that do not always seek to ensure theoretically required model quality.

- Our main convergence rate results provide a bound on the *expected number of iterations* that the algorithms take before they achieve a desired level of accuracy. This is in contrast to a typical analysis of randomized or stochastic methods, where what is bounded is the expected error after a given number of iterations. Both bounds are useful, but we believe that the bound on the expected number of steps is a somewhat more meaningful complexity bound in our setting. The only other work that we are aware of which provides bounds in terms of the number of required steps is [12] where probabilistic bounds are derived in the particular context of random direct search.

An additional goal of this paper is to present a general theoretical framework, which could be used to analyze the behavior of other algorithms, and different possible model construction mechanisms under the assumption that the objective function is deterministic. We propose a general analysis of an optimization scheme by reducing it to the analysis of a stochastic process. Convergence results for a trust region method in [1] also rely on a stochastic process analysis, but only in terms of behavior in the limit. These results have now been extended to noisy (stochastic) functions, see [8, 9]. Deriving convergence *rates* for methods applied to stochastic functions is the subject of future work and is likely to depend on the results in this paper.

The rest of the paper is organized as follows. In Section 2 we describe the general scheme which encompasses several unconstrained optimization methods. This scheme is based on using random models, which are assumed to satisfy some "quality" conditions with probability at least p, conditioned on the past. Applying this optimization scheme results in a stochastic process, whose behavior is analyzed in the later parts of Section 2. Analysis of the stochastic process allows us to bound the expected number of steps of our generic scheme until a desired accuracy is reached. In Section 3 we analyze a linesearch algorithm based on random models and show how its behavior fits into our general framework for the cases of non convex, convex and strongly convex functions. In Section 4 we apply our generic analysis to the case of the Adaptive Regularization method with Cubics (ARC). Finally, in Section 5 we describe different settings where the models of the objective functions satisfy the probabilistic conditions of our schemes.

2 A general optimization scheme with random models

This section presents the main features of our algorithms and analysis, in a general framework that we will, in subsequent sections, particularize to specific algorithms (such as linesearch and cubic regularization) and classes of functions (convex, nonconvex). The reasons for the initial generic approach is to avoid repetition of the common elements of the analysis for the different algorithms and to emphasize the key ingredients of our analysis, which is possibly applicable to other algorithms (provided they satisfy our framework).

2.1 A general optimization scheme

We first describe a generic algorithmic framework that encompasses the main components of the unconstrained optimization schemes we analyze in this paper. The scheme relies on building a model of the objective function at each iteration, minimizing this model or reducing it in a sufficient manner and considering the step which is dependent on a stepsize parameter and which provides the model reduction (the stepsize parameter may be present in the model or independent of it). This step determines a new candidate point. The function value is then computed (accurately) at the new candidate point. If the function reduction provided by the candidate point is deemed sufficient, then the iteration is declared successful, the candidate point becomes the new iterate and the step size parameter is increased. Otherwise, the iteration is unsuccessful, the iterate is not updated and the step size parameter is reduced.

We summarize the main steps of the scheme below.

Algorithm 2.1 Generic optimization framework based on random models

Initialization
Choose a class of (possibly random) models $m_k(x)$, *choose constants* $\gamma \in (0,1)$, $\theta \in (0,1)$, $\alpha_{max} > 0$. *Initialize the algorithm by choosing* x_0, $m_0(x)$, $0 < \alpha_0 < \alpha_{\max}$.

1. Compute a model and a step
Compute a local (possibly random) model $m_k(x)$ *of* f *around* x^k.
Compute a step $s^k(\alpha_k)$ *which reduces* $m_k(x)$, *where the parameter* $\alpha_k > 0$ *is present in the model or in the step calculation.*

2. Check sufficient decrease
Compute $f(x^k + s^k(\alpha_k))$ *and check if sufficient reduction (parametrized by* θ*) is achieved in* f *with respect to* $m_k(x^k) - m_k(x^k + s^k(\alpha_k))$.

3. Successful step
If sufficient reduction is achieved then, $x^{k+1} := x^k + s^k(\alpha_k)$, *set* $\alpha_{k+1} = \min\{\alpha_{max}, \gamma^{-1}\alpha_k\}$. *Let* $k := k+1$.

4. Unsuccessful step
Otherwise, $x^{k+1} := x^k$, *set* $\alpha_{k+1} = \gamma\alpha_k$. *Let* $k := k+1$.

Let us illustrate how the above scheme relates to standard optimization methods. In *line-search methods*, one minimizes a linear model $m_k(x) = f(x^k) + (x - x^k)^T g^k$ (subject to some normalization), or a quadratic one $m_k(x) = f(x^k) + (x - x^k)^T g^k + \frac{1}{2}(x - x^k)^\top b^k (x - x^k)$ (when the latter is well-defined, with b^k - a Hessian approximation matrix), to find directions $d^k = -g^k$ or $d^k = -(b^k)^{-1} g^k$, respectively. Then the step is defined as $s^k(\alpha_k) = \alpha_k d^k$ for some α_k and, commonly, the (Armijo) decrease condition is checked,

$$f(x^k) - f(x^k + s^k(\alpha_k)) \geq -\theta s^k(\alpha_k)^T g^k,$$

whose right-hand side is nothing but the change in the above linear model $m_k(x^k) - m_k(x^k + s^k(\alpha_k))$. Note that if the model stays the same in that $m_k(x) \equiv m_{k-1}(x)$ for each k, such that

$(k-1)$st iteration is unsuccessful, then the above framework essentially reduces to a standard deterministic linesearch.

In the case of *cubic regularization methods*, $s^k(\alpha_k)$ is computed to approximately minimize a cubic model $m_k(x) = f(x^k) + (x-x^k)^T g^k + \frac{1}{2}(x-x^k)^\top b^k(x-x^k) + \frac{1}{3\alpha_k}\|x-x^k\|^3$ and the sufficient decrease condition is

$$\frac{f(x^k) - f(x^k + s^k(\alpha_k))}{m(x^k) - m(x^k + s^k(\alpha_k))} \geq \theta > 0.$$

Note that here as well, in the deterministic case, $g^k = g^{k-1}$ and $b^k = b^{k-1}$ for each k such that $(k-1)$st iteration is unsuccessful but $\alpha_k \neq \alpha_{k-1}$.

The key assumption in the usual deterministic case is that the models $m_k(x)$ are sufficiently accurate in a small neighborhood of the current iterate x^k. The goal of this paper is to relax this requirement and allow the use of random local models which are accurate only with certain probability (conditioned on the past). In that case, note that the models need to be re-drawn after each iteration, whether successful or not.

Note that our general setting includes the cases when the model (the derivative information, for example) is always accurate, but the step s^k is computed approximately, in a probabilistic manner. For example, s^k can be an approximation of $-(b^k)^{-1}g^k$. It is easy to see how randomness in s^k calculation can be viewed as the randomness in the model, by considering that instead of the accurate model

$$f(x^k) + (x-x^k)^T g^k + \frac{1}{2}(x-x^k)^\top b^k(x-x^k)$$

we use an approximate model

$$m_k(x) = f(x^k) + (x-x^k)^T b^k s^k + \frac{1}{2}(x-x^k)^\top b^k(x-x^k).$$

Hence, as long as the accuracy requirements are carried over accordingly the approximate random models subsume the case of approximate random step computations. The next section makes precise our requirements on the probabilistic models.

2.2 Generic probabilistic models

We will now introduce the key probabilistic ingredients of our scheme. In particular we assume that our models m_k are random and that they satisfy some notion of good quality with some probability p. We will consider random models M_k, and then use the notation $m_k = M_k(\omega_k)$ for their realizations. The randomness of the models will imply the randomness of the points x^k, the step length parameter α_k, the computed steps s^k and other quantities produced by the algorithm. Thus, in our paper, these random variables will be denoted by X^k, $\mathcal{A}_k$, S^k and so on, respectively, while $x^k = X^k(\omega_k)$, $\alpha_k = \mathcal{A}_k(\omega_k)$, $s^k = S^k(\omega_k)$, etc, denote their realizations (we will omit the ω_k in the notation for brevity).

For each specific optimization method, we will define a notion of sufficiently accurate models. The desired accuracy of the model depends on the current iterate x^k, step parameter α_k and, possibly, the step $s^k(\alpha_k)$. This notion involves model properties which make sufficient decrease in f achievable by the step $s^k(\alpha_k)$. Specific conditions on the models will be stated for each algorithm in the respective sections and how these conditions may be achieved will be discussed in Section 5.

Definition 2.1 [sufficiently accurate models; true and false iterations] *We say that a sequence of random models $\{M_k\}$ is (p)-probabilistically "sufficiently accurate" for a corresponding sequence $\{\mathcal{A}_k, X^k\}$, if the events*

$$I_k \ = \ \{M_k \textit{ is a sufficiently accurate model of } f \textit{ for the given } X^k \textit{and } \mathcal{A}_k\}$$

satisfy the following submartingale-like condition

$$P(I_k | \, \mathcal{F}^M_{k-1}) \ \geq \ p,$$

where $\mathcal{F}^M_{k-1} = \sigma(M_0, \ldots, M_{k-1})$ is the σ-algebra generated by $M_0, \ldots, M_{k-1}$ - in other words, the history of the algorithm up to iteration k.

We say that iteration k is a **true** *iteration if event I_k occurs. Otherwise the iteration is called* **false***.*

Note that M_k is a random model that, given the past history, encompasses all the randomness of iteration k of our algorithm. The iterates X^k and the step length parameter$\mathcal{A}_k$ are random variables defined over the σ-algebra generated by $M_0, \ldots, M_{k-1}$. Each M_k depends on X^k and $\mathcal{A}_k$ and hence on $M_0, \ldots, M_{k-1}$. Definition 2.1 serves to enforce the following property: even though the accuracy of M_k may be dependent on the history, $(M_1, \ldots, M_{k-1})$, via its dependence on X^k and $\mathcal{A}_k$, it is sufficiently good with probability at least p, regardless of that history. This condition is more reasonable than complete independence of M^k from the past, which is difficult to ensure. It is important to note that, from this assumption, it follows that whether or not the step is deemed successful and the iterate x^k is updated, our scheme always updates the model m_k, unless m_k is somehow known to be sufficiently accurate for $x^{k+1} = x^k$ and α_{k+1}. We will discuss this in more detail in Section 5.

When Algorithm 2.1 is based on probabilistic models (and all its specific variants under consideration), it results in a discrete time stochastic process. This stochastic process encompasses random elements such $\mathcal{A}_k$, X^k, S^k, which are directly computed by the algorithm, but also some quantities that can be derived as functions of $\mathcal{A}_k$, X^k, S^k, such as $f(X^k)$, $\|\nabla f(X^k)\|$ and a quantity F_k, which we will use to denote some measure of progress towards optimality. Each realization of the sequence of random models results in a realization of the algorithm, which in turn produces the corresponding sequences $\{\alpha_k\}$, $\{x^k\}$, $\{s^k\}$, $\{f(x^k)\}$, $\{\|\nabla f(x^k)\|\}$ and $\{f_k\}$[1]. We will analyze the stochastic processes for different cases of the algorithm and under different conditions on f. Each time we will restrict our attention to some of the random quantities that belong to this process and will ignore the rest, for the brevity of the presentation. Hence when we say that Algorithm 2.1 generates the stochastic process $\{X^k, \mathcal{A}_k\}$, this means we want to focus on the properties of these random variables, but keeping in mind that there are other random quantities in this stochastic process.

We will derive complexity bounds for each algorithm in the following sense. We will define the accuracy goal that we aim to reach and then we will bound the expected number of steps that the algorithm takes until this goal is achieved. The analyses will follow common steps and will make use of auxiliary stochastic processes, which have a simple structure and serve as a lower bound on our main process in each case. Below we describe the main ingredients of the analysis. We then apply these steps to each case under consideration.

[1] Note that throughout, $f(x^k) \neq f_k$, since f_k is a related measure of progress towards optimality.

2.3 Elements of global convergence rate analysis

First we recall a standard notion from stochastic processes.

Hitting time. For a given discrete time stochastic process, Z_t, recall the concept of a *hitting time* for an event $\{Z_t \in S\}$, which we denote by $T_S^{Z_t}$. This is a random variable, defined as $T_S = \min\{t : \; Z_t \in S\}$ - the first time the event $\{Z_t \in S\}$ occurs. In our context, set S will either be a set of real numbers larger than some given value, or smaller than some other given value. Since it will be clear from the context, for simplicity, we will use the value instead of the definition of the set S when defining the hitting time.

Number of iterations N_ϵ to reach ϵ accuracy. Given a level of accuracy ϵ, we aim to derive a bound on the expected number of iterations $\mathbb{E}(N_\epsilon)$ which occur in the algorithm until the given accuracy level is reached. The number of iterations N_ϵ is a random variable, which can be defined as a hitting time of some stochastic process, dependent on the case under analysis. In particular,

- If $f(x)$ is not known to be convex, then $N_\epsilon = T_\epsilon^{\|\nabla f(X^k)\|}$ is the hitting time for $\{\|\nabla f(X_k)\| \leq \epsilon\}$, namely, the number of steps the algorithm takes until $\|\nabla f(X^k)\| \leq \epsilon$ occurs for the first time.
- If $f(x)$ is convex or strongly convex then $N_\epsilon = T_\epsilon^{f(X^k)-f_*}$ is the hitting time for $\{f(X^k) - f_* \leq \epsilon\}$, namely, the number of steps the algorithm takes until $f(X^k) - f_* \leq \epsilon$ occurs for the first time, where $f_* = f(x^*)$ with x^*, a global minimizer of f.

Instead of estimating $\mathbb{E}(N_\epsilon)$ directly we will bound N_ϵ by another random variable, which will be different for different cases, but will obey some common properties. Towards this end we need to define the following random variable and its upper bound.

Measure of progress towards optimality, F_k. This measure is defined by the total function decrease or by the distance to the optimum. In particular,

- If $f(x)$ is not known to be convex, then $F_k = f(X^0) - f(X^k)$.
- If $f(x)$ is convex, then $F_k = 1/(f(X^k) - f_*)$.
- If $f(x)$ is strongly convex, then $F_k = \log(1/(f(X^k) - f_*))$.

Upper bound F_ϵ on F_k. From the algorithm construction, F_k defined above is always non-decreasing and there exists a deterministic upper bound F_ϵ in each case, defined as follows.

