Randomized Derivative-Free Optimization of Noisy Convex Functions[*]

Ruobing Chen[†]   Stefan M. Wild[‡]

July 14, 2015

Abstract

We propose STARS, a randomized derivative-free algorithm for unconstrained optimization when the function evaluations are contaminated with random noise. STARS takes dynamic, noise-adjusted smoothing stepsizes that minimize the least-squares error between the true directional derivative of a noisy function and its finite difference approximation. We provide a convergence rate analysis of STARS for solving convex problems with additive or multiplicative noise. Experimental results show that (1) STARS exhibits noise-invariant behavior with respect to different levels of stochastic noise; (2) the practical performance of STARS in terms of solution accuracy and convergence rate is significantly better than that indicated by the theoretical result; and (3) STARS outperforms a selection of randomized zero-order methods on both additive- and multiplicative-noisy functions.

1 Introduction

We propose STARS, a randomized derivative-free algorithm for unconstrained optimization when the function evaluations are contaminated with random noise. Formally, we address the stochastic optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) = \mathbb{E}_\xi \left[\tilde{f}(x; \xi) \right], \tag{1.1}$$

where the objective $f(x)$ is assumed to be differentiable but is available only through noisy realizations $\tilde{f}(x; \xi)$. In particular, although our analysis will at times assume that the gradient of the objective function $f(x)$ exist and be Lipschitz continuous, we assume that direct evaluation of these derivatives is impossible. Of special interest to this work are situations when derivatives are unavailable or unreliable because of stochastic noise in the objective function evaluations. This type of noise introduces the dependence on the random variable ξ in (1.1) and may arise if random fluctuations or measurement errors occur in a simulation producing the objective f. In addition to stochastic and Monte Carlo simulations, this stochastic noise can also be used to model the variations in iterative or adaptive simulations resulting from finite-precision calculations and specification of internal tolerances [14].

[†]Data Mining Services and Solutions, Bosch Research and Technology Center, Palo Alto, CA 94304.

[‡]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439.

Various methods have been designed for optimizing problems with noisy function evaluations. One such class of methods, dating back half a century, are *randomized search methods* [11]. Unlike classical, deterministic direct search methods [1, 2, 4, 10, 20, 21], randomized search methods attempt to accelerate the optimization by using random vectors as search directions. These randomized schemes share a simple basic framework, allow fast initialization, and have shown promise for solving large-scale derivative-free problems [7, 19]. Furthermore, optimization folklore and intuition suggest that these randomized steps should make the methods less sensitive to modeling errors and "noise" in the general sense; we will systematically revisit such intuition in our computational experiments.

Recent works have addressed the special cases of zero-order minimization of convex functions with additive noise. For instance, Agarwahl et al. [3] utilize a bandit feedback model, but the regret bound depends on a term of order n^{16}. Recht et al. [17] consider a coordinate descent approach combined with an approximate line search that is robust to noise, but only theoretical bounds are provided. Moreover, the situation where the noise is nonstationary (for example, varying relative to the objective function) remains largely unstudied.

Our approach is inspired by the recent work of Nesterov [15], which established complexity bounds for convergence of random derivative-free methods for convex and nonconvex functions. Such methods work by iteratively moving along directions sampled from a normal distribution surrounding the current position. The conclusions are true for both the smooth and nonsmooth Lipschitz-continuous cases. Different improvements of these random search ideas appear in the latest literature. For instance, Stich et al. [19] give convergence rates for an algorithm where the search directions are uniformly distributed random vectors in a hypersphere and the stepsizes are determined by a line-search procedure. Incorporating the Gaussian smoothing technique of Nesterov [15], Ghadimi and Lan [7] present a randomized derivative-free method for stochastic optimization and show that the iteration complexity of their algorithm improves Nesterov's result by a factor of order n in the smooth, convex case. Although complexity bounds are readily available for these randomized algorithms, the practical usefulness of these algorithms and their potential for dealing with noisy functions have been relatively unexplored.

In this paper, we address ways in which a randomized method can benefit from careful choices of noise-adjusted smoothing stepsizes. We propose a new algorithm, STARS, short for STepsize Approximation in Random Search. The choice of stepsize work is greatly motivated by Moré and Wild's recent work on estimating computational noise [12] and derivatives of noisy simulations [13]. STARS takes dynamically changing smoothing stepsizes that minimize the least-squares error between the true directional derivative of a noisy function and its finite-difference approximation. We provide a convergence rate analysis of STARS for solving convex problems with both additive and multiplicative stochastic noise. With nonrestrictive assumptions about the noise, STARS enjoys a convergence rate for noisy convex functions identical to that of Nesterov's random search method for smooth convex functions.

The second contribution of our work is a numerical study of STARS. Our experimental

results illustrate that (1) the performance of STARS exhibits little variability with respect to different levels of stochastic noise; (2) the practical performance of STARS in terms of solution accuracy and convergence rate is often significantly better than that indicated by the worst-case, theoretical bounds; and (3) STARS outperforms a selection of randomized zero-order methods on both additive- and multiplicative-noise problems.

The remainder of this paper is organized as follows. In Section 2 we review basic assumptions about the noisy function setting and results on Gaussian smoothing. Section 3 presents the new STARS algorithm. In Sections 4 and 5, a convergence rate analysis is provided for solving convex problems with additive noise and multiplicative noise, respectively. Section 6 presents an empirical study of STARS on popular test problems by examining the performance relative to both the theoretical bounds and other randomized derivative-free solvers.