- If $f(x)$ is not known to be convex, then $F_\epsilon = f(X^0) - f_*$, where f_* is a global lower bound on f.
- If $f(x)$ is convex, then $F_\epsilon = 1/\epsilon$.
- If $f(x)$ is strongly convex, then $F_\epsilon = \log(1/\epsilon)$.

We observe that F_k is a nondecreasing process and F_ϵ is the largest possible value that F_k can achieve until desired accuracy is reached, that is until N_ϵ iterations have occurred. In other words,

$$\mathbb{E}(N_\epsilon) \leq \mathbb{E}(T_{F_\epsilon}^{F_K}) \tag{1}$$

It is sufficient, hence, to bound $\mathbb{E}(T_{F_\epsilon}^{F_K})$, which we devote the rest of the analysis to (firstly, in a general framework that we then particularize to the different algorithms).

Our analysis will be based on the following observations, which are borrowed from the global rate analysis of the deterministic methods [14].

- **Guaranteed amount of increase in f_k.** Until N_ϵ iterations have been reached, if the kth iteration is true and successful, then f_k is increased by an amount proportional to α_k.
- **Guaranteed threshhold for α_k.** There exists a constant, which we will call C, such that if $\alpha_k \leq C$ and the kth iteration is true, then the kth iteration is also successful, and hence $\alpha_{k+1} = \gamma^{-1}\alpha_k$. This constant C depends on the algorithm and Lipschitz constants of f.
- **Bound on the number of iterations.** If all iterations were true, then by the above observations, $\alpha_k \geq \gamma C$ and, hence, f_k increases by at least a constant for all k. From this a bound on the number of iterations, until f^k have reached F_ϵ, can be derived.

In our case not all iterations are true, however, as we will show, when $\mathcal{A}_k \leq C$, then iterations "tend" to be true, $\mathcal{A}_k$ "tends" to stay near the value C and the values F_k "tend" to increase by a constant. The analysis is then performed via a study of stochastic processes, which we describe in detail next.

2.4 Analysis of the stochastic processes

Let us consider the stochastic process $\{\mathcal{A}_k, F_k\}$ generated by Algorithm 2.1 using random, p-probabilistically sufficiently accurate models M_k, with F_k defined above. Under the assumption that the sequence of models M_k are p-probabilistically sufficiently accurate, each iteration is true with probability at least p, conditioned on the past.

We assume now (and we show later for each specific case) that $\{\mathcal{A}_k, F_k\}$ obeys the following rules.

Assumption 2.1 *There exists a constant $C > 0$ and a sequence of functions $h_k(\alpha)$, $\alpha \in \mathbb{R}$, which may or may not be dependent on k and some algorithmic constants and which satisfy $h_k(\alpha) > 0$ for any $\alpha > 0$, such that for any realization of Algorithm 2.1 the following hold.*

(i) If iteration k is true (i.e. event I_k occurs) and successful, then $f_{k+1} \geq f_k + h_k(\alpha_k)$.

(ii) If $\alpha_k \leq C$ and iteration k is true then iteration k is also successful, and hence $\alpha_{k+1} = \gamma^{-1}\alpha_k$ and $f_{k+1} \geq f_k + h_k(\alpha_k)$.

(iii) $f_{k+1} \geq f_k$ for all k.

Without loss of generality, we assume that $C = \gamma^{-c}\alpha_0 < \gamma^{-1}\alpha_{max}$ for some integer c. In other words, C is the largest value that the step size $\mathcal{A}_k$ actually achieves for which part (ii) of Assumption 2.1 holds. The condition $C < \gamma^{-1}\alpha_{max}$ is a simple technical condition, which is not necessary, but which simplifies the presentation later in this section. Under Assumption 2.1, recalling the update rules for α_k in Algorithm 2.1 and the assumption that true iterations occur with probability at least p, we can write the stochastic process $\{\mathcal{A}_k, F_k\}$ as obeying the expressions below.

- If $\mathcal{A}_k \leq C$ then

$$\mathcal{A}_{k+1} \geq \begin{cases} \gamma^{-1}\mathcal{A}_k & \text{if event } I_k \text{ occurs} \\ \gamma\mathcal{A}_k & \text{otherwise} \end{cases} \tag{2}$$

$$F_{k+1} \geq \begin{cases} F_k + h_k(\mathcal{A}_k) & \text{if event } I_k \text{ occurs} \\ F_k & \text{otherwise} \end{cases} \tag{3}$$

- If $\mathcal{A}_k > C$ then

$$\mathcal{A}_{k+1} \geq \begin{cases} \min\{\alpha_{max}, \gamma^{-1}\mathcal{A}_k\} & \text{or} \\ \gamma\mathcal{A}_k & \end{cases} \tag{4}$$

$$F_{k+1} \geq \begin{cases} F_k + h_k(\mathcal{A}_k) & \text{if } \mathcal{A}_{k+1} = \min\{\alpha_{max}, \gamma^{-1}\mathcal{A}_k\} \\ & \text{and if event } I_k \text{ occurs} \\ F_k & \text{otherwise} \end{cases} \tag{5}$$

We conclude that, when $\mathcal{A}_k \leq C$, a successful iteration happens with probability at least p, and in that case $\mathcal{A}_{k+1} = \gamma^{-1}\mathcal{A}_k$, and that an unsuccessful iteration happens with probability at most $1-p$, in which case $\mathcal{A}_{k+1} = \gamma\mathcal{A}_k$. Note that there is no known probability bound for the different outcomes when $\mathcal{A}_k > C$. However, we know that I_k still occurs with probability at least p and if it does, and iteration k happened to be successful, then F_k is increased by at least $h_k(\mathcal{A}_k)$.

Since there are two distinct possible behaviors of our process depending on the size of $\mathcal{A}_k$, we now partition the sequence of iterates into two subsequences: K', such that $\mathcal{A}_k \leq C$ for all $k \in K'$, and K'', such that $\mathcal{A}_k \geq \gamma^{-1}C$ for all $k \in K''$. Observe that $K' \cup K'' = \{1, 2, 3, \ldots, \}$ and $K' \cap K'' = \emptyset$. We will, henceforth, consider two stochastic processes: $\{\mathcal{A}'_k, F'_k\}$, for $\{k = 1, 2, \ldots : \ i_k \in K'\}$ and $\{\mathcal{A}''_k, F''_k\}$, for $\{k = 1, 2, \ldots : \ i_k \in K''\}$. In other words we split the original process $\{\mathcal{A}_k, F_k\}$ into two separate processes, $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$, with indices k simply denoting consecutive steps for each process.

In our analysis for each algorithmic case we derive bounds on the expected hitting times for $\{F'_k \geq F_\epsilon\}$ and for $\{F''_k \geq F_\epsilon\}$ for the processes $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$, respectively. The sum of these bounds will then be an upper bound on the hitting time for $\{F_k \geq \epsilon\}$ of the original process $\{\mathcal{A}_k, F_k\}$.

To derive such bounds we will utilize the following basic result.

Proposition 2.1 *Consider two nondecreasing stochastic processes V_k and W_k, a target value V and hitting times $T_V^{W_k}$ and $T_V^{V_k}$. Suppose that for each realization ω, $V_k(\omega) \leq W_k(\omega)$ for all $k \leq T_V^{W_k(\omega)}$. Then for all ω,*

$$T_V^{W_k(\omega)} \leq T_V^{V_k(\omega)}$$

and, therefore,

$$\mathbb{E}(T_V^{W_k}) \leq \mathbb{E}(T_V^{V_k}).$$

We will consider two auxiliary stochastic processes, which serve as lower bounds on $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$, respectively, and using Proposition 2.1 will help us derive the bounds on $\mathbb{E}(T^{F_K}_{F_\epsilon})$.

Remark 2.1 *Note that, by construction, one of the two processes, $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$, may contain a finite number of steps (since one of the subsequences K' and K'' may be finite). If the number of such steps is smaller than the hitting time of our algorithm, this only implies a smaller bound on N_ϵ, the total number of iterations the entire algorithm takes. Hence, without loss of generality, we can assume that the number of steps in each subsequence is larger than any relevant hitting time that we are seeking to bound.*

Remark 2.2 *Since, by assumption, events I_k in the original algorithmic sequence occur with probability at least p conditioned on the past, which includes the size of $\mathcal{A}_k$, hence the events I_i with i such that $k_i \in K'$ also occur with probability at least p conditioned on the past. Similarly, events I_i with i such that $k_i \in K''$ also occur with probability at least p. This will be used below, when we analyze the behavior of each of the two processes, by constructing two independent, lower-bounding processes on their respective probability spaces.*

2.5 Bounding the stochastic process $\{\mathcal{A}'_k, F'_k\}$

The process $\{\mathcal{A}'_k, F'_k\}$, defined above, behaves as follows

$$\mathcal{A}'_{k+1} \geq \begin{cases} \min\{C, \gamma^{-1}\mathcal{A}_k\} & \text{if event } I'_k \text{ occurs} \\ \gamma\mathcal{A}'_k & \text{otherwise} \end{cases} \tag{6}$$

$$F'_{k+1} \geq \begin{cases} F'_k + h_k(\mathcal{A}'_k) & \text{if event } I'_k \text{ occurs} \\ F'_k & \text{otherwise} \end{cases} \tag{7}$$

Consider the stopping time $T^{F'_k}_{F_\epsilon}$. We now define an auxiliary process on the same probability space as I'_k and $\{\mathcal{A}'_k, F'_k\}$. Let us define a sequence of random events J'_k on the same probability space as the events I'_k and which occur with probability *exactly* p and have the property that if event J'_k occurs then I'_k also occurs (simply put, J'_k occurs on a subset of realizations for which I'_k occurs, whose measure is exactly p). Given the constants $C > 0$, $\gamma \in (0,1)$ and some non-decreasing function $h(z) > 0$, for all $z > 0$, consider an auxiliary stochastic process $\{Z'_k, Y'_k\}$ defined as follows:

$$Z'_{k+1} = \begin{cases} \min\{C, \gamma^{-1}Z'_k\} & \text{if event } J'_k \text{ occurs} \\ \gamma Z'_k & \text{otherwise} \end{cases} \tag{8}$$

$$Y'_{k+1} = \begin{cases} Y'_k + h(Z'_k) & \text{if event } J'_k \text{ occurs} \\ Y'_k & \text{otherwise} \end{cases} \tag{9}$$

Consider any (joint) realization $\omega \in \Omega$ and $\{i_k\} = \{I'_k(\omega)\}$, $\{\alpha'_k, f'_k\} = \{\mathcal{A}'_k(\omega), F'_k(\omega)\}$ and the auxiliary process $\{z'_k, y'_k\} = \{Z'_k(\omega), Y'_k(\omega)\}$. Clearly we have $z'_k \leq \alpha'_k$. Assume now that $h(z'_k) \leq h_k(\alpha'_k)$ for all true and successful iterations $k \leq t^{F'_k}_{F_\epsilon}$ (where $t^{F'_k}_{F_\epsilon} = T^{F'_k}_{F_\epsilon}(\omega)$). Then we have $y'_k \leq f'_k$. This is easy to see, because for all k that correspond to true and successful iterations Y'_k is increased by $h(z'_k)$ and f'_k is increased by at least $h_k(\alpha'_k)$ with $\alpha'_k \geq z'_k$, and on all other iterations f'_k may increase, while y'_k does not. Hence, using Proposition 2.1, we have the following result.

Lemma 2.1 *Conditioned on*

$$h(Z'_k) \leq h_k(\mathcal{A}'_k) \quad \forall k \leq T^{F'_k}_{F_\epsilon} \text{ true and successful,} \tag{10}$$

the hitting time for $\{Y'_k \geq F_\epsilon\}$ *is always greater or equal that the hitting time for* $\{F'_k \geq F_\epsilon\}$ *and, thus,*

$$\mathbb{E}(T^{F'_k}_{F_\epsilon}) \leq \mathbb{E}(T^{Y'_k}_{F_\epsilon}).$$

Analysis of the process $\{Z'_k, Y'_k\}$. We will now bound $\mathbb{E}(T^{Y'_k}_{F_\epsilon})$.

Lemma 2.2 *For the stochastic processes* (8) *and* (9) *and* $T^{Y'_k}_{F_\epsilon}$ *- the hitting time for* $\{Y'_k \geq F_\epsilon\}$, *under the condition that* $p > 1/2$, *we have*

$$\mathbb{E}(T^{Y'_k}_{F_\epsilon}) \leq \frac{p^2 F_\epsilon}{(2p-1)h(C)}.$$

Proof. First let us analyze Z'_k. Note that Z'_k is an ergodic Markov chain, specifically it can be represented by a random walk on nonnegative integers. For all k, Z'_k equals $C\gamma^i$ for some i. Let $i = 0, 1, \ldots$ be the states of the Markov chain, indicating that $Z'_k = C\gamma^i$. The transition probabilities are, trivially, as follows:

$$\{p_{0,0} = p,\ p_{0,1} = 1-p,\ p_{i,i+1} = 1-p,\ p_{i,i-1} = p,\ p_{i,j} = 0, \forall i \geq 1,\ j \neq i+1, i-1\}.$$

We can write down balance equations for the steady state probabilities of the Markov chain:

$$\begin{aligned}(1-p)P_0 &= P_1,\\ (1-p)P_1 + pP_3 &= P_2,\\ (1-p)P_2 + pP_4 &= P_3,\\ &\cdots\end{aligned}$$

where P_i, $i = 0, 1, \ldots$, is the steady state probability of the Markov chain being in state i.

Solving these equations for each $i = 1, 2, \ldots$ we have

$$P_i = [(1-p)/p]^i P_0$$

and using the fact that $\sum_{i=0}^n P_i = 1$ gives

$$P_0 \sum_{i=0}^{\infty} [(1-p)/p]^i = 1,$$

which then implies $P_0 = (2p-1)/p$ for $p > 1/2$ and $P_i = [(1-p)/p]^i (2p-1)/p$. We can now compute the expected number of steps between recurrences of state 0, that is when $Z'_k = C$. This expectation is simply $1/P_0 = p/(2p-1)$. In other words, $Z'_k = C$ on average at least every $1/P_0 = p/(2p-1)$ steps. We also know that whenever $Z'_k = C$, $Y'_{k+1} = Y'_k + h(C)$ with probability p. Hence the process Y'_k increases by at least $h(C)$ every $p^2/(2p-1)$ steps on average and the expected number of steps after which $Y'_k \geq F_\epsilon$ is

$$\mathbb{E}(T^{Y'_k}_{F_\epsilon}) \leq \frac{F_\epsilon p^2}{h(C)(2p-1)}.$$

□

We will use this bound together with Lemma 2.1, for each of the cases of our algorithm. Towards that purpose, for each case, we will derive $h(z)$ for which condition (10) holds.