2 Randomized Optimization Method Preliminaries

One of the earliest randomized algorithms for the nonlinear, deterministic optimization problem

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

where the objective function f is assumed to be differentiable but evaluations of the gradient ∇f are not employed by the algorithm, is attributed to Matyas [11]. Matyas introduced a *random optimization approach* that, at every iteration k, randomly samples a point x_+ from a Gaussian distribution centered on the current point x_k. The function is evaluated at $x_+ = x_k + u_k$, and the iterate is updated depending on whether decrease has been seen:

$$x_{k+1} = \begin{cases} x_+ & \text{if } f(x_+) < f(x_k) \\ x_k & \text{otherwise.} \end{cases}$$

Polyak [16] improved this scheme by describing stepsize rules for iterates of the form

$$x_{k+1} = x_k - h_k \frac{f(x_k + \mu_k u_k) - f(x_k)}{\mu_k} u_k, \tag{2.2}$$

where $h_k > 0$ is the stepsize, $\mu_k > 0$ is called the smoothing stepsize, and $u_k \in \mathbb{R}^n$ is a random direction.

Recently, Nesterov [15] has revived interest in Poljak-like schemes by showing that Gaussian directions $u \in \mathbb{R}^n$ allow one to benefit from properties of a Gaussian-smoothed version of the function f,

$$f_\mu(x) = \mathbb{E}_u[f(x + \mu u)], \tag{2.3}$$

where $\mu > 0$ is again the smoothing stepsize and where we have made explicit that the expectation is being taken with respect to the random vector u.

Before proceeding, we review additional notation and results concerning Gaussian smoothing.

2.1 Notation

We say that a function $f \in \mathcal{C}^{0,0}(\mathbb{R}^n)$ if $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuous and there exists a constant L_0 such that

$$|f(x) - f(y)| \le L_0 \|x - y\|, \quad \forall x, y \in \mathbb{R}^n,$$

where $\| \cdot \|$ denotes the Euclidean norm. We say that $f \in \mathcal{C}^{1,1}(\mathbb{R}^n)$ if $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable and there exists a constant L_1 such that

$$\|\nabla f(x) - \nabla f(y)\| \le L_1 \|x - y\| \quad \forall x, y \in \mathbb{R}^n. \tag{2.4}$$

Equation (2.4) is equivalent to

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \le \frac{L_1}{2} \|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n, \tag{2.5}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product.

Similarly, if x^* is a global minimizer of $f \in \mathcal{C}^{1,1}(\mathbb{R}^n)$, then (2.5) implies that

$$\|\nabla f(x)\|^2 \le 2L_1(f(x) - f(x^*)) \quad \forall x \in \mathbb{R}^n. \tag{2.6}$$

We recall that a differentiable function f is convex if

$$f(y) \ge f(x) + \langle \nabla f(x), y - x \rangle \quad \forall x, y \in \mathbb{R}^n. \tag{2.7}$$

2.2 Gaussian Smoothing

We now examine properties of the Gaussian approximation of f in (2.3). For $\mu \ne 0$, we let $g_\mu(x)$ be the first-order-difference approximation of the derivative of $f(x)$ in the direction $u \in \mathbb{R}^n$,

$$g_\mu(x) = \frac{f(x + \mu u) - f(x)}{\mu} u,$$

where the nontrivial direction u is implicitly assumed. By $\nabla f_\mu(x)$ we denote the gradient (with respect to x) of the Gaussian approximation in (2.3). For standard (mean zero, covariance I_n) Gaussian random vectors u and a scalar $p \ge 0$, we define

$$M_p \equiv \mathbb{E}_u[\|u\|^p] = \frac{1}{(2\pi)^{\frac{n}{2}}} \int_{\mathbb{R}^n} \|u\|^p e^{-\frac{1}{2}\|u\|^2} du. \tag{2.8}$$

We summarize the relationships for Gaussian smoothing from [15] upon which we will rely in the following lemma.

LEMMA 2.1. Let $u \in \mathbb{R}^n$ be a normally distributed Gaussian vector. Then, the following are true.

(a) For M_p defined in (2.8), we have

$$M_p \le n^{p/2}, \quad \text{for } p \in [0, 2], \quad \text{and} \tag{2.9}$$

$$M_p \le (n + p)^{p/2}, \quad \text{for } p > 2. \tag{2.10}$$

4

Algorithm 1 (STARS: STep-size Approximation in Randomized Search)

1: Choose initial point x_1, iteration limit N, stepsizes $\{h_k\}_{k\geq 1}$. Evaluate the function at the initial point to obtain $\tilde{f}(x_1; \xi_0)$. Set $k \leftarrow 1$.
2: Generate a random Gaussian vector u_k, and compute the smoothing parameter μ_k.
3: Evaluate the function value $\tilde{f}(x_k + \mu_k u_k; \xi_k)$.
4: Call the stochastic gradient-free oracle

$$s_{\mu_k}(x_k; u_k, \xi_k, \xi_{k-1}) = \frac{\tilde{f}(x_k + \mu_k u_k; \xi_k) - \tilde{f}(x_k; \xi_{k-1})}{\mu_k} u_k. \tag{3.1}$$

5: Set $x_{k+1} = x_k - h_k s_{\mu_k}(x_k; u_k, \xi_k, \xi_{k-1})$.
6: Evaluate $\tilde{f}(x_{k+1}; \xi_k)$, update $k \leftarrow k + 1$, and return to Step 2.

(b) If f is convex, then

$$f_\mu(x) \;\geq\; f(x) \quad \forall x \in \mathbb{R}^n. \tag{2.11}$$

(c) If f is convex and $f \in \mathcal{C}^{1,1}(\mathbb{R}^n)$, then

$$|f_\mu(x) - f(x)| \;\leq\; \frac{\mu^2}{2} L_1 n \quad \forall x \in \mathbb{R}^n. \tag{2.12}$$

(d) If f is differentiable at x, then

$$\mathbb{E}_u[g_\mu(x)] \;=\; \nabla f_\mu(x) \quad \forall x \in \mathbb{R}^n. \tag{2.13}$$

(e) If f is differentiable at x and $f \in \mathcal{C}^{1,1}(\mathbb{R}^n)$, then

$$\mathbb{E}_u[\|g_\mu(x)\|^2] \;\leq\; 2(n+4)\|\nabla f(x)\|^2 + \frac{\mu^2}{2} L_1^2 (n+6)^3 \quad \forall x \in \mathbb{R}^n. \tag{2.14}$$

3 The STARS Algorithm

The STARS algorithm for solving (1.1) while having access to the objective f only through its noisy version $\tilde{f}$ is summarized in Algorithm 1.