2.6 Bounding the stochastic process $\{\mathcal{A}''_k, F''_k\}$

Let us now consider the stochastic process restricted to the sequence K''. This process, which we denoted by $\{\mathcal{A}''_k, F''_k\}$, behaves as follows

$$\mathcal{A}''_{k+1} \geq \begin{cases} \min\{\alpha_{max}, \gamma^{-1}\mathcal{A}''_k\} & \text{or} \\ \max\{\gamma^{-1}C, \gamma\mathcal{A}''_k\} \end{cases} \tag{11}$$

$$F''_{k+1} \geq \begin{cases} F''_k + h_k(\mathcal{A}''_k) & \text{if } \mathcal{A}''_{k+1} = \min\{\alpha_{max}, \gamma^{-1}\mathcal{A}''_k\} \\ & \text{and if event } I''_k \text{ occurs} \\ F''_k & \text{otherwise} \end{cases} \tag{12}$$

Here we used the condition $C \leq \gamma\alpha_{max}$ to ensure that $\mathcal{A}''_{k+1} = \min\{\alpha_{max}, \gamma^{-1}\mathcal{A}''_k\}$ indicates a successful iteration. This is done for simplicity of presentation only.

We repeat the argument from the analysis of $\{\mathcal{A}'_k, F'_k\}$ and define an auxiliary process $\{Z''_k, Y''_k\}$ on the same probability space using events J''_k whose probability is *exactly* p and a nondecreasing function $h(z) > 0$ for $z > 0$.

$$Z''_{k+1} = \begin{cases} \min\{\alpha_{max}, \gamma^{-1}Z''_k\} & \text{or} \\ \max\{\gamma^{-1}C, \gamma Z''_k\} \end{cases} \tag{13}$$

$$Y''_{k+1} = \begin{cases} Y''_k + h(Z''_k) & \text{if } Z''_{k+1} = \min\{\alpha_{max}, \gamma^{-1}Z''_k\} \\ & \text{and if event } J''_k \text{ occurs} \\ Y''_k & \text{otherwise} \end{cases} \tag{14}$$

Again, it is easy to see that for each realization, $z''_k \leq \alpha''_k$ for all k. We use the same arguments as for $\{\mathcal{A}'_k, F'_k\}$ to show the following lemma.

Lemma 2.3 *Conditioned on*

$$h(Z''_k) \leq h_k(\mathcal{A}''_k) \quad \forall k \leq T_{F_\epsilon}^{F''_k} \text{ true and successful,} \tag{15}$$

we have

$$\mathbb{E}(T_{F_\epsilon}^{F''_k}) \leq \mathbb{E}(T_{F_\epsilon}^{Y''_k}).$$

The function $h(z)$, which we derive later for each particular case and which satisfies (10), will also satisfy (15).

We now derive the bound for $\mathbb{E}(T_{F_\epsilon}^{Y''_k})$.

Analysis of the process $\{Z''_k, Y''_k\}$. Relating this stochastic process to our algorithmic framework, we note that if $Z''_{k+1} = \min\{\alpha_{max}, Z''_k/\gamma\}$, then iteration k is successful. If, additionally, J''_k occurs, then iteration k is also true. When J''_k does not occur, the iteration may be false and it may be successful or not. Now let us consider our algorithm which induces the process (13)-(14). We will use the terminology of the algorithm while analyzing the process $\{Z''_k, Y''_k\}$, for clearer intuition. Specifically, the following random variables measure the number of certain type of iterations that can happen until $Y''_k \geq F_\epsilon$.

- N_1 is the number of false successful iterations; i.e., the iterations where J_k'' is false and $Z_{k+1}'' = \min\{\alpha_{max}, Z_k''/\gamma\}$.
- M_1 is the number of false iterations; i.e., the iterations where J_k'' is false.
- N_2 is the number of true successful iterations; i.e., the iterations where J_k'' is true and $Z_{k+1}'' = \min\{\alpha_{max}, Z_k''/\gamma\}$.
- M_2 is the number of true iterations; i.e., iterations where J_k'' is true.
- N_3 is the number of true unsuccessful iterations, i.e., iterations where J_k'' is true and $Z_{k+1}'' = \max\{\gamma^{-1}C, \gamma Z_k''\}$.
- M_3 is the number of unsuccessful iterations, i.e, iterations where $Z_{k+1}'' = \max\{\gamma^{-1}C, \gamma Z_k''\}$.

We have the following trivial relations: $N_1 \le M_1$ and $N_2 + N_3 = M_2$, $N_3 \le M_3$.

Our goal is to bound $\mathbb{E}(M_1) + \mathbb{E}(M_2)$ as this is the expected total number of steps until $Y_k'' \ge F_\epsilon$.

We know that $Z_k'' \ge C$ for all k and, so, on all true and successful iterations $Y_{k+1} \ge Y_k + h(C)$. Thus, clearly, we have

$$N_2 \le F_\epsilon / h(C). \tag{16}$$

We have the following simple lemma.

Lemma 2.4

$$\mathbb{E}(M_1) = \frac{1-p}{p}\mathbb{E}(M_2). \tag{17}$$

Proof. Each iteration k for which J_k'' occurs is true with probability p. Hence, the random variable M_2 follows the discrete binomial distribution with probability p, and so we have that

$$\mathbb{E}(M_2) = p\mathbb{E}(M_2 + M_1). \tag{18}$$

Relation (17) now follows from (18). □

We use this lemma and (16) to derive the following result.

Lemma 2.5 *For the stochastic process* (13)-(14) *and* $T_{F_\epsilon}^{Y_k''}$ *- the hitting time for* $\{Y_k'' \ge F_\epsilon\}$, *under the condition that* $p > 1/2$, *we have*

$$\mathbb{E}(T_{F_\epsilon}^{Y_k''}) \le \frac{2F_\epsilon}{h(C)(2p-1)} + \frac{\log_\gamma(\alpha_0/C)}{2p-1}.$$

Proof.

From the trivial relations, observed earlier, we have

$$M_2 = N_2 + N_3 \le N_2 + M_3,$$

and using (17) it follows that

$$\mathbb{E}(N_1) \le \mathbb{E}(M_1) = \frac{1-p}{p}\mathbb{E}(M_2) \le \frac{1-p}{p}\mathbb{E}(N_2 + M_3) = \frac{1-p}{p}[\mathbb{E}(N_2) + \mathbb{E}(M_3)]. \tag{19}$$

Since Z''_k is decreased by the constant factor γ at each unsuccessful iteration but increased only at successful iterations, then we deduce

$$M_3 \leq N_1 + N_2 + \log_\gamma(\alpha_0/C).$$

In other words, the total number of decreases of Z'' is bounded by the total number of increases plus the number of decreases it takes to reduce Z'' from the initial value to C. After this, the iteration limit for K'' is reached, which we assumed to be larger than $M_1 + M_2$. Taking into account the above expressions and using the bound (16) on N_2 we have

$$\mathbb{E}(M_3) \leq \mathbb{E}(N_1) + \mathbb{E}(N_2) + \log_\gamma(\alpha_0/C) \leq \mathbb{E}(N_1) + F_\epsilon/h(C) + \log_\gamma(\alpha_0/C). \tag{20}$$

Plugging this into (19) and using the bound (16) on N_2 again, we obtain

$$\mathbb{E}(N_1) \leq \frac{1-p}{p}\left[\frac{F_\epsilon}{h(C)} + \mathbb{E}(N_1) + \frac{F_\epsilon}{h(C)} + \log_\gamma\left(\frac{\alpha_0}{C}\right)\right],$$

and, hence,

$$\frac{2p-1}{p}\mathbb{E}(N_1) \leq \frac{1-p}{p}\left[2\frac{F_\epsilon}{h(C)} + \log_\gamma\left(\frac{\alpha_0}{C}\right)\right].$$

This finally implies

$$\mathbb{E}(N_1) \leq \frac{1-p}{2p-1}\cdot\frac{2F_\epsilon}{h(C)} + \frac{1-p}{2p-1}\log_\gamma\left(\frac{\alpha_0}{C}\right). \tag{21}$$

Now we can bound the expected total number of iterations until $Y''_k \geq F_\epsilon$, using (16), (20) and (21) and adding the terms to obtain the result of the lemma, namely,

$$\mathbb{E}(M_3 + N_1 + N_2) \leq \frac{2}{2p-1}\cdot\frac{F_\epsilon}{h(C)} + \frac{1}{2p-1}\log_\gamma\left(\frac{\alpha_0}{C}\right).$$

□

Summary of our complexity analysis framework. We have considered a(ny) algorithm in the framework Algorithm 2.1 with probabilistically sufficiently accurate models as in Definition 2.1. We have developed a methodology to obtain (complexity) bounds on the number of iterations N_ϵ that such an algorithm takes to reach desired accuracy. It is important to note that, while we simply provide the bound on $\mathbb{E}(N_\epsilon)$ it is easy to extend the analysis of the same stochastic processes to provide bounds on $P\{N_\epsilon > K\}$, for any K larger than the bound on $\mathbb{E}(N_\epsilon)$, in particular it can be shown that $P\{N_\epsilon > K\}$ decays exponentially with K.

Our approach is valid provided the following hold: the bound (1), Assumption 2.1, and the equations (10) and (15) in Lemmas 2.1 and 2.3, respectively. Next we show that all these conditions are satisfied by steepest-descent linesearch methods in the nonconvex, convex and strongly convex case; by general linesearch methods in the nonconvex case; by cubic regularization methods (ARC) for nonconvex objectives. In particular, we will specify what we mean by a probablistically sufficiently accurate first-order and second-order model in the case of linesearch and cubic regularization methods, respectively.

3 The line-search algorithm

We will now apply the generic analysis outlined in the previous section to the case of the following simple probabilistic line-search algorithm.

Algorithm 3.1 A line-search algorithm with random models

```
Initialization
    Choose constants $\gamma \in (0,1)$, $\theta \in (0,1)$ and $\alpha_{\max} > 0$. Pick initial $x^0$ and $\alpha_0 < \alpha_{max}$.
    Repeat for $k = 0, 1, \ldots$
1. Compute a model and a step
    Compute a random model $m_k$ and use it to generate a direction $g^k$.
    Set the step $s^k = -\alpha_k g^k$.
2. Check sufficient decrease
    Check if
        $f(x^k - \alpha_k g^k) \leq f(x^k) - \alpha_k \theta \|g^k\|^2.$    (22)
3. Successful step
    If (22) holds, then $x^{k+1} := x^k - \alpha_k g^k$ and $\alpha_{k+1} = \min\{\alpha_{max}, \gamma^{-1}\alpha_k\}$. Let $k := k+1$.
4. Unsuccessful step
    Otherwise, $x^{k+1} := x^k$, set $\alpha_{k+1} = \gamma\alpha_k$. Let $k := k+1$.
```

For the linesearch algorithm, the key ingredient is a search direction selection on each iteration. In our case we assume that the search direction is random and satisfies some accuracy requirement that we discuss below. The choice of model in this algorithm is a simple linear model $m_k(x)$, which gives rise to the search direction g^k, specifically, $m_k(x) = f(x^k) + (x - x^k)^T g^k$. We will consider more general models in the next section, Section 3.2.

Recall Definition 2.1. Here we describe the specific requirement we apply to the models in the case of line search.

Definition 3.1 *We say that a sequence of random models and corresponding directions $\{M_k, G_k\}$ is (p)-probabilistically "sufficiently accurate" for Algorithm 3.1 for a corresponding sequence $\{\mathcal{A}_k, X^k\}$, if the events*

$$I_k \ = \ \{\|G^k - \nabla f(X^k)\| \leq \kappa \mathcal{A}_k \|G^k\|\}$$

satisfy the following submartingale-like condition

$$P(I_k | \mathcal{F}^M_{k-1}) \ \geq \ p,$$

where $\mathcal{F}^M_{k-1} = \sigma(M_0, \ldots, M_{k-1})$ is the σ-algebra generated by $M_0, \ldots, M_{k-1}$.

As before, each iteration for which I_k holds is called a true iteration. It follows that for every realization of the algorithm, on all true iterations, we have

$$\|g^k - \nabla f(x^k)\| \leq \kappa \alpha_k \|g^k\|, \tag{23}$$

which implies, using $\alpha_k \leq \alpha_{\max}$ and the triangle inequality, that

$$\|g^k\| \geq \frac{\|\nabla f(x^k)\|}{1+\kappa\alpha_{max}}. \tag{24}$$

For the remainder of the analysis of Algorithm 3.1, we make the following assumption.

Assumption 3.1 *The sequence of random models and corresponding directions $\{M_k, G_k\}$, generated in Algorithm 3.1, is (p)-probabilistically "sufficiently accurate" for the corresponding random sequence $\{\mathcal{A}_k, X^k\}$, with $p > 1/2$.*

We also make a standard assumption on the smoothness of $f(x)$ for the remainder of the paper.

Assumption 3.2 *$f \in \mathcal{C}^1(\mathbb{R}^n)$, is globally bounded below by f_*, and has globally Lipschitz continuous gradient ∇f, namely,*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\| \quad \text{for all } x,\, y \in \mathbb{R}^n \text{ and some } L > 0. \tag{25}$$

3.1 The nonconvex case, steepest descent

As mentioned before, our goal in the non-convex case is to compute a bound on the expected number of iterations k that Algorithm 3.1 requires to obtain an iterate x^k for which $\|\nabla f(x^k)\| \leq \epsilon$. We will now compute the specific quantities and expressions defined in Sections 2.3 and 2.4, that allow us to apply the analysis of our general framework to the specific case of Algorithm 3.1 for non-convex functions.

Let N_ϵ denote, as before, the number of iterations that are taken until $\|\nabla f(X^k)\| \leq \epsilon$ occurs (which is a random variable). Let us consider the stochastic process $\{\mathcal{A}_k, F_k\}$, with $F_k = f(x^0) - f(X^k)$. Since for all k, $F_k \leq F_\epsilon = f(x^0) - f_*$, then (1) holds, namely, the hitting time for $F_k \geq F_\epsilon$, denoted by $T^{F_k}_{F_\epsilon}$, is greater or equal to N_ϵ.