In general, the Gaussian directions used by Algorithm 1 can come from general Gaussian directions (e.g., with the covariance informed by knowledge about the scaling or curvature of f). For simplicity of exposition, however, we focus on standard Gaussian directions as formalized in Assumption 3.1. The general case can be recovered by a change of variables with an appropriate scaling of the Lipschitz constant(s).

Assumption 3.1 (Assumption about direction u). *In each iteration k of Algorithm 1, u_k is a vector drawn from a multivariate normal distribution with mean 0 and covariance matrix I_n; equivalently, each element of u is independently and identically distributed (i.i.d.) from a standard normal distribution, $\mathcal{N}(0,1)$.*

What remains to be specified is the smoothing stepsize μ_k. It is computed by incorporating the noise information so that the approximation of the directional derivative has minimum error. We address two types of noise: *additive noise* (Section 4) and *multiplicative noise* (Section 5). These two forms of how $\tilde{f}$ depends on the random variable ξ correspond to two ways that noise often enters a system. The following sections provide near-optimal expressions for μ_k and a convergence rate analysis for both cases.

Importantly, we note Algorithm 1 allows the random variables ξ_k and ξ_{k-1} used in (3.1) to be different from one another. This generalization is in contrast to the stochastic optimization methods examined in [15], where it is assumed the same random variables are used in the smoothing calculation. This generalization does not affect the additive noise case, but will complicate the multiplicative noise case.

4 Additive Noise

We first consider an *additive noise* model for the stochastic objective function $\tilde{f}$:

$$\tilde{f}(x;\xi) = f(x) + \nu(x;\xi), \tag{4.1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth, deterministic function, $\xi \in \Xi$ is a random vector with probability distribution $P(\xi)$, and $\nu(x;\xi)$ is the stochastic noise component.

We make the following assumptions about f and ν.

Assumption 4.1 (Assumption about f). $f \in \mathcal{C}^{1,1}(\mathbb{R}^n)$ *and f is convex.*

Assumption 4.2 (Assumption about additive ν).

1. *For all $x \in \mathbb{R}^n$, ν is i.i.d. with bounded variance $\sigma_a^2 = Var(\nu(x;\xi)) > 0$.*

2. *For all $x \in \mathbb{R}^n$, the noise is unbiased; that is, $\mathbb{E}_\xi[\nu(x;\xi)] = 0$.*

We note that σ_a^2 is independent of x since $\nu(x;\xi)$ is identically distributed for all x. The second assumption is nonrestrictive, since if $\mathbb{E}_\xi[\nu(x;\xi)] \neq 0$, we could just redefine $f(x)$ to be $f(x) - \mathbb{E}_\xi[\nu(x;\xi)]$.

4.1 Noise and Finite Differences

Moré and Wild [13] introduce a way of computing the smoothing stepsize μ that mitigates the effects of the noise in $\tilde{f}$ when estimating a first-order directional directive. The method involves analyzing the expectation of the least-squared error between the forward-difference approximation, $\frac{\tilde{f}(x+\mu u;\xi_1) - \tilde{f}(x;\xi_2)}{\mu}$, and the directional derivative of the smooth function, $\langle \nabla f(x), u \rangle$. The authors show that a near-optimal μ can be computed in such a way that the expected error has the tightest upper bound among all such values μ. Inspired by their approach, we consider the least-square error between $\frac{\tilde{f}(x+\mu u;\xi_1) - \tilde{f}(x;\xi_2)}{\mu}u$ and $\langle \nabla f(x), u \rangle u$. That is, our goal is to find μ^* that minimizes an upper bound on $\mathbb{E}[\mathcal{E}(\mu)]$, where

$$\mathcal{E}(\mu) \equiv \mathcal{E}(\mu; x, u, \xi_1, \xi_2) = \left\| \frac{\tilde{f}(x+\mu u;\xi_1) - \tilde{f}(x;\xi_2)}{\mu}u - \langle \nabla f(x), u \rangle u \right\|^2.$$

We recall that u, ξ_1, and ξ_2 are independent random variables.

Theorem 4.3. *Let Assumptions 3.1, 4.1, and 4.2 hold. If a smoothing stepsize is chosen as*

$$\mu^* = \left[\frac{8\sigma_a^2 n}{L_1^2(n+6)^3}\right]^{\frac{1}{4}}, \tag{4.2}$$

then for any $x \in \mathbb{R}^n$, we have

$$\mathbb{E}_{u,\xi_1,\xi_2}[\mathcal{E}(\mu^*)] \leq \sqrt{2}L_1\sigma_a\sqrt{n(n+6)^3}. \tag{4.3}$$

Proof. Using (4.1) and (2.5), we derive

$$
\begin{aligned}
\mathcal{E}(\mu) &\leq \left\|\frac{\nu(x+\mu u;\xi_1) - \nu(x;\xi_2)}{\mu}u + \frac{\mu L_1}{2}\|u\|^2 u\right\|^2 \\
&\leq \left(\frac{\nu(x+\mu u;\xi_1) - \nu(x;\xi_2)}{\mu} + \frac{\mu L_1}{2}\|u\|^2\right)^2 \|u\|^2.
\end{aligned}
$$

Let $X = \frac{\nu(x+\mu u;\xi_1) - \nu(x;\xi_2)}{\mu} + \frac{\mu L_1}{2}\|u\|^2$. By Assumption 4.2, the expectation of X with respect to ξ_1 and ξ_2 is $\mathbb{E}_{\xi_1,\xi_2}[X] = \frac{\mu L_1}{2}\|u\|^2$, and the corresponding variance is $\text{Var}(X) = \frac{2\sigma_a^2}{\mu^2}$. It then follows that

$$\mathbb{E}_{\xi_1,\xi_2}[X^2] = (\mathbb{E}_{\xi_1,\xi_2}[X])^2 + \text{Var}(X) = \frac{\mu^2 L_1^2}{4}\|u\|^4 + \frac{2\sigma_a^2}{\mu^2}.$$

Hence, taking the expectation of $\mathcal{E}(\mu)$ with respect to u, ξ_1, and ξ_2 yields

$$
\begin{aligned}
\mathbb{E}_{u,\xi_1,\xi_2}[\mathcal{E}(\mu)] &\leq \mathbb{E}_u\left[\mathbb{E}_{\xi_1,\xi_2}[X^2\|u\|^2]\right] \\
&= \mathbb{E}_u\left[\frac{\mu^2 L_1^2}{4}\|u\|^6 + \frac{2\sigma_a^2}{\mu^2}\|u\|^2\right].
\end{aligned}
$$

Using (2.9) and (2.10), we can further derive

$$\mathbb{E}_{u,\xi_1,\xi_2}[\mathcal{E}(\mu)] \leq \frac{\mu^2 L_1^2}{4}(n+6)^3 + \frac{2\sigma_a^2}{\mu^2}n. \tag{4.4}$$

The right-hand side of (4.4) is uniformly convex in μ and has a global minimizer of

$$\mu^* = \left[\frac{8\sigma_a^2 n}{L_1^2(n+6)^3}\right]^{\frac{1}{4}},$$

with the corresponding minimum value yielding (4.3). $\qquad\square$

Remarks:

- A key observation is that for a function $\tilde{f}(x;\xi)$ with additive noise, as long as the noise has a constant variance $\sigma_a > 0$, the optimal choice of the stepsize μ^* is independent of x.

- Since the proof of Theorem 4.3 does not rely on the convexity assumption about f, the error bound (4.3) for the finite-difference approximation also holds for the nonconvex case. The convergence rate analysis for STARS presented in the next section, however, will assume convexity of f; the nonconvex case is out of the scope of this paper but is of interest for future research.

4.2 Convergence Rate Analysis

We now examine the convergence rate of Algorithm 1 applied to the additive noise case of (4.1) and with $\mu_k = \mu^*$ for all k. One of the main ideas behind this convergence proof relies on the fact that we can derive the improvement in f achieved by each step in terms of the change in x. Since the distance between the starting point and the optimal solution, denoted by $R = \|x_0 - x^*\|$, is finite, one can derive an upper bound for the "accumulative improvement in f," $\frac{1}{N+1}\sum_{k=0}^{N}(\mathbb{E}[f(x_k)] - f^*)$. Hence, we can show that increasing the number of iterations, N, of Algorithm 1 yields higher accuracy in the solution.

For simplicity, we denote by $\mathbb{E}[\cdot]$ the expectation over all random variables (i.e., $\mathbb{E}[\cdot] = \mathbb{E}_{u_k,\ldots,u_1,\xi_k,\ldots,\xi_0}[\cdot]$), unless otherwise specified. Similarly, we denote $s_{\mu_k}(x_k; u_k, \xi_k, \xi_{k-1})$ in (3.1) by s_{μ_k}. The following lemma directly follows from Theorem 4.3.

LEMMA 4.4. Let Assumptions 3.1, 4.1, and 4.2 hold. If the smoothing stepsize μ_k is set to the constant μ^* from (4.2), then Algorithm 1 generates steps satisfying

$$\mathbb{E}[\|s_{\mu_k}\|^2] \leq 2(n+4)\|\nabla f(x_k)\|^2 + C_2,$$

where $C_2 = 2\sqrt{2}L_1\sigma_a\sqrt{n(n+6)^3}$.

Proof. Let $g_0(x_k) = \langle\nabla f(x_k), u_k\rangle u_k$. Then (4.3) implies that

$$\mathbb{E}[\|s_{\mu_k}\|^2 - 2\langle s_{\mu_k}, g_0(x_k)\rangle + \|g_0(x_k)\|^2] \leq C_1, \tag{4.5}$$

where $C_1 = \sqrt{2}L_1\sigma_a\sqrt{n(n+6)^3}$.

The stochastic gradient-free oracle s_{μ_k} in (3.1) is a random approximation of the gradient $\nabla f(x_k)$. Furthermore, the expectation of s_{μ_k} with respect to ξ_k and ξ_{k-1} yields the forward-difference approximation of the derivative of f in the direction u_k at x_k:

$$\mathbb{E}_{\xi_k,\xi_{k-1}}[s_{\mu_k}] = \frac{f(x_k + \mu_k u_k) - f(x_k)}{\mu_k}u_k = g_\mu(x_k). \tag{4.6}$$

Combining (4.5) and (4.6) yields

$$
\begin{aligned}
\mathbb{E}\left[\|s_{\mu_k}\|^2\right] &\leq & \mathbb{E}[2\langle s_{\mu_k}, g_0(x_k)\rangle - \|g_0(x_k)\|^2] + C_1 \\
&\overset{(4.6)}{=} & \mathbb{E}_{u_k}[2\langle g_\mu(x_k), g_0(x_k)\rangle - \|g_0(x_k)\|^2] + C_1 \\
&= & \mathbb{E}_{u_k}[-\|g_0(x_k) - g_\mu(x_k)\|^2 + \|g_\mu(x_k)\|^2] + C_1 \\
&\leq & \mathbb{E}_{u_k}[\|g_\mu(x_k)\|^2] + C_1 \\
&\overset{(2.14)}{\leq} & 2(n+4)\|\nabla f(x_k)\| + C_2,
\end{aligned}
$$

where $C_2 = C_1 + \frac{\mu_k^2}{2}L_1^2(n+6)^3 = 2\sqrt{2}L_1\sigma_a\sqrt{n(n+6)^3}$. $\qquad\square$