Next we show that Assumption 2.1 is verified. First we derive an expression for the constant C, related to the size of the stepsize α_k.

Lemma 3.1 *Let Assumption 3.2 hold. For every realization of Algorithm 3.1, if iteration k is true (i.e. I_k holds), and if*

$$\alpha_k \leq C = \frac{1-\theta}{0.5L+\kappa}, \tag{26}$$

then (22) *holds. In other words, when* (26) *holds, any true iteration is also a successful one.*

Proof. Condition (25) implies the following overestimation property for all x and s in $\mathbb{R}^n$,

$$f(x+s) \leq f(x) + s^T\nabla f(x) + \frac{L}{2}\|s\|^2,$$

which implies

$$f(x^k - \alpha_k g^k) \quad \leq \quad f(x^k) - \alpha_k (g^k)^T \nabla f(x^k) + \tfrac{L}{2}\alpha_k^2\|g^k\|^2.$$

Applying the Cauchy-Schwarz inequality and (23) we have

$$\begin{aligned} f(x^k - \alpha_k g^k) &\leq f(x^k) - \alpha_k (g^k)^T[\nabla f(x^k) - g^k] - \alpha_k\|g^k\|^2\left[1 - \tfrac{L}{2}\alpha_k\right] \\ &\leq f(x^k) + \alpha_k\|g^k\| \cdot \|\nabla f(x^k) - g^k\| - \alpha_k\|g^k\|^2\left[1 - \tfrac{L}{2}\alpha_k\right] \\ &\leq f(x^k) - \alpha_k\|g^k\|^2\left[1 - \left(\kappa + \tfrac{L}{2}\right)\alpha_k\right]. \end{aligned}$$

It follows that (22) holds whenever $f(x^k) - \alpha_k \|g^k\|^2[1-(\kappa+0.5L)\alpha_k] \leq f(x^k) - \alpha_k\theta\|g^k\|^2$ which is equivalent to (26). □

From Lemma 3.1, and from (22) and (24), for any realization of Algorithm 3.1 which gives us the specific sequence $\{\alpha_k, f_k\}$, the following hold.

- If k is a true and successful iteration, then
$$f_{k+1} \geq f_k + \frac{\theta\|\nabla f(x^k)\|^2\alpha_k}{(1+\kappa\alpha_{max})^2}$$
and
$$\alpha_{k+1} = \gamma^{-1}\alpha_k.$$
- If $\alpha_k \leq C$, where C is defined in (26), and iteration k is true, then it is also successful.

Hence, Assumption 2.1 holds and the process $\{\mathcal{A}_k, F_k\}$ behaves exactly as our generic process (2)-(5) in Section 2.4, with C defined in (26) and the specific choice of $h_k(\mathcal{A}_k) = \frac{\theta\|\nabla f(X^k)\|^2\mathcal{A}_k}{(1+\kappa\alpha_{max})^2}$.

Recall the partitioning of $\{\mathcal{A}_k, F_k\}$ into two processes: $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$. It remains for us to show that there exists a function $h(z) > 0$ for $z > 0$ for which Lemmas 2.1 and 2.3 hold. Let us define the stochastic processes $\{Z'_k, Y'_k\}$ as discussed in Section 2.5 and $\{Z''_k, Y''_k\}$ as discussed in Section 2.6 with $C = \frac{1-\theta}{0.5L+\kappa}$ and $h(z) = \frac{\theta\epsilon^2}{(1+\kappa\alpha_{max})^2}z$. Condition (10) in Lemma 2.1 in our case can be written as

$$\frac{\theta\epsilon^2}{(1+\kappa\alpha_{max})^2}\mathcal{A}'_k \leq \frac{\theta\|\nabla f((X')^k)\|^2\mathcal{A}'_k}{(1+\kappa\alpha_{max})^2} \quad \forall k \leq T_{F_\epsilon}^{Y'_k} \text{ true and successful.} \tag{27}$$

which is equivalent to

$$\|\nabla f((X')^k)\| \geq \epsilon \quad \forall k \leq T_{F_\epsilon}^{Y'_k} \text{ true and successful.}$$

If this condition holds, we have

$$T_{F_\epsilon}^{Y'_k} \geq T_{F_\epsilon}^{F'_k}, \tag{28}$$

and so Lemma 2.1 is satisfied.

We now employ a simple argument: either (27) holds, or the required accuracy of the gradient is reached before $T_{F_\epsilon}^{Y'_k}$. Hence we have the following lemma.

Lemma 3.2 *Let N'_ϵ be the random variable which is the number of steps occurring in the process $\{\mathcal{A}'_k, F'_k\}$ until $\|\nabla f((X')^k)\| \leq \epsilon$ is satisfied. Then*

$$\mathbb{E}(N'_\epsilon) \leq \mathbb{E}(T_{F_\epsilon}^{Y'_k}).$$

Proof. Let us consider a particular realization of Algorithm 3.1 and the corresponding realization of the process $\{Z'_k, Y'_k\}$. For such a realization, we either have $n'_\epsilon \leq t_{F_\epsilon}^{Y'_k}$ or $n'_\epsilon > t_{F_\epsilon}^{Y'_k}$.

Assume that $n'_\epsilon > t_{F_\epsilon}^{Y'_k}$ for this realization. Then for all $k \leq t_{F_\epsilon}^{Y'_k}$, we have $\|\nabla f(x^k)\| \geq \epsilon$. Hence (27) holds, which in turn implies (28). Since $t_{F_\epsilon}^{F'_k} \geq n'_\epsilon$, we conclude that $n'_\epsilon \leq t_{F_\epsilon}^{Y'_k}$. Hence, we conclude that $n'_\epsilon \leq t_{F_\epsilon}^{Y'_k}$ for all realizations, and the lemma holds. □

Similarly, condition (15) in Lemma 2.3 becomes in this case

$$\frac{\theta\epsilon^2}{(1+\kappa\alpha_{max})^2}\mathcal{A}''_k \leq \frac{\theta\|\nabla f((X'')^k)\|^2\mathcal{A}''_k}{(1+\kappa\alpha_{max})^2} \quad \forall k \leq T_{F_\epsilon}^{Y''_k} \text{ true and successful,} \tag{29}$$

and it implies

$$T_{F_\epsilon}^{Y''_k} \geq T_{F_\epsilon}^{F''_k}. \tag{30}$$

Lemma 3.3 *Let N''_ϵ be the random variable which is the number of steps occurring in the process $\{\mathcal{A}''_k, F''_k\}$ until $\|\nabla f((X'')^k)\| \leq \epsilon$ is satisfied. Then*

$$\mathbb{E}(N''_\epsilon) \leq \mathbb{E}(T_{F_\epsilon}^{Y''_k}).$$

Proof. The proof is identical to the proof of Lemma 3.2. □

Since, clearly, $\mathbb{E}(N_\epsilon) \leq \mathbb{E}(N'_\epsilon) + \mathbb{E}(N''_\epsilon)$, we can finally use the bounds in Lemmas 2.2 and 2.5 to bound $\mathbb{E}(N_\epsilon)$.

Theorem 3.1 *Let Assumptions 3.1 and 3.2 hold. Then the expected number of iterations that Algorithm 3.1 takes until $\|\nabla f(X^k)\| \leq \epsilon$ occurs is bounded as follows*

$$\mathbb{E}(N_\epsilon) \leq \frac{M(p^2+2)}{(2p-1)\epsilon^2} + \frac{1}{2p-1}\log\left(\frac{\alpha_0(0.5L+\kappa)}{1-\theta}\right),$$

where $M = \frac{(f(x^0)-f_*)(1+\kappa\alpha_{\max})^2(0.5L+\kappa)}{\theta(1-\theta)}$ is a constant independent of p and ϵ.

Proof. The proof follows from substituting the expressions for C, $h(C)$ and F_ϵ into the bounds in Lemmas 2.2 and 2.5 and applying Lemmas 3.2 and 3.3. □

Remark 3.1 *We note that the dependency of the expected number of iterations on ϵ is of the order $1/\epsilon^2$, as expected from a line-search method applied to a smooth non-convex problem. The dependency on p is rather intuitive as well: if $p = 1$, then the deterministic complexity is recovered, while as p approaches $1/2$, the expected number of iterations goes to infinity, since the models/directions are arbitrarily bad as often as they are good.*

3.2 The nonconvex case, general descent

In this subsection, we explain how the above analysis of the line-search method extends from the non-convex steepest descent case to a general non-convex descent case.

In particular, we consider that in Algorithm 3.1, $s^k = \alpha_k d^k$ (instead of $-\alpha_k g^k$), where d^k is any direction that satisfies the following standard conditions.

- There exists a constant $\beta > 0$, such that

$$\frac{(d^k)^T g^k}{\|d^k\| \cdot \|g^k\|} \leq -\beta, \quad \forall k. \tag{31}$$

- There exist constants $\kappa_1, \kappa_2 > 0$, such that

$$\kappa_1 \|g^k\| \leq \|d^k\| \leq \kappa_2 \|g^k\|, \quad \forall k. \tag{32}$$

The sufficient decrease condition (22) is replaced by

$$f(x^k + \alpha_k d^k) \leq f(x^k) + \alpha_k \theta (d^k)^T g^k. \tag{33}$$

It is easy to show that a simple variant of Lemma 3.1 applies.

Lemma 3.4 *Let Assumption 3.2 hold. Consider Algorithm 3.1 with* $s^k = \alpha_k d^k$ *and sufficient decrease condition* (33)*. Assume that* d^k *satisfies* (31) *and* (32)*. Then, for every realization of the resulting algorithm, if iteration* k *is true (i.e.* I_k *holds), and if*

$$\alpha_k \leq C = \frac{\beta(1-\theta)}{0.5 L \kappa_2 + \kappa}, \tag{34}$$

then (33) *holds. In other words, when* (34) *holds, any true iteration is also a successful one.*

Proof. The first displayed equation in the proof of Lemma 3.1 provides

$$f(x^k + \alpha_k d^k) \quad \leq \quad f(x^k) + \alpha_k (d^k)^T \nabla f(x^k) + \tfrac{L}{2} \alpha_k^2 \|d^k\|^2.$$

Applying the Cauchy-Schwarz inequality, (23) and the conditions (32) on d^k we have

$$\begin{aligned} f(x^k + \alpha_k d^k) &\leq f(x^k) + \alpha_k (d^k)^T [\nabla f(x^k) - g^k] + \alpha_k (d^k)^T g^k + \tfrac{L}{2} \alpha_k^2 \|d^k\|^2 \\ &\leq f(x^k) + \alpha_k \|d^k\| \cdot \|\nabla f(x^k) - g^k\| + \alpha_k (d^k)^T g^k + \tfrac{L}{2} \alpha_k^2 \|d^k\|^2 \\ &\leq f(x^k) + \alpha_k^2 \kappa \|d^k\| \|g^k\| + \alpha_k (d^k)^T g^k + \tfrac{L}{2} \alpha_k^2 \kappa_2 \|d^k\| \|g^k\| \\ &= f(x^k) + \alpha_k (d^k)^T g^k + \alpha_k^2 \|d^k\| \|g^k\| \left(\kappa + \kappa_2 \tfrac{L}{2}\right). \end{aligned}$$

It follows that (33) holds whenever

$$\alpha_k (d^k)^T g^k + \alpha_k^2 \|d^k\| \|g^k\| \left(\kappa + \kappa_2 \frac{L}{2}\right) \leq \alpha_k \theta (d^k)^T g^k,$$

or equivalently, since $\alpha_k > 0$, whenever

$$\alpha_k \|d^k\| \|g^k\| \left(\kappa + \kappa_2 \frac{L}{2}\right) \leq -(1-\theta)(d^k)^T g^k.$$

Using (31), the latter displayed equation holds whenever α_k satisfies (34). □

We conclude this extension to general descent directions by observing that if k is a true and successful iteration, using the sufficient decrease condition (33), the conditions (31) and (32) on d^k and (24), we obtain that

$$f_{k+1} \geq f_k + \frac{\theta \kappa_1 \beta \|\nabla f(x^k)\|^2 \alpha_k}{(1 + \kappa \alpha_{max})^2}.$$

Hence, Assumption 2.1 holds for this case as well and the remainder of the analysis is exactly the same as for the steepest descent case.

3.3 The convex case

We now analyze the expected complexity of Algorithm 3.1 in the case when $f(x)$ is a convex function, that is when the following assumption holds.

Assumption 3.3 *$f \in \mathcal{C}^1(\mathbb{R}^n)$ is convex and has bounded level sets so that*

$$\|x - x^*\| \leq D \quad \text{for all } x \text{ with } f(x) \leq f(x^0), \tag{35}$$

where x^* is a global minimizer of f. Let $f^* = f(x^*)$.

In this case, our goal is to bound the expectation of N_ϵ - the number of iterations taken by Algorithm 3.1 until

$$f(X^k) - f^* \leq \epsilon \tag{36}$$

occurs. We denote $f(X^k) - f^*$ by Δ_k and define $F_k = \frac{1}{\Delta_k}$. Clearly, N_ϵ is also the number of iterations taken until $F_k \geq \frac{1}{\epsilon} = F_\epsilon$ occurs, and so (1) trivially holds.

Regarding Assumption 2.1, Lemma 3.1 provides the value for the constant C, namely, that whenever $\mathcal{A}_k \leq C$ with $C = \frac{1-\theta}{0.5L+\kappa}$, then every true iteration is also successful. We now show that on true and successful iterations, F_k is increased by at least some function value $h_k(\mathcal{A}_k)$.