We are now ready to show convergence of the algorithm. Denote $x^* \in \mathbb{R}^n$ a minimizer associated with $f^* = f(x^*)$. Denote by $\mathcal{U}_k = \{u_1, \cdots, u_k\}$ the set of i.i.d. random variable realizations attached to each iteration of Algorithm 1. Similarly, let $\mathcal{P}_k = \{\xi_0, \cdots, \xi_k\}$. Define $\phi_0 = f(x_0)$ and $\phi_k = \mathbb{E}_{\mathcal{U}_{k-1}, \mathcal{P}_{k-1}}[f(x_k)]$ for $k \geq 1$.

Theorem 4.5. *Let Assumptions 3.1, 4.1, and 4.2 hold. Let the sequence $\{x_k\}_{k\geq 0}$ be generated by Algorithm 1 with the smoothing stepsize μ_k set as μ^* in (4.2). If the fixed step length is $h_k = h = \frac{1}{4L_1(n+4)}$ for all k, then for any $N \geq 0$, we have*

$$\frac{1}{N+1}\sum_{k=0}^{N}(\phi_k - f^*) \leq \frac{4L_1(n+4)}{N+1}\|x_0 - x^*\|^2 + \frac{3\sqrt{2}}{5}\sigma_a(n+4).$$

Proof. We start with deriving the expectation of the change in x of each step, that is, $\mathbb{E}[r_{k+1}^2] - r_k^2$, where $r_k = \|x_k - x^*\|$. First,

$$
\begin{aligned}
r_{k+1}^2 &= \|x_k - h_k s_{\mu_k} - x^*\|^2 \\
&= r_k^2 - 2h_k\langle s_{\mu_k}, x_k - x^*\rangle + h_k^2\|s_{\mu_k}\|^2.
\end{aligned}
$$

$\mathbb{E}[s_{\mu_k}]$ can be derived by using (2.13) and (4.6). $\mathbb{E}[\|s_{\mu_k}\|^2]$ is derived in Lemma 4.4. Hence,

$$\mathbb{E}\left[r_{k+1}^2\right] \leq r_k^2 - 2h_k\langle\nabla f_\mu(x_k), x_k - x^*\rangle + h_k^2[2(n+4)\|\nabla f(x_k)\|^2 + C_2].$$

By using (2.7), (2.11), and (2.6), we derive

$$\mathbb{E}\left[r_{k+1}^2\right] \leq r_k^2 - 2h_k(f(x_k) - f_\mu(x^*)) + 4h_k^2 L_1(n+4)(f(x_k) - f(x^*)) + h_k^2 C_2.$$

Combining this expression with (2.12), which bounds the error between $f_\mu(x)$ and $f(x)$, we obtain

$$\mathbb{E}\left[r_{k+1}^2\right] \leq r_k^2 - 2h_k(1 - 2h_k L_1(n+4))(f(x_k) - f^*) + C_3,$$

where $C_3 = h_k^2 C_2 + 2h_k\frac{\mu_k^2}{2}L_1 n = h_k^2 C_2 + 2\sqrt{2}h_k\sigma_a\sqrt{\frac{n^3}{(n+6^3)}}$.

Let $h_k = h = \frac{1}{4L_1(n+4)}$. Then,

$$\mathbb{E}\left[r_{k+1}^2\right] \leq r_k^2 - \frac{f(x_k) - f^*}{4L_1(n+4)} + C_3, \tag{4.7}$$

where $C_3 = \frac{\sqrt{2}\sigma_a}{2L_1}g_1(n)$ and $g_1(n) = \frac{\sqrt{n(n+6)^3}}{4(n+4)^2} + \frac{1}{n+4}\sqrt{\frac{n^3}{(n+6)^3}}$. By showing that $g_1'(n) < 0$ for all $n \geq 10$ and $g_1'(n) > 0$ for all $n \leq 9$, we can prove that $g_1(n) \leq \max\{g(9), g(10)\} = \max\{0.2936, 0.2934\} \leq 0.3$. Hence, $C_3 \leq \frac{3\sqrt{2}\sigma_a}{20L_1}$.

Taking the expectation in $\mathcal{U}_k$ and $\mathcal{P}_k$, we have

$$\mathbb{E}_{\mathcal{U}_k,\mathcal{P}_k}[r_{k+1}^2] \leq \mathbb{E}_{\mathcal{U}_{k-1},\mathcal{P}_{k-1}}[r_k^2] - \frac{\phi_k - f^*}{4L_1(n+4)} + \frac{3\sqrt{2}\sigma_a}{20L_1}.$$

Summing these inequalities over $k = 0, \cdots, N$ and dividing by $N + 1$, we obtain the desired result. $\square$

The bound in Theorem 4.5 is valid also for $\hat{\phi}_N = \mathbb{E}_{\mathcal{U}_{k-1},\mathcal{P}_{k-1}}[f(\hat{x}_N)]$, where $\hat{x}_N = \arg\min_x\{f(x) : x \in \{x_0, \cdots, x_N\}\}$. In this case,

$$
\begin{aligned}
\mathbb{E}_{\mathcal{U}_{k-1},\mathcal{P}_{k-1}}[f(\hat{x}_N)] - f^* &\leq \mathbb{E}_{\mathcal{U}_{k-1},\mathcal{P}_{k-1}}\left[\frac{1}{N+1}\sum_{k=0}^{N}(\phi_k - f^*)\right] \\
&\leq \frac{4L_1(n+4)}{N+1}\|x_0 - x^*\|^2 + \frac{3\sqrt{2}}{5}\sigma_a(n+4).
\end{aligned}
$$