Lemma 3.5 *Let Assumptions 3.2 and 3.3 hold. Consider any realization of Algorithm 3.1. For every iteration k that is true and successful, we have*

$$f_{k+1} \geq f_k + \frac{\theta \alpha_k}{D^2(1+\kappa\alpha_{\max})^2}. \tag{37}$$

Proof. Note that convexity of f implies that for all x and y,

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

and so by using $x = x^*$ and $y = x^k$, we have

$$-\Delta_k = f(x^*) - f(x^k) \geq \nabla f(x^k)^T(x^* - x^k) \geq -D\|\nabla f(x^k)\|,$$

where to obtain the last inequality, we used Cauchy-Schwarz inequality and (35). Thus when k is a true iteration, (24) further provides

$$\frac{1}{D}\Delta_k \leq \|\nabla f(x^k)\| \leq (1+\kappa\alpha_{\max})\|g^k\|.$$

When k is also successful,

$$f(x^k) - f(x^{k+1}) = \Delta_k - \Delta_{k+1} \geq \theta\alpha_k\|g^k\|^2 \geq \frac{\theta\alpha_k}{D^2(1+\kappa\alpha_{\max})^2}\Delta_k^2.$$

Dividing the above expression by $\Delta_k\Delta_{k+1}$, we have that on all true and successful iterations

$$\frac{1}{\Delta_{k+1}} - \frac{1}{\Delta_k} \geq \frac{\theta\alpha_k}{D^2(1+\kappa\alpha_{\max})^2}\frac{\Delta_k}{\Delta_{k+1}} \geq \frac{\theta\alpha_k}{D^2(1+\kappa\alpha_{\max})^2},$$

since $\Delta_k \geq \Delta_{k+1}$. Recalling the definition of f_k completes the proof. □

Similarly to the non-convex case, we conclude from Lemmas 3.1 and 3.5, that for any realization of Algorithm 3.1 the following have to happen.

- If k is a true and successful iteration, then

$$f_{k+1} \geq f_k + \frac{\theta \alpha_k}{D^2(1+\kappa\alpha_{\max})^2}$$

and

$$\alpha_{k+1} = \gamma^{-1}\alpha_k.$$

- If $\alpha_k \leq C$, where C is defined in (26), and iteration k is true, then it is also successful.

Hence, Assumption 2.1 holds and the process $\{\mathcal{A}_k, F_k\}$ behaves exactly as our generic process (2)-(5) in Section 2.4, with C defined in (26) and the specific choice of $h_k(\mathcal{A}_k) = \frac{\theta \mathcal{A}_k}{D^2(1+\kappa\alpha_{\max})^2}$.

As before, we partition $\{\mathcal{A}_k, F_k\}$ into two processes: $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$, and define the stochastic processes $\{Z'_k, Y'_k\}$ as discussed in Section 2.5 and $\{Z''_k, Y''_k\}$, as in Section 2.6 with $C = \frac{1-\theta}{0.5L+\kappa}$, $F_\epsilon = \frac{1}{\epsilon}$ and $h(z) = \frac{\theta z}{D^2(1+\kappa\alpha_{\max})^2}$. Since $h(z) \equiv h_k(z)$ for all k, conditions (10) and (15) always hold. Thus Lemmas 2.1 and 2.3 are satisfied with $N'_\epsilon := T^{F'_k}_{F_\epsilon}$ and $N''_\epsilon := T^{F''_k}_{F_\epsilon}$ (the latter definitions arise from those of N_ϵ and F_k in this convex case). Lemmas 2.2 and 2.5 can be immediately applied, yielding the following complexity bound, together with $\mathbb{E}(N_\epsilon) \leq \mathbb{E}(N'_\epsilon) + \mathbb{E}(N''_\epsilon)$ and the above expressions for C, $h(C)$ and F_ϵ.

Theorem 3.2 *Let Assumptions 3.1, 3.2 and 3.3 hold. Then the expected number of iterations that Algorithm 3.1 takes until $f(X^k) - f^* \leq \epsilon$ occurs is bounded by*

$$\mathbb{E}(N_\epsilon) \leq \frac{(p^2+2)M}{(2p-1)\epsilon} + \frac{1}{2p-1}\log\left(\frac{\alpha_0(0.5L+\kappa)}{1-\theta}\right),$$

where $M = \frac{(1+\kappa\alpha_{\max})^2 D^2(0.5L+\kappa)}{\theta(1-\theta)}$ is a constant independent of p and ϵ.

Remark 3.2 *We again note the same dependence on ϵ in the complexity bound in Theorem 3.2 as in the deterministic convex case and on p, as in the non-convex case.*

3.4 The strongly convex case

We now consider the case of strongly convex objective functions, hence the following assumption holds.

Assumption 3.4 *$f \in \mathcal{C}^1(\mathbb{R}^n)$ is strongly convex, namely, for all x and y and some $\mu > 0$,*

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

Recall our notation $\Delta_k = f(X^k) - f^*$. Our goal here is again, as in the convex case, to bound the expectation of the number of iteration that occur until $\Delta_k \leq \epsilon$. In the strongly convex case, however, this bound is logarithmic in $\frac{1}{\epsilon}$, just as it is in the case of the deterministic algorithm.

Lemma 3.6 *Let Assumption 3.4 hold. Consider any realization of Algorithm 3.1. For every iteration k that is true and successful, we have*

$$f(x^k) - f(x^{k+1}) = \Delta_k - \Delta_{k+1} \geq \frac{2\mu\theta}{(1+\kappa\alpha_{\max})^2}\alpha_k\Delta_k, \tag{38}$$

or equivalently,

$$\Delta_{k+1} \leq \left(1 - \frac{2\mu\theta}{(1+\kappa\alpha_{\max})^2}\alpha_k\right)\Delta_k. \tag{39}$$

Proof. Assumption 3.4 implies, for $x = x^k$ and $y = x^*$, that [see [14], Th 2.1.10]

$$\Delta_k \leq \frac{1}{2\mu}\|\nabla f(x^k)\|^2$$

or equivalently,

$$\sqrt{2\mu\Delta_k} \leq \|\nabla f(x^k)\| \leq (1 + \kappa\alpha_{\max})\|g^k\|,$$

where in the second inequality we used (24). The bound (38) now follows from the sufficient decrease condition (22). □

Note that for (38) to imply linear rate of decrease in Δ_k on true and successful iterations, we must have $\alpha_k \leq (1 + \kappa\alpha_{\max})^2/(2\mu\theta)$. The latter inequality can be achieved by an appropriate choice of the parameters. For example, if we set $\mu \in (0, 1]$ and $\theta \in (0, 1]$ then, since $\alpha_k \leq \alpha_{\max}$ for all k and $\kappa > 1$ (w.l.o.g.), we have that $\alpha_k \leq (1 + \kappa\alpha_{\max})^2/(2\mu\theta)$ and hence

$$1 - \frac{2\mu\theta\alpha_k}{(1 + \kappa\alpha_{\max})^2} \in (0, 1). \tag{40}$$

We now define $F_k = \log \frac{1}{\Delta_k}$ and $F_\epsilon = \log \frac{1}{\epsilon}$, and note that the hitting time $T_{F_\epsilon}^{F_k}$ is always equal to N_ϵ - the number of iterations taken until $\Delta_k \leq \epsilon$. Hence, (1) trivially holds, and as before, we seek to bound $\mathbb{E}(T_{F_\epsilon}^{F_k})$.

As in the convex case, using Lemmas 3.1 and 3.6, we conclude that, for any realization of Algorithm 3.1, the following have to happen.

- If k is a true and successful iteration, then

$$f_{k+1} \geq f_k - \log\left(1 - \frac{2\mu\theta}{(1 + \kappa\alpha_{\max})^2}\alpha_k\right).$$

  and

$$\alpha_{k+1} = \gamma^{-1}\alpha_k.$$

- If $\alpha_k \leq C$, where C defined in (26), and iteration k is true, then it is also successful.

Hence, again, Assumption 2.1 holds and the process $\{\mathcal{A}_k, F_k\}$ behaves exactly as our generic process (2)-(5) in Section 2.4, with C defined in (26) and the specific choice of

$$h_k(\mathcal{A}_k) = -\log\left(1 - \frac{2\mu\theta}{(1 + \kappa\alpha_{\max})^2}\mathcal{A}_k\right).$$

We proceed again as in the convex case, partitioning $\{\mathcal{A}_k, F_k\}$ into two processes: $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$, defining N'_ϵ and N''_ϵ appropriately (the same as $T_{F_\epsilon}^{F'_k}$ and $T_{F_\epsilon}^{F''_k}$, respectively) and defining the stochastic processes $\{Z'_k, Y'_k\}$ as discussed in Section 2.5 and $\{Z''_k, Y''_k\}$, as in Section 2.6 with $C = \frac{1-\theta}{0.5L+\kappa}$, $F_\epsilon = \log\left(\frac{1}{\epsilon}\right)$ and $h(z) = -\log\left(1 - \frac{2\mu\theta}{(1+\kappa\alpha_{\max})^2}z\right)$. Here again $h(z) \equiv h_k(z)$ for all k, and, hence, conditions (10) and (15) always hold and Lemmas 2.2 and 2.5 can be immediately applied.

By using the above expressions for C, $h(C)$ and F_ϵ, we have the following complexity bound for the strongly convex case.

Theorem 3.3 *Let Assumptions 3.1, 3.2 and 3.4 hold. Then the expected number of iterations that Algorithm 3.1 takes until $f(X^k) - f^* \leq \epsilon$ occurs is bounded by*

$$\mathbb{E}(N_\epsilon) \leq \frac{(2+p^2)M}{(2p-1)} \log\left(\frac{1}{\epsilon}\right) + \frac{1}{2p-1} \log\left(\frac{\alpha_0(0.5L+\kappa)}{1-\theta}\right),$$

where $M = -\log\left(1 - \frac{2\mu\theta(1-\theta)}{(1+\kappa\alpha_{\max})^2(0.5L+\kappa)}\right)$ is a constant independent of p and ϵ.

Remark 3.3 *Again, note the same dependence of the complexity bound in Theorem 3.3 on ϵ as for the deterministic line-search algorithm, and the same dependence on p as for the other problem classes discussed above.*

4 Probabilistic second-order models and cubic regularization methods

In this section we consider a randomized version of second-order methods, whose deterministic counterpart achieves optimal complexity rate [7, 4]. As in the line-search case, we show that in expectation, the same rate of convergence applies as in the deterministic (cubic regularization) case, augmented by a term that depends on the probability of having accurate models. Here we revert back to considering general objective functions that are not necessarily convex.

4.1 A cubic regularization algorithm with random models

Let us now consider a cubic regularization method where the following model

$$m_k(x^k + s) = f(x^k) + s^T g^k + \frac{1}{2} s^T b^k s + \frac{\sigma_k}{3}\|s\|^3, \tag{41}$$

is approximately minimized on each iteration k with respect to s, for some vector g^k and a matrix b^k and some regularization parameter $\sigma^k > 0$. As before we assume that g_k and b_k are realizations of some random variables G_k and B_k, which imply that the model is random and we assume that it is sufficiently accurate with probability at least p; the details of this assumption will be given after we state the algorithm.

The step s^k is computed as in [6, 7] to approximately minimize the model (41), namely, it is required to satisfy

$$(s^k)^T g^k + (s^k)^T b^k s^k + \sigma_k \|s^k\|^3 = 0 \text{ and } (s^k)^T b^k s^k + \sigma_k \|s^k\|^3 \geq 0 \tag{42}$$

and

$$\|\nabla m_k(x^k + s^k)\| \leq \kappa_\theta \min\{1, \|s^k\|\} \|g^k\|, \tag{43}$$

where $\kappa_\theta \in (0, 1)$ is a user-chosen constant.

Note that (42) is satisfied if s^k is the global minimizer of the model m_k over some subspace; in fact, it is sufficient for s^k to be the global minimizer of m_k along the line αs^k [7]. Condition (43) is a relative termination condition for the model minimization (say over increasing subspaces) and it is clearly satisfied at stationary points of the model; ideally it will be satisfied sooner at least in the early iterations of the algorithm [7].

The probabilistic Adaptive Regularization with Cubics (ARC) framework is presented below.

Algorithm 4.1 An ARC algorithm with random models

Initialization

Choose parameters $\gamma \in (0,1)$, $\theta \in (0,1)$, $\sigma_{\min} > 0$ *and* $\kappa_\theta \in (0,1)$. *Pick initial* x^0 *and* $\sigma_0 > \sigma_{\min}$. *Repeat for* $k = 0, 1, \ldots,$

1. Compute a model

Compute an approximate gradient g^k *and Hessian* b^k *and form the model* (41).

2. Compute the trial step s^k

Compute the trial step s^k *to satisfy* (42) *and* (43).

3. Check sufficient decrease

Compute $f(x^k + s^k)$ *and*

$$\rho_k = \frac{f(x^k) - f(x^k + s^k)}{f(x^k) - m_k(x^k + s^k)}. \tag{44}$$

4. Update the iterate

Set

$$x^{k+1} = \begin{cases} x^k + s^k & \text{if} \quad \rho_k \geq \theta \qquad [k \text{ successful}] \\ x^k & \text{otherwise} \qquad\quad [k \text{ unsuccessful}] \end{cases} \tag{45}$$

5. Update the regularization parameter σ_k

Set

$$\sigma_{k+1} = \begin{cases} \max\{\gamma\sigma_k, \sigma_{\min}\} & \text{if} \quad \rho_k \geq \theta \\ \frac{1}{\gamma}\sigma_k & \text{otherwise.} \end{cases} \tag{46}$$

Remark 4.1 *Typically (see e.g. [6]) one would further refine* (45) *and* (46) *by distinguishing between successful and very successful iterations, when* ρ_k *is not just positive but close to* 1. *It is beneficial in the deterministic setting to keep the regularization parameter unchanged on successful iterations when* ρ_k *is greater than* θ *but is not close to* 1 *and only to decrease it when* ρ_k *is substantially larger than* θ. *For simplicity and uniformity of our general framework, we simplified the parameter update rule. However, the analysis presented here can be quite easily extended to the more general case by slightly extending the flexibility of the stochastic processes. In practice it is yet unclear if the same strategy will be beneficial, as "accidentally" bad models and the resulting unsuccessful steps may drive the parameter* σ_k *to be larger than it should be, and hence a more aggressive decrease of* σ_k *may be desired. This practical study is a subject of future research.*

Remark 4.2 *We have stated Algorithm 4.1 so that it is as close as possible to known/deterministic ARC frameworks for ease of reading. We note however, that it is perfectly coherent with the generic algorithmic framework, Algorithm 2.1, if one sets* $\alpha_k = 1/\sigma_k$ *and* $\rho_k \geq \theta$ *as the sufficient decrease condition. We will exploit this connection in the analysis that follows.*

The requirement of sufficient model accuracy considered here is similar to the definition of probabilistically fully quadratic models introduced in [1], though note that we only require the second-order condition along the trial step s^k.