Hence, in order to achieve a final accuracy of ϵ for $\hat{\phi}_N$ (that is, $\hat{\phi}_N - f^* \leq \epsilon$), the allowable absolute noise in the objective function has to satisfy $\sigma_a \leq \dfrac{5\epsilon}{6\sqrt{2}(n+4)}$. Furthermore, under this bound on the allowable noise, this ϵ accuracy can be ensured by STARS in

$$N = \frac{8(n+4)L_1 R^2}{\epsilon} - 1 \sim \mathcal{O}\left(\frac{n}{\epsilon} L_1 R^2\right) \tag{4.8}$$

iterations, where R^2 is an upper bound on the squared Euclidean distance between the starting point and the optimal solution: $\|x_0 - x^*\|^2 \leq R^2$. In other words, given an optimization problem that has bounded absolute noise of variance σ_a^2, the best accuracy that can be ensured by STARS is

$$\epsilon_{\text{pred}} \geq \frac{6\sqrt{2}\sigma_a(n+4)}{5}, \tag{4.9}$$

and we can solve this noisy problem in $\mathcal{O}\left(\dfrac{n}{\epsilon_{\text{pred}}} L_1 R^2\right)$ iterations. Unsurprisingly, a price must be paid for having access only to noisy realizations, and this price is that arbitrary accuracy cannot be reached in the noisy setting.

5   Multiplicative Noise

A *multiplicative noise* model is described by

$$\tilde{f}(x;\xi) = f(x)[1 + \nu(x;\xi)] = f(x) + f(x)\nu(x;\xi). \tag{5.1}$$

In practice, $|\nu|$ is bounded by something smaller (often much smaller) than 1. A canonical example is when f corresponds to a Monte Carlo integration, with the a stopping criterion based on the value $f(x)$. Similarly, if f is simple and computed in double precision, the relative errors are roughly 10^{-16}; in single precision, the errors are roughly 10^{-8} and in half precision we get errors of roughly 10^{-4}.

Formally, we make the following assumptions in our analysis of STARS for the problem (1.1) with multiplicative noise.

Assumption 5.1 (Assumption about f). *f is continuously differentiable and convex and has Lipschitz constant L_0. ∇f has Lipschitz constant L_1.*

Assumption 5.2 (Assumption about multiplicative ν).

1. *ν is i.i.d., with zero mean and bounded variance; that is, $\mathbb{E}[\nu] = 0$, $\sigma_r^2 = Var(\nu) > 0$.*

2. *The expectation of the signal-to-noise ratio is bounded; that is, $\mathbb{E}[\frac{1}{1+\nu}] \leq b$.*

3. *The support of ν (i.e., the range of values that ν can take with positive probability) is bounded by $\pm a$, where $a < 1$.*

The first part of Assumption 5.2 is analogous to that in Assumption 4.2 and guarantees that the distribution of ν is independent of x. Although not specifying a distributional form for ν (with respect to ξ), the final two parts of Assumption 5.2 are made to simplify the presentation and rule out cases where the noise completely corrupts the function.

5.1 Noise and Finite Differences

Analogous to Theorem 4.3, Theorem 5.3 shows how to compute the near-optimal stepsizes in the multiplicative noise setting.

Theorem 5.3. *Let Assumptions 5.1 and 5.2 hold. If a forward-difference parameter is chosen as*

$$\mu^* = C_4\sqrt{|f(x)|}, \quad where \ C_4 = \left[\frac{16\sigma_r^2 n}{L_1^2(1+3\sigma_r^2)(n+6)^3}\right]^{\frac{1}{4}},$$

then for any $x \in \mathbb{R}^n$ we have

$$\mathbb{E}_{u,\xi_1,\xi_2}[\mathcal{E}(\mu^*)] \leq 2L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3}|f(x)| + 3L_0^2\sigma_r^2(n+4)^2. \tag{5.2}$$

Proof. By using (5.1) and (2.5), we derive

$$
\begin{aligned}
\mathcal{E}(\mu) &\leq \left\|\frac{f(x+\mu u)\nu(x+\mu u;\xi_1) - f(x)\nu(x;\xi_2)}{\mu}u + \frac{\mu L_1}{2}\|u\|^2 u\right\|^2 \\
&\leq \left(\frac{f(x+\mu u)\nu(x+\mu u;\xi_1) - f(x)\nu(x;\xi_2)}{\mu} + \frac{\mu L_1}{2}\|u\|^2\right)^2 \|u\|^2.
\end{aligned}
$$

Again applying (2.5), we get $\mathcal{E}(\mu) \leq X^2\|u\|^2$, where

$$
\begin{aligned}
X &= \frac{f(x+\mu u)\nu(x+\mu u;\xi_1) - f(x)\nu(x;\xi_2)}{\mu} + \frac{\mu L_1}{2}\|u\|^2 \\
&\leq \left(\frac{f(x)}{\mu} + \nabla f(x)^T u + \frac{\mu L_1}{2}\|u\|^2\right)\nu(x+\mu u;\xi_1) - \frac{f(x)}{\mu}\nu(x;\xi_2) + \frac{\mu L_1}{2}\|u\|^2.
\end{aligned}
$$

The expectation of X with respect to ξ_1 and ξ_2 is

$$E_{\xi_1,\xi_2}[X] = \frac{\mu L_1}{2}\|u\|^2$$

and the corresponding variance is

$$
\begin{aligned}
\mathrm{Var}(X) &= \left(\frac{f(x)}{\mu} + \nabla f(x)^T u + \frac{\mu L_1}{2}\|u\|^2\right)^2\sigma_r^2 + \frac{f^2(x)}{\mu^2}\sigma_r^2 \\
&\leq \left(\frac{3f^2(x)}{\mu^2} + 3(\nabla f(x)^T u)^2 + \frac{3\mu^2 L_1^2}{4}\|u\|^4\right)\sigma_r^2 + \frac{f^2(x)}{\mu^2}\sigma_r^2 \\
&= \left(\frac{4f^2(x)}{\mu^2} + 3(\nabla f(x)^T u)^2 + \frac{3\mu^2 L_1^2}{4}\|u\|^4\right)\sigma_r^2,
\end{aligned}
$$

where the inequality holds because $(a + b + c)^2 \leq 3a^2 + 3b^2 + 3c^2$ for any a, b, c. Since $\mathbb{E}[X^2] = \mathrm{Var}(X) + (\mathbb{E}[X])^2$, we have that

$$
\begin{aligned}
\mathbb{E}_{\xi_1,\xi_2}[X^2] &\leq \frac{\mu^2 L_1^2(1+3\sigma_r^2)}{4}\|u\|^4 + \frac{4\sigma_r^2}{\mu^2}f^2(x) + 3(\nabla f(x)^T u)^2\sigma_r^2 \\
&\leq \frac{\mu^2 L_1^2(1+3\sigma_r^2)}{4}\|u\|^4 + \frac{4\sigma_r^2}{\mu^2}f^2(x) + 3L_0^2\sigma_r^2\|u\|^2.
\end{aligned}
$$

11

Hence, we can derive

$$
\begin{aligned}
\mathbb{E}[\mathcal{E}(\mu)] & \leq \mathbb{E}_u[\mathbb{E}_{\xi_1,\xi_2}[X^2 \|u\|^2]] \\
& = \mathbb{E}_u[\|u\|^2 \mathbb{E}_{\xi_1,\xi_2}[X^2]] \\
& \leq \mathbb{E}_u\left[\frac{\mu^2 L_1^2(1+3\sigma_r^2)}{4}\|u\|^6 + \frac{4\sigma_r^2}{\mu^2}f^2(x)\|u\|^2 + 3L_0^2\sigma_r^2\|u\|^4\right].
\end{aligned}
$$

By using (2.9), (2.10), and this last expression, we get

$$
\mathbb{E}[\mathcal{E}(\mu)] \leq \frac{\mu^2 L_1^2(1+3\sigma_r^2)}{4}(n+6)^3 + \frac{4\sigma_r^2 n}{\mu^2}f^2(x) + 3L_0^2\sigma_r^2(n+4)^2.
$$

The right-hand side of this expression is uniformly convex in μ and attains its global minimum at $\mu^* = C_4\sqrt{|f(x)|}$; the corresponding expectation of the least-squares error is

$$
\mathbb{E}_{u,\xi_1,\xi_2}[\mathcal{E}(\mu^*)] \leq 2L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3}|f(x)| + 3L_0^2\sigma_r^2(n+4)^2.
$$

$\square$

Unlike for the absolute noise case of Section 4, the optimal μ value in Theorem 5.3 is not independent of x. Furthermore, letting $\mu_k = \mu^* = C_4\sqrt{|f(x)|}$ assumes that f is known. Unfortunately, we have access to f only through $\tilde{f}$. However, we can compute an estimate, $\tilde{\mu}$, of μ^* by substituting f with $\tilde{f}$ and still derive an error bound. To simplify the derivations, we introduce another random variable, ξ_3, independent of ξ_1 and ξ_2, to compute $\tilde{\mu} \equiv \tilde{\mu}(x; \xi_3)$. The goal is to obtain an upper bound on $\mathbb{E}_{\xi_3}[\mathbb{E}_{\xi_1,\xi_2,u}[\mathcal{E}(\tilde{\mu})]]$, where

$$
\mathcal{E}(\tilde{\mu}) \equiv \mathcal{E}(\tilde{\mu}, x; u, \xi_1, \xi_2, \xi_3) = \left\|\frac{\tilde{f}(x+\tilde{\mu};\xi_1) - \tilde{f}(x;\xi_2)}{\tilde{\mu}}u - \langle\nabla f(x), u\rangle u\right\|^2.
$$

This then allows us to proceed with the usual derivations while requiring only an additional expectation over ξ_3.

LEMMA 5.4. Let Assumptions 5.1 and 5.2 hold. If a forward-difference parameter is chosen as

$$
\tilde{\mu} = C_4\sqrt{|\tilde{f}(x;\xi_3)|}, \quad \text{where } C_4 = \left[\frac{16\sigma_r^2 n}{L_1^2(1+3\sigma_r^2)(n+6)^3}\right]^{\frac{1}{4}}, \tag{5.3}
$$

then for any $x \in \mathbb{R}^n$, we have

$$
\mathbb{E}_{u,\xi_1,\xi_2,\xi_3}[\mathcal{E}(\tilde{\mu})] \leq (1+b)L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3}|f(x)| + 3L_0^2\sigma_r^2(n+4)^2. \tag{5.4}
$$

Proof.

$$
\begin{aligned}
\mathbb{E}[\mathcal{E}(\tilde{\mu})] & = \mathbb{E}_{\xi_3}[\mathbb{E}_{u,\xi_1,\xi_2}[\mathcal{E}(\tilde{\mu})]] \\
& \leq \mathbb{E}_{\xi_3}\left[\frac{\tilde{\mu}^2 L_1^2(1+3\sigma_r^2)}{4}(n+6)^3 + \frac{4\sigma_r^2 n}{\tilde{\mu}^2}f^2(x) + 3L_0^2\sigma_r^2(n+4)^2\right] \\
& = L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3}|f(x)|\mathbb{E}_{\xi_3}\left[1 + \nu(x;\xi_3) + \frac{1}{1+\nu(x;\xi_3)}\right] + 3L_0^2\sigma_r^2(n+4)^2 \\
& \leq (1+b)L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3}|f(x)| + 3L_0^2\sigma_r^2(n+4)^2,
\end{aligned}
$$

where the last inequality holds by Assumption 5.2 because the expectation of the signal-to-noise ratio is bounded by b. $\square$

Remark: Similar to the additive noise case, Theorem 5.3 and Theorem 5.4 do not require f to be convex. Hence, (5.2) and (5.4) both hold in the nonconvex case. However, the following convergence rate analysis applies only to the convex case, since Lemma 5.6 relies on a convexity assumption for f.