Definition 4.1 *We say that a sequence of random models and corresponding directions* $\{M_k\}$ *is* (p)*-probabilistically "sufficiently accurate" for Algorithm 4.1 (below) if there exist constants* κ_g *and* κ_H *such that for any corresponding random sequence* $\{\mathcal{A}_k = 1/\Sigma_k, X^k\}$*, the events*

$$I_k \; = \; \{\|\nabla f(X^k) - G^k\| \le \kappa_g \|S^k\|^2 \quad \text{and} \quad \|(H(X^k) - B^k)S^k\| \le \kappa_H \|S^k\|^2\}$$

satisfy the following submartingale-like condition

$$P(I_k | F^M_{k-1}) \; \ge \; p,$$

where $F^M_{k-1} = \sigma(M_0, \dots, M_{k-1})$ *is the* σ*-algebra generated by* $M_0, \dots, M_{k-1}$*.*

As before, for any realization of Algorithm 4.1, we refer to iterations k when I_k occurs as *true* iterations, and otherwise, as *false* iterations. Hence for all true iterations k,

$$\|\nabla f(x^k) - g^k\| \le \kappa_g \|s^k\|^2 \quad \text{and} \quad \|(H(x^k) - b^k)s^k\| \le \kappa_H \|s^k\|^2. \tag{47}$$

For the remainder of the analysis of Algorithm 4.1 we make the following assumption.

Assumption 4.1 *The sequence of random models and corresponding directions* $\{M_k, S_k\}$*, generated in Algorithm 3.1, is* (p)*-probabilistically "sufficiently accurate" for the corresponding random sequence* $\{\mathcal{A}_k = 1/\Sigma_k, X^k\}$*, with* $p > 1/2$*.*

Regarding the possibly nonconvex objective f, in addition to Assumption 3.2, we also need the following assumption.

Assumption 4.2 $f \in \mathcal{C}^2(\mathbb{R}^n)$ *and has globally Lipschitz continuous Hessian* H*, namely,*

$$\|H(x) - H(y)\| \le L_H \|x - y\| \quad \text{for all } x,\, y \in \mathbb{R}^n \text{ and some } L_H > 0. \tag{48}$$

4.2 Global convergence rate analysis, nonconvex case

The next four lemmas give useful properties of Algorithm 4.1 that are needed later for our stochastic analysis.

Lemma 4.1 (Lemma 3.3 in [6]) *Consider any realization of Algorithm 4.1. Then on each iteration* k *we have*

$$f(x^k) - m_k(x^k + s^k) \ge \frac{1}{6}\sigma_k \|s^k\|^3. \tag{49}$$

Thus on every successful iteration k*, we have*

$$f(x^k) - f(x^{k+1}) \ge \frac{\theta}{6}\sigma_k \|s^k\|^3. \tag{50}$$

Proof. Clearly, (50) follows from (49) and the sufficient decrease condition (45). It remains to prove (49). Combining the first condition on step s^k in (42), with the model expression (41) for $s = s^k$ we can write

$$f(x^k) - m_k(x^k + s^k) = \frac{1}{2}(s^k)^T B^k s^k + \frac{2}{3}\sigma_k \|s^k\|^3.$$

The second condition on s^k in (42) implies $(s^k)^T B^k s^k \ge -\sigma_k \|s^k\|^3$, which, when used with the above equation, gives (49). □

Lemma 4.2 *Let Assumptions 3.2 and 4.2 hold. For any realization of Algorithm 4.1, if iteration k is true (i.e., I_k occurs), and if*

$$\sigma_k \geq \sigma_c = \frac{2\kappa_g + \kappa_H + L + L_H}{1 - \frac{1}{3}\theta}, \tag{51}$$

then iteration k is also successful.

Proof. Clearly, if $\rho_k - 1 \geq 0$, then k is successful by definition. Let us consider the case when $\rho_k < 1$; then if $1 - \rho_k \leq 1 - \theta$, k is successful. We have from (44), that

$$1 - \rho_k = \frac{f(x^k + s^k) - m_k(x^k + s^k)}{f(x^k) - m_k(x^k + s^k)}.$$

Taylor expansion and triangle inequalities give, for some $\xi^k \in [x^k, x^k + s^k]$,

$$\begin{aligned}
&f(x^k + s^k) - m_k(x^k + s^k) \\
&= [\nabla f(x^k) - g^k]^T s^k + \tfrac{1}{2}(s^k)^T[H(\xi^k) - H(x^k)]s^k + \tfrac{1}{2}(s^k)^T[H(x^k) - b^k]s^k - \tfrac{1}{3}\sigma_k\|s^k\|^3 \\
&\leq \|\nabla f(x^k) - g^k\| \cdot \|s^k\| + \tfrac{1}{2}\|H(\xi^k) - H(x^k)\| \cdot \|s^k\|^2 + \tfrac{1}{2}\|(H(x^k) - b^k)s^k\| \cdot \|s^k\| - \tfrac{1}{3}\sigma_k\|s^k\|^3 \\
&\leq \left(\kappa_g + \tfrac{L_H}{2} + \tfrac{\kappa_H}{2} - \tfrac{1}{3}\sigma_k\right)\|s^k\|^3 = (6\kappa_g + 3L_H + 3\kappa_H - 2\sigma_k)\tfrac{1}{6}\|s^k\|^3,
\end{aligned}$$

where the last inequality follows from the fact that the iteration is true and hence (47) holds, and from Assumption 4.2. This and (49) now give that $1 - \rho_k \leq 1 - \theta$ when σ_k satisfies (51). □

Lemma 4.3 *Let Assumptions 3.2 and 4.2 hold. Consider any realization of Algorithm 4.1. On each true iteration k we have*

$$\|s^k\| \geq \sqrt{\frac{1 - \kappa_\theta}{\sigma_k + \kappa_s}\|\nabla f(x^k + s^k)\|}, \tag{52}$$

where $\kappa_s = 2\kappa_g + \kappa_H + L + L_H$.

Proof. Triangle inequality, equality $\nabla m_k(x^k + s) = g^k + b^k s + \sigma_k\|s\|s$ and condition (43) on s^k together give

$$\begin{aligned}
\|\nabla f(x^k + s^k)\| &\leq \|\nabla f(x^k + s^k) - \nabla m_k(x^k + s^k)\| + \|\nabla m_k(x^k + s^k)\| \\
&\leq \|\nabla f(x^k + s^k) - g^k - b^k s^k\| + \sigma_k\|s^k\|^2 + \kappa_\theta \min\{1, \|s^k\|\}\|g^k\|.
\end{aligned} \tag{53}$$

Recalling Taylor expansion of $\nabla f(x^k)$

$$\nabla f(x^k + s^k) = \nabla f(x^k) + \int_0^1 H(x^k + ts^k)s^k dt,$$

and applying triangle inequality, again, we have

$$\begin{aligned}
\|\nabla f(x^k + s^k) - g^k - b^k s^k\| &\leq \|\nabla f(x^k) - g^k\| + \\
&\quad \left\|\int_0^1 [H(x^k + ts^k) - H(x^k)]s^k dt\right\| + \|H(x^k)s^k - b^k s^k\| \\
&\leq \left\{\kappa_g + \tfrac{1}{2}L_H + \kappa_H\right\}\|s^k\|^2,
\end{aligned}$$

where to get the second inequality, we also used (47) and Assumption 4.2.

We can bound $\|g^k\|$ as follows

$$\|g^k\| \leq \|g^k - \nabla f(x^k)\| + \|\nabla f(x^k) - \nabla f(x^k + s^k)\| + \|\nabla f(x^k + s^k)\| \leq (\kappa_g + L)\|s^k\| + \|\nabla f(x^k + s^k)\|.$$

Thus finally, we can bound all the terms on the right hand side of (53) in terms of $\|s^k\|^2$ and using the fact that $\kappa_\theta \in (0, 1)$ we can write

$$(1 - \kappa_\theta)\|\nabla f(x^k + s^k)\| \leq (2\kappa_g + \kappa_H + L + L_H + \sigma_k)\|s^k\|^2,$$

which is equivalent to (52). □

Lemma 4.4 *Let Assumptions 3.2 and 4.2 hold. Consider any realization of Algorithm 4.1. On each true and successful iteration k, we have*

$$f(x^k) - f(x^{k+1}) \geq \frac{\kappa_f}{(\max\{\sigma_k, \sigma_c\})^{3/2}}\|\nabla f(x^{k+1})\|^{3/2}, \tag{54}$$

where $\kappa_f := \frac{\theta}{12\sqrt{2}}(1 - \kappa_\theta)^{3/2}\sigma_{\min}$ and σ_c is defined in (51).

Proof. Combining Lemma 4.3, inequality (50) from Lemma 4.1 and the definition of successful iteration in Algorithm 4.1 we have, for all true and successful iterations k,

$$f(x^k) - f(x^{k+1}) \geq \frac{\theta}{6}(1 - \kappa_\theta)^{3/2}\frac{\sigma_k}{(\sigma_k + \kappa_s)^{3/2}}\|\nabla f(x^{k+1})\|^{3/2}. \tag{55}$$

Using that $\sigma_k \geq \sigma_{\min}$ and that $\kappa_s \leq \sigma_c$, (55) implies (54). □

The stochastic processes and global convergence rate analysis We are now ready to cast Algorithm 4.1 and its behavior into the generic stochastic analysis framework of Section 2.

For each realization of Algorithm 4.1, we define

$$\alpha_k = \frac{1}{\sigma_k} \quad \text{and} \quad f_k = f(x^0) - f(x^k),$$

and consider the corresponding stochastic process $\{\mathcal{A}_k = 1/\Sigma_k, F_k = f(X^0) - f(X^k)\}$. Let $F_\epsilon = f(x^0) - f^*$ denote the upper bound on the progress measure F_k.

As in the case of the line-search algorithm applied to nonconvex objectives, we would like to bound the expected number of iterations that Algorithm 4.1 takes until $\|\nabla f(X^k)\| \leq \epsilon$ occurs. As usual, let N_ϵ denote this number of iterations (which is a random variable). Since for all k, $f(X^k) \geq f^*$, then the hitting time for $F_k \geq F_\epsilon$, denoted by $T^{F_k}_{F_\epsilon}$, is greater or equal to N_ϵ, and so (1) trivially holds.

Regarding Assumption 2.1, Lemmas 4.2 and 4.4 provide that the following must hold for any realization of Algorithm 4.1.

- If k is a true and successful iteration, then

$$f_{k+1} \geq f_k + \frac{\kappa_f}{(\max\{\sigma_k, \sigma_c\})^{3/2}}\|\nabla f(x^{k+1})\|^{3/2}$$

and

$$\alpha_{k+1} = \gamma^{-1}\alpha_k.$$

- If $\alpha_k \leq C = \frac{1}{\sigma_c}$, where σ_c is defined in (51), and iteration k is true, then it is also successful.

Hence, once again, Assumption 2.1 holds and the process $\{\mathcal{A}_k, F_k\}$ behaves exactly as our generic process (2)-(5) in Section 2.4, with $C = \frac{1}{\sigma_c} = \frac{1-\frac{1}{3}\theta}{2\kappa_g+\kappa_H+L+L_H}$, and the specific choice

$$h_k(\mathcal{A}_k) = \kappa_f(\min\{\mathcal{A}_k, C\})^{3/2}\|\nabla f(X^k + S^k)\|^{3/2}.$$

Here the dependence of h_k on X^k is more complex that in the line-search case as it also depends on the step S_k. Since we are only interested in the value of $h_k(\mathcal{A}_k)$ on true and successful iterations, we know that $X^{k+1} = X^k + S^k$ for all such iterations. In fact, we are only interested in bounding $h_k(\mathcal{A}_k)$ from below, for all k such that $\|\nabla f(X^k + S^k)\| \geq \epsilon$.

Recall, also, the partitioning of $\{\mathcal{A}_k, F_k\}$ into two processes: $\{\mathcal{A}'_k, F'_k\}$ and $\{\mathcal{A}''_k, F''_k\}$. Let us define the stochastic processes $\{Z'_k, Y'_k\}$ as discussed in Section 2.5 and $\{Z''_k, Y''_k\}$, as in Section 2.6 with $C = \frac{1}{\sigma_c}$, where σ_c is defined in (51), and $h(z) = \kappa_f(\min\{z, C\})^{3/2}\epsilon^{3/2}$. Condition (10) in Lemma 2.1 in this case can be written as

$$\begin{aligned}\kappa_f(\min\{\mathcal{A}'_k, C\})^{3/2}\epsilon^{3/2} \leq\ & \kappa_f(\min\{\mathcal{A}'_k, C\})^{3/2}\|\nabla f((X')^k + (S')^k)\|^{3/2} \\ & \forall k \leq T^{F'_k}_{F_\epsilon},\ k \text{ is true and successful,}\end{aligned} \tag{56}$$

which is equivalent to

$$\|\nabla f((X')^k + (S')^k)\| \geq \epsilon \quad \forall k \leq T^{F'_k}_{F_\epsilon},\ k \text{ is true and successful.}$$

If this condition holds, then we have

$$T^{Y'_k}_{F_\epsilon} \geq T^{F'_k}_{F_\epsilon}, \tag{57}$$

and so Lemma 2.1 holds.

As in the nonconvex, line-search case, we employ the same simple argument: either (56) holds, or the required accuracy of the gradient is reached before $T^{Y'_k}_{F_\epsilon}$. Hence we have the following lemma.

Lemma 4.5 *Let N'_ϵ be the random variable which is the number of steps occurring in the process $\{\mathcal{A}'_k, F'_k\}$ until $\|\nabla f((X')^k)\| \leq \epsilon$ is satisfied. Then*

$$\mathbb{E}(N'_\epsilon) \leq \mathbb{E}(T^{Y'_k}_{F_\epsilon}) + 1.$$

Proof. Now let us consider a particular realization of our algorithm and the corresponding realization of the process $\{Z'_k, Y'_k\}$. For such a realization, we either have $n'_\epsilon \leq t^{Y'_k}_{F_\epsilon} + 1$ or $n_\epsilon > t^{Y'_k}_{F_\epsilon} + 1$.

Assume that $n'_\epsilon > t^{Y'_k}_{F_\epsilon} + 1$ for a particular realization. Then for all $k \leq t^{Y'_k}_{F_\epsilon}$, such that k is successful, we have $\|\nabla f(x^k + s^k)\| \geq \epsilon$. This clearly follows from our assumption and from the fact that $x^{k+1} = x^k + s^k$ on all successful iterations. Hence, (56) holds, which in turn implies (57). Since $t^{F'_k}_{F_\epsilon} \geq n'_\epsilon$, we have $n'_\epsilon \leq t^{Y'_k}_{F_\epsilon}$. Hence we conclude that $n'_\epsilon \leq t^{Y'_k}_{F_\epsilon} + 1$, for all realizations and the lemma holds. □

Similarly, recalling condition (15) in Lemma 2.3, we have the following result.

Lemma 4.6 *Let N''_ϵ be the random variable which is the number of steps occurring in the process $\{\mathcal{A}''_k, F''_k\}$ until $\|\nabla f((X'')^k)\| \leq \epsilon$ is satisfied. Then*

$$\mathbb{E}(N''_\epsilon) \leq \mathbb{E}(T^{Y''_k}_{F_\epsilon}) + 1.$$

Proof. The proof is identical to the proof of Lemma 4.5. □

Since $\mathbb{E}(N_\epsilon) \leq \mathbb{E}(N'_\epsilon) + \mathbb{E}(N''_\epsilon)$, we can finally use the bound from Lemmas 2.2 and 2.5 to bound $\mathbb{E}(N_\epsilon)$.

Theorem 4.1 *Let Assumptions 3.2, 4.1 and 4.2 hold. Then the expected number of iterations that Algorithm 4.1 takes until $\|\nabla f(X^k)\| \leq \epsilon$ occurs is bounded by*

$$\mathbb{E}(N_\epsilon) \leq \frac{(2+p^2)M}{(2p-1)\epsilon^{3/2}} + \frac{1}{2p-1}\log\left(\frac{2\kappa_g + \kappa_H + L + L_H}{\sigma_0(1-1/3\theta)}\right),$$

where $M = \frac{(f(x^0)-f^*)(2\kappa_g+\kappa_H+L+L_H)^{3/2}}{\kappa_f(1-1/3\theta)^{3/2}}$ is a constant independent of p and ϵ.

Proof. The proof follows from substituting the expressions for C, $h(C)$ and F_ϵ into the bounds in Lemmas 2.2 and 2.5, and applying Lemmas 4.5 and 4.6. □

Remark 4.3 *We note that the dependency on ϵ in the above bound on the expected number of iterations is of the order $\epsilon^{-3/2}$, which is of the same order as for the deterministic ARC algorithm and is the optimal rate for non-convex optimization using second order models [4]. The dependence on p is, again, the same as in the case of line-search and it is intuitive.*

5 Random models

In this section we will discuss and motivate the definition of probabilistically "sufficiently accurate" models. In particular, Definition 3.1 is a modification of the definition of probabilistically fully-linear models, which is used in [1]. Similarly, Definition 4.1 is similar to that of probabilistically fully-quadratic models in [1]. These definitions serve to provide properties of the model (with some probability) which are sufficient for first-order (in the case of Definition 3.1) and second-order (in the case of Definition 4.1) convergence rates.

We will now describe several setting where the models are random and satisfy our definitions.

5.1 Stochastic gradients and batch sampling

In [3] an adaptive sample size strategy was proposed in the setting where $\nabla f(x) = \sum_{i=1}^N \nabla f_i(x)$, for large values of N. In this case computing $\nabla f(x)$ accurately can be prohibitive, hence, instead an estimate $\nabla f_S(x) = \sum_{i\in S}\nabla f_i(x)$ is often computed in hopes that it provides a good estimate of the gradient and a descent direction. It is observed in [3] that if sample sets S_k on each iteration ensure that

$$\|\nabla f_{S_k}(x^k) - \nabla f(x^k)\| \leq \mu\|\nabla f_{S_k}(x^k)\| \tag{58}$$

for some $\mu \in (0,1)$, then using a fixed step size

$$\alpha_k \equiv \alpha \leq \frac{1-\mu}{L} \tag{59}$$

the step $s_k = -\alpha \nabla f_{S_k}(x)$ is always a descent step and the line search algorithm converges with the rate $O(\log(1/\epsilon))$ if f is strongly convex. Clearly, condition (58) implies that the model $M_k(x) = f(x^k) + \nabla f_{S_k}(x^k)^\top (x - x^k)$ is sufficiently accurate according to Definition 3.1 for the given fixed step size α. Hence Assumption 3.1 on the models can be viewed as a relaxed version of those in [3], since we allow the condition (58) to fail, as long as it fails with probability less than 1/2, conditioned on the past. Moreover, we analyze the practical version of line search algorithm, with a variable step size, which does not have to remain smaller than $\frac{1-\mu}{L}$ and we provide convergence rates in convex, strongly convex and non convex setting.

Convergence in expectation of a stochastic algorithm is further shown in [3]. In particular, under the assumption that the variance of $\|\nabla f_i(x)\|$ is bounded for all i and that $\mathbb{E}_S[\nabla f_S(x^k)] = \nabla f(x^k)$, it is shown that, for X^k computed after k steps of stochastic gradient descent with a fixed step size, $\mathbb{E}[f(X^k)]$ converges linearly of f^*, when $f(x)$ is strongly convex and if $|S_k|$ - the size of the sample set S_k - grows exponentially with k.

Here, again, our results can be viewed as a generalization of the results in [3]. Indeed, let us assume that $\mathbb{E}_S[\nabla f_S(x^k)] = \nabla f(x^k)$ for each x^k and let $t_k = |S_k|$ - the size of the sample set S_k. Since variance of $\|\nabla f_i(x)\|$ is bounded for all i, have that that $\mathbb{E}_{S_k}[\|\nabla f_{S_k}(x^k) - \nabla f(x^k)]\| \leq \frac{w}{t_k}$, for some fixed w, where the expectation is taken over all random sample sets S_k of size t_k. In other words, the variance of one sample of the stochastic gradient $\|\nabla f_i(x)\|$ is bounded and hence the variable $\nabla f_{S_k}(x^k)$ decays proportionally to the size of S_k.

By Chebychev inequality

$$Pr\{\|\nabla f_{S_k}(x^k) - \nabla f(x^k)\| > \min\{1/2, \alpha_k\}\|\nabla f(x^k)\|\} \leq \frac{w}{\min\{1/2, \alpha_k\}^2 \|\nabla f(x^k)^2\| |S_k|}.$$

If $\|\nabla f_{S_k}(x^k) - \nabla f(x^k)\| \leq \min\{1/2, \alpha_k\}\|\nabla f(x^k)\|$, for a particular x_k and a sample set S_k, then by applying triangle inequality we have

$$\|\nabla f_{S_k}(x^k) - \nabla f(x^k)\| \leq \frac{\alpha_k \|\nabla f_{S_k}(x^k)\|}{2}.$$

Hence the probability of the event $\|\nabla f_S(x^k) - \nabla f(x^k)\| \leq \frac{\alpha_k \|\nabla f_{S_k}(x^k)\|}{2}$ is at least

$$1 - \frac{w}{\min\{1/2, \alpha_k\}^2 \|\nabla f(x^k)\|^2 |S_k|} \geq 1 - \frac{w}{\min\{1/2, \alpha_k\}^2 (1+\alpha_k)^2 \|\nabla f_{S_k}(x^k)\|^2 |S_k|},$$

hence as long as $|S_k|$ is chosen sufficiently large, then this probability is greater than 1/2 and $\nabla f_S(x^k)$ provides us with a probabilistically sufficiently accurate model according to Definition 3.1. Hence the theory described in this paper applies to the case of line search based on stochastic gradient. Note that, on top of the results in [3], we not only analyze line search in non convex and convex setting, but also show convergence with probability one, not simply in expectation, and show the bound on the expected number of iterations, rather than the expected accuracy.

Analyzing complexity of methods in this setting in terms of the total number of gradient samples is a subject of some current research [17]. We leave the exact comparison that can be obtained from our results and those existing in current literature as future research, as this requires defining a sample size selection strategy and possible improvement of our results. Similarly, we leave for future research the derivations of the models in this setting that satisfy Definition 4.1 for the use within the ARC algorithm.

5.2 Models based on random sampling of function values

The motivation behind the notions of probabilistically fully-linear and fully-quadratic models introduced in [1] is based on derivative-free models, which are models based on function values, rather than gradient estimates. We will now show how such models fit into our framework.

Let us first recall the definition of probabilistically fully-linear and quadratic models and pose it in the terms closest to the ones used in this paper

Definition 5.1 *1. We say that a sequence of random models $\{M_k\}$ is (p)-probabilistically fully-linear if there exists constant κ_g such that for any corresponding random sequence Δ_k, X^k, the events*

$$I_k^l \;=\; \{\|\nabla f(X^k) - G^k\| \leq \kappa_g \Delta_k\}$$

satisfy the following submartingale-like condition

$$P(I_k^l | F_{k-1}^M) \;\geq\; p,$$

where $F_{k-1}^M = \sigma(M_0, \ldots, M_{k-1})$ is the σ-algebra generated by $M_0, \ldots, M_{k-1}$.

2. We call sequence $\{M_k\}$ is (p)-probabilistically fully-quadratic if there exist constants κ_g and κ_H such that for any corresponding random sequence Δ_k, X^k, the events

$$I_k^q \;=\; \{\|\nabla f(X^k) - G^k\| \leq \kappa_g \Delta_k^2 \quad \text{and} \quad \|H(X^k) - B^k\| \leq \kappa_H \Delta_k\}$$

satisfy the following submartingale-like condition

$$P(I_k^q | F_{k-1}^M) \;\geq\; p,$$

where $F_{k-1}^M = \sigma(M_0, \ldots, M_{k-1})$ is the σ-algebra generated by $M_0, \ldots, M_{k-1}$.

The key difference between the conditions in Definition 5.1 and those in Definitions 3.1 and 4.1 is the right hand side of the error bounds - in the case of fully-linear and fully-quadratic models Δ_k is a random variable that does not depend on M_k, but in the case of this paper, Δ_k is replaced by $\mathcal{A}_k\|G_k\|$ in the case of Definition 3.1 and by $\|S_k\|$ in the case of Definition 3.1. In other words, the accuracy of the model has to be proportional to the step size which this model produces. Since in [1] trust region methods are analyzed instead of line search and ARC, Definition 5.1 is sufficient.

Models in [1] are constructed by sampling function values in a ball of a given radius around the current iterate x^k and in all cases construction of the k-th model M_k relies on the knowledge of the sampling radius. We will now show that, given a mechanism of constructing probabilistically fully-linear and fully-quadratic models for any sequence of radii (as described in [1]), we can modify our line search algorithm and ARC algorithm, respectively, and extend the convergence rate analysis to utilize these models.

Line-search with probabilistically fully-linear models Let us consider Algorithm 3.1 and corresponding random sequence of iterates X^k and step sizes $\mathcal{A}_k$. If a given model M_k is fully-linear in $B(X^k, \mathcal{A}_k\Xi_k)$ and $\|G_k\| \geq \kappa_\Delta \Xi_k$, for some positive constant κ_Δ, then model M_k is sufficiently accurate, according to Definitions 3.1.

To achieve this, for instance, in nonconvex case, for all $\|\nabla f(X^k)\| \geq \epsilon$ consider $\Xi_k \leq \frac{\epsilon}{2\kappa_g \max\{\mathcal{A}_k, 1\}}$, where κ_g is the constant in the definition of fully-linear models. Then any fully-linear model $M_k(x)$ is also sufficiently accurate, simply because $\|\nabla f(X^k) - G^k\| \leq \kappa_g \mathcal{A}_k \Xi_k \leq \min\{\mathcal{A}_k, 1\}\frac{\epsilon}{2}$ implies $\|G_k\| \geq \frac{\epsilon}{2} \geq \Xi_k$. Similar bounds can be derived for the convex and strongly convex cases.

Consider the following example of a method that produces probabilistically sufficiently accurate models, based on the arguments above. Suppose we are estimating gradients of $f(x)$ by a finite difference scheme using step size $\mathcal{A}_k\Xi_k$, with Ξ_k, sufficiently small, and suppose we compute the function values using parallel computations. If some of the computations fail to complete (due to an overloaded processor, say) with some probability and the total probability of having a computational failure in any of the processors at each iteration is less than $1/2$, conditioned on the past, then we obtain probabilistically sufficiently accurate models. Note that, we do not assume the nature of the computational error, when such error occurs, hence allowing for the gradient estimate to be, occasionally, completely inaccurate.

Another example can be derived from [1], where it is shown that sparse gradient and Hessian estimates can be obtained by randomly sampling fewer function values than is needed to construct gradient and/or Hessian by finite differences. Using this sampling strategy, probabilistically fully-linear and fully-quadratic models can be generated at reduced computation cost. Here again, choosing sampling radius to equal $\Delta_k = \mathcal{A}_k\Xi_k$, with sufficiently small Ξ_k will guarantee that the models are also probabilistically sufficiently accurate.

We now address a more practical approach, when estimates Ξ_k are not chosen to be small enough a priory, but are dynamically decreased, as another parameter in the algorithm. We will outline how our theory can be extended in this case. Consider the following modification of Algorithm 3.1.

Algorithm 5.1 Line-search with probabilistically fully-linear models

Initialization

Chose constants $\theta \in (0,1)$, $\gamma \in (0,1)$, $\alpha_{\max} > 0$ and $\kappa_\Delta > 1$. Pick initial x^0 and $\alpha_0 < \alpha_{max}$, ξ_0. Repeat for $k = 0, 1, \ldots$

1. **Compute a model**
   Compute a model m_k, which is probabilistically fully-linear in $B(x^k, \alpha_k\xi_k)$ and use it to generate a direction g^k.

2. **Check model accuracy**
   If $\|g^k\| \geq \kappa_\Delta\xi_k$, then set the step $s^k = -\alpha_k g^k$ and continue to Step 3.
   Otherwise, $x^{k+1} = x^k$, $\alpha_{k+1} = \alpha_k$, $\xi_{k+1} = \xi_k/\kappa_\Delta$, return to Step 1.

3. **Check sufficient decrease**
   Check if
   $$f(x^k - \alpha_k g^k) \leq f(x^k) - \alpha_k\theta\|g^k\|^2. \tag{60}$$

4. **Successful step**
   If (60) holds, then $x^{k+1} := x^k - \alpha_k g^k$ and $\alpha_{k+1} = \min\{\alpha_k/\gamma, \alpha_{max}\}$.

5. Unsuccessful step

Otherwise, $x^{k+1} = x^k$.
$\alpha_{k+1} = \gamma\alpha_k$.

In the above algorithm, at each iteration we maintain ξ_k, which is expected to be an underestimate of the norm of the descent direction, up to a constant, κ_Δ. The algorithm then uses $\delta_k = \alpha_k\xi_k$ as the radius for constructing fully-linear models. After the model is produced, condition $\|g_k\| \geq \kappa_\Delta\xi_k$ is checked. If this condition holds, the algorithm proceeds exactly as the original version, but if this condition fails, then ξ_k is reduced by a constant (κ_Δ is a practical choice, but any other constant can be used) and the iteration is declared to be unsuccessful (hence $x^{k+1} = x^k$), and the step size α_k remains the same.

Let us consider different possible outcomes for each iteration k for which $\|\nabla f(x^k)\| \geq \epsilon$. From our analysis above, we know that if $\xi_k \leq \frac{\epsilon}{2\kappa_g\alpha_{max}}$ and the model m_k is fully linear, then $\|g_k\| \geq \xi_k$, hence the model is also sufficiently accurate and the iteration of Algorithm 5.1 proceeds as in Algorithm 3.1. Since ξ_k is never increased, then, once it is small enough, the analysis of Algorithm 5.1 can be reduced to that of Algorithm 3.1. Then what remains is to estimate the number of iterations that Algorithm 5.1 takes until $\xi_k \leq \frac{\epsilon}{2\kappa_g\alpha_{max}}$ or $\|\nabla f(x^k)\| \leq \epsilon$ occurs.

While ξ_k is not sufficiently small, we can have the following outcomes: 1) $\|g_k\| < \kappa_\Delta\xi_k$, in which case ξ_k is reduced, 2) the model is not fully linear and $\|g_k\| \geq \kappa_\Delta\xi_k$, hence the model may not be sufficiently accurate, but ξ_k is not reduced and 3) the model is fully-linear and $\|g_k\| \geq \kappa_\Delta\xi_k$, hence the model is also sufficiently accurate. Hence with probability at least p, ξ_k is reduced or the model is sufficiently accurate. It is possible to extend the definition of our stochastic processes and their analysis to compute the upper bounds on the expected number of iterations Algorithm 5.1 takes until $\|\nabla f(x^k)\| \leq \epsilon$ occurs. This bound will be increased by adding a constant times the number of iterations it takes to achieve $\xi_k \leq \frac{\epsilon}{2\kappa_g\alpha_{max}}$, which is $O(\log(1/\epsilon))$. Again, similar analysis can be carried out for the cases of convex and strongly convex functions.

ARC with probabilistically fully-quadratic models Let us consider Algorithm 4.1. In this case, in the same vein with line-search, we consider setting Δ_k in the Definition 5.1 of probabilistically fully-quadratic models, to a sufficiently small value or adjusting it in the run of the algorithm so as to ensure that when the model is fully-quadratic, it is also sufficiently accurate (at least asymptotically). We will make these two approaches to the choice of Δ_k more precise in what follows. To this end, we need a new variant of Lemma 4.3 for the case of probabilistically fully-quadratic models.

Lemma 5.1 *Let Assumptions 3.2 and 4.2 hold. Consider any realization of Algorithm 4.1 where we generate models that are p-probabilistically fully-quadratic according to Definition 5.1. Then on each iteration k in which I_k^q occurs, we have*

$$(1-\kappa_\theta)\|\nabla f(x^k+s^k)\| \leq (2\kappa_g+\kappa_H)\delta_k \max\{\delta_k, 1\} + (L+L_H+\sigma_k)\|s^k\|^2. \tag{61}$$

In particular, if $\epsilon \in (0,1]$, $\max\{L, L_H\} \geq 1$, *and*

$$\delta_k \leq \frac{(1-\kappa_\theta)\epsilon}{\max\{2(2\kappa_g+\kappa_H), L+L_H+\sigma_k\}}, \tag{62}$$

then on each iteration k with $\|\nabla f(x^k+s^k)\| \geq \epsilon$ and in which I_k^q occurs, we have $\|s^k\| \geq \delta_k$.

Assume now that $\delta_k = \frac{\xi_k}{\sigma_k}$. *Then if* $\epsilon \in (0,1]$, $\max\{L, L_H\} \geq 1$, *and*

$$\xi_k \leq \frac{(1-\kappa_\theta)\sigma_{\min}\epsilon}{\max\{2(2\kappa_g+\kappa_H), L+L_H+\sigma_{\min}\}} := \xi_\epsilon, \tag{63}$$

then on each iteration k *with* $\|\nabla f(x^k+s^k)\| \geq \epsilon$ *and in which* I_k^q *occurs, we have* $\|s^k\| \geq \delta_k$.

Proof. It follows from Definition 5.1 that on each realization of Algorithm 4.1, we have

$$\|\nabla f(x^k) - g^k\| \leq \kappa_g \delta_k^2 \quad \text{and} \quad \|H(x^k) - b^k\| \leq \kappa_H \delta_k \tag{64}$$

The proof of (61) now follows identically to the proof of Lemma 4.3 if one uses (64) instead of (47).

The choice of δ_k in (62) implies $\delta_k \leq 1$ and so $\|s^k\| \geq \delta_k$ trivially holds when $\|s^k\| \geq 1$. When $\|s^k\| < 1$, $\|\nabla f(x^k+s^k)\| \geq \epsilon$, and $\delta_k \leq 1$, (61) implies

$$(1-\kappa_\theta)\epsilon - (2\kappa_g+\kappa_H)\delta_k \leq (L+L_H+\sigma_k)\|s^k\|.$$

Now the condition (62) on δ_k implies $(L+L_H+\sigma_k)\|s^k\| \geq (1-\kappa_\theta)\epsilon/[2(2\kappa_g+\kappa_H)]$. Applying again the upper bound on δ_k provides $\|s^k\| \geq \delta_k$.

Finally, if $\delta_k = \frac{\xi_k}{\sigma_k}$, and using $\sigma^k \geq \sigma_{\min}$ for all k due to the algorithm construction, (63) implies (62). □

The second part of Lemma 5.1 provides that if p-probabilistically fully-quadratic models are generated with δ_k chosen sufficiently small so that (62) holds, then the models are also p-probabilistically sufficiently accurate. Thus Algorithm 4.1 can be run with models sampled in this way and the analysis carries through as before. For example, as in the case of linesearch, g^k and b^k could be generated by (sufficiently accurate) finite-difference schemes using function values, where computations are done in parallel and where the total probability of computational failure in any of the processors at each iteration is less than 1/2.

Note however, that the bound that dictates the choice of a suitably small δ_k depends on problem constants that may not be known a priori. Thus it would be better – and computationally more efficient – to adjust δ_k during the run of Algorithm 4.1. A modification of Algorithm 4.1 that allows this is given next, and can be viewed as the analogue for ARC of the line-search Algorithm 5.1.

Algorithm 5.2 ARC with probabilistically fully-quadratic models

Initialization

Choose parameters $\sigma_{\min} > 0$, $\gamma \in (0,1)$, $\theta \in (0,1)$, $0 < \kappa_\theta < 1$ *and* $\kappa_\Delta > 1$. *Pick a starting point* x^0, *a starting value* $\sigma_0 > \sigma_{\min}$ *and* $\xi_0 > 0$. *Repeat for* $k = 0, 1, \ldots,$

1. Compute a model

Compute a model which is probabilistically fully-quadratic in $B\left(x^k, \frac{\xi_k}{\sigma_k}\right)$, *and hence generate approximate gradient* g^k *and Hessian* b^k.

2. Compute the trial step s^k

Compute the trial step s^k *to satisfy* (42) *and* (43).

3. Check model accuracy

If $\|s^k\| \geq \kappa_\Delta \delta_k := \xi_k/\sigma_k$, then go to Step 4.
Otherwise, set $x^{k+1} = x^k$, $\sigma_{k+1} = \sigma_k$, $\xi_{k+1} = \xi_k/\kappa_\Delta$, and return to Step 1.

4. Check sufficient decrease

Compute $f(x^k + s^k)$ and

$$\rho_k = \frac{f(x^k) - f(x^k + s^k)}{f(x^k) - m_k(x^k + s^k)}.$$

5. Update the iterate

Set

$$x^{k+1} = \begin{cases} x^k + s^k & \text{if} \quad \rho_k \geq \theta \qquad [k \text{ successful}] \\ x^k \quad \text{otherwise} & \qquad\qquad [k \text{ unsuccessful}] \end{cases}$$

6. Update the regularization parameter σ_k

Set

$$\sigma_{k+1} = \begin{cases} \max\{\gamma\sigma_k, \sigma_{\min}\} & \text{if} \quad \rho_k \geq \theta \\ \frac{1}{\gamma}\sigma_k \quad \text{otherwise.} \end{cases}$$

Algorithm 5.2 updates ξ_k in order to obtain an underestimate $\delta_k := \xi_k/\sigma_k$ on the length of the step s^k. It constructs probabilistically fully-quadratic models in $B(x^k, \xi_k/\sigma_k)$ and checks whether $\|s^k\| \geq \kappa_\Delta \delta_k$. If that is the case, then the iteration of the above algorithm proceeds as (Algorithm 4.1) before; note that then, if the model is fully quadratic then it is also sufficiently accurate. Otherwise, if the step is too short, then ξ_k is decreased by κ_Δ, x^k and σ^k remain unchanged and a new model is generated (within the smaller ball).

Let us consider the behaviour of Algorithm 5.2 while $\|\nabla f(x^k + s^k)\| \geq \epsilon$. It follows from the last part of Lemma 5.1 that since ξ_ϵ is independent of k and ξ_k is never increased in the algorithm, if $\kappa_\Delta \xi_j \leq \xi_\epsilon$ for some j, then ξ_k will remain below this threshold for all subsequent iterations $k \geq j$; from this j onwards, whenever the model is fully quadratic, then $\|s^k\| \geq \kappa_\Delta \delta_k$ and the model is also sufficiently accurate. Thus from iteration j onwards, Algorithm 5.2 reduces to Algorithm 4.1 and the complexity analysis is the same as before. It remains to estimate the size of j, namely, the number of iterations Algorithm 5.2 takes until $\xi_k \leq \xi_\epsilon$ or $\|\nabla f(x^k+s^k)\| < \epsilon$.

Similarly to the linesearch analysis of possible outcomes above, we can argue that while ξ_k is not sufficiently small, at least with probability p, ξ_k is reduced or the model is sufficiently accurate. Thus, extending our earlier ARC analysis (and definitions of stochastic processes, etc) to account for the ξ_k updates as well, we would find that the complexity bound for Algorithm 5.2 is essentially that of Algorithm 4.1 plus a $\mathcal{O}(\log(1/\epsilon))$ term (coming from $\log(\xi_0/\xi_\epsilon)\log\kappa_\Delta$) that accounts for the number of iterations to drive ξ_k below ξ_ϵ.

6 Conclusions

We have proposed a general algorithmic framework with random models and a methodology for analyzing its complexity that relies on bounding the hitting time of a nondecreasing stochastic process that measures progress towards optimality. Our framework accounts for linesearch and cubic regularization methods, for example, and we particularize our results to obtain precise complexity bounds in the case of nonconvex and convex functions. Despite allowing our models to be arbitrarily inaccurate sometimes, the bounds we obtained match their deterministic counterparts in the order of the accuracy ϵ. The effect of model inaccuracy is reflected by the constant multiple of the bound, which is a function of the probability that the model is sufficiently accurate. We have also briefly discussed ways to obtain probabilistically sufficiently accurate models as required by our framework.

The results in the paper assume that the objective f is deterministic. Obtaining global rates of convergence results for similar algorithmic frameworks when f is stochastic is a topic of future research. Also, further exploring ways to efficiently generate probabilistically sufficiently accurate models may increase the applicability of our results to a diverse set of problems.

Acknowledgements We would like to thank Alexander Stolyar for helpful discussions on stochastic processes.

References

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

[2] R. Byrd, J. Nocedal, and F. Oztoprak, *An inexact successive quadratic approximation method for convex l-1 regularized optimization*, tech. rep., 2013.

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

[4] C. Cartis, N. Gould, and P. L. Toint, *Optimal Newton-type methods for nonconvex smooth optimization problems*, Tech. Rep. Optimization Online, 2011.

[5] ——, *On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization*, SIAM Journal on Optimization, 22 (2012), pp. 66–86.

[6] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results*, Math. Program., 127 (2011), pp. 245–295.

[7] ——, *Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity*, Math. Program., 130 (2011), pp. 295–319.

[8] R. Chen, *Stochastic Derivative-Free Optimization of Noisy Functions*, PhD thesis, Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, USA, 2015.

[9] R. Chen, M. Menickelly, and K. Scheinberg, *Stochastic optimization using a trust-region method and random models*, tech. rep., ISE Dept., Lehigh University.

[10] O. Devolder, F. Glineur, and Y. Nesterov, *First-order methods of smooth convex optimization with inexact oracle*, Math. Program., 146 (2014), pp. 37–75.

[11] S. Ghadimi and G. Lan, *Stochastic first- and zeroth-order methods for nonconvex stochastic programming*, SIAM Journal on Optimization, 23 (2013), pp. 2341–2368.

[12] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang, *Direct search based on probabilistic descent*, Tech. Rep. 14-11, Dept. Mathematics, Univ. Coimbra, 2014.

[13] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal newton-type methods for convex optimization*, in NIPS, 2012.

[14] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[15] Y. Nesterov, *Random gradient-free minimization of convex functions*, Tech. Rep. 2011/1, CORE, 2011.

[16] Y. Nesterov and B. T. Polyak, *Cubic regularization of newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.

[17] R. Pasupathy, P. W. Glynn, S. Ghosh, and F. Hahemi, *How much to sample in simulation-based stochastic recursions?*, (2014). Under Review.

[18] H. Robbins and S. Monro, *A stochastic approximation method*, Annals of Mathematical Statistics, 22 (1951), pp. 400–407.

[19] M. W. Schmidt, N. L. Roux, and F. Bach, *Convergence rates of inexact proximal-gradient methods for convex optimization*, in NIPS, 2011, pp. 1458–1466.

[20] ——, *Minimizing finite sums with the stochastic average gradient*, CoRR, abs/1309.2388 (2013).

[21] J. Spall, *Multivariate stochastic approximation using a simultaneous perturbation gradient approximation*, IEEE Transactions on Automatic Control, 37 (1992), pp. 332–341.