5.2 Convergence Rate Analysis

Let $\mu_k = \tilde{\mu} = C_4\sqrt{|\tilde{f}(x_k; \xi_{k'})|}$ in Algorithm 1. Before showing the convergence result, we derive $\mathbb{E}[\langle s_{\tilde{\mu}}, x_k - x^* \rangle]$ and $\mathbb{E}[\|s_{\tilde{\mu}}\|^2]$, where $s_{\tilde{\mu}}$ denotes $s_\mu(x_k; u_k, \xi_k, \xi_{k-1}, \xi_{k'})$ and $\mathbb{E}[\cdot]$ denotes the expectation over all random variables u_k, ξ_k, ξ_{k-1}, and $\xi_{k'}$(i.e., $\mathbb{E}[\cdot] = \mathbb{E}_{u_k, \xi_k, \xi_{k-1}, \xi_{k'}}[\cdot]$), unless otherwise specified.

LEMMA 5.5. Let Assumptions 5.1 and 5.2 hold. If $\mu_k = \tilde{\mu} = C_4\sqrt{|\tilde{f}(x_k; \xi_{k'})|}$, then

$$\mathbb{E}[\|s_{\tilde{\mu}}\|^2] \leq 2(n+4)\|\nabla f(x_k)\|^2 + C_5|f(x_k)| + C_6,$$

where $C_5 = \frac{1}{2}C_4^2 L_1^2(n+6)^3 + (1+b)L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3}$ and $C_6 = 3L_0^2\sigma_r^2(n+4)^2$.

Proof. Let $g_0(x_k) = \langle \nabla f(x_k), u_k \rangle u_k$. The bound (5.3) in Theorem 5.4 implies that

$$\mathbb{E}[\|s_{\tilde{\mu}} - g_0(x_k)\|^2] \leq (1+b)L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3}|f(x)| + 3L_0^2\sigma_r^2(n+4)^2 \equiv \ell(x).$$

Hence,

$$
\begin{aligned}
& \mathbb{E}\left[\|s_{\tilde{\mu}}\|^2\right] \\
\leq\; & \mathbb{E}_{\xi_{k'}}\left[\mathbb{E}_{u_k, \xi_k, \xi_{k-1}}[2\langle s_\mu, g_0(x_k)\rangle - \|g_0(x_k)\|^2]\right] + \ell(x) \\
\overset{(4.6)}{=}\; & \mathbb{E}_{\xi_{k'}}\left[\mathbb{E}_{u_k}[2\langle g_{\mu_k}(x_k), g_0(x_k)\rangle - \|g_0(x_k)\|^2]\right] + \ell(x) \\
\leq\; & \mathbb{E}_{\xi_{k'}}\left[\mathbb{E}_{u_k}[\|g_{\mu_k}(x_k)\|^2]\right] + \ell(x) \\
\overset{(2.14)}{\leq}\; & 2(n+4)\|\nabla f(x_k)\|^2 + \mathbb{E}_{\xi_{k'}}\left[\frac{\mu_k^2}{2}L_1^2(n+6^3)\right] + \ell(x) \\
=\; & 2(n+4)\|\nabla f(x_k)\|^2 + C_5|f(x_k)| + C_6,
\end{aligned}
$$

where the last equality holds since $\mathbb{E}_{\xi_{k'}}[\mu_k^2] = \mathbb{E}_{\xi_{k'}}[C_4^2|f(x_k)|(1+\nu(x_k; \xi_{k'})] = C_4^2|f(x_k)|$. $\quad\square$

LEMMA 5.6. Let Assumptions 5.1 and 5.2 hold. If $\mu_k = \tilde{\mu} = C_4\sqrt{|\tilde{f}(x_k; \xi_{k'})|}$, then

$$\mathbb{E}[\langle s_{\tilde{\mu}}, x_k - x^* \rangle] \geq f(x_k) - f^* - \frac{C_4^2 L_1 n}{2}|f(x_k)|.$$

Proof. First, we have

$$
\begin{aligned}
\mathbb{E}_{u_k,\xi_k,\xi_{k-1}}[s_{\tilde{\mu}}] &= \mathbb{E}_{u_k,\xi_k,\xi_{k-1}}\left[\frac{\tilde{f}(x_k+\mu_k u_k;\xi_k)-\tilde{f}(x_k;\xi_{k-1})}{\mu_k}u_k\right] \\
&= \mathbb{E}_{u_k,\xi_k,\xi_{k-1}}\left[\frac{f(x_k+\mu_k u_k)[1+\nu(x_k+\mu_k u_k;\xi_k)]-f(x_k)[1+\nu(x_k;\xi_{k-1})]}{\mu_k}u_k\right] \\
&= \mathbb{E}_{u_k}\left[\frac{f(x_k+\mu_k u_k)-f(x_k)}{\mu_k}u_k\right] \\
&= \mathbb{E}_{u_k}[g_{\mu_k}(x_k)] \\
&\overset{(2.13)}{=} \nabla f_{\mu_k}(x_k).
\end{aligned}
$$

Then, we get

$$
\begin{aligned}
\mathbb{E}_{u_k,\xi_k,\xi_{k-1}}[\langle s_{\tilde{\mu}}, x_k-x^*\rangle] &= \langle \nabla f_{\mu_k}(x_k), x_k-x^*\rangle \\
&\overset{(2.7)}{\geq} f_{\mu_k}(x_k)-f_{\mu_k}(x^*) \\
&\overset{(2.11)}{\geq} f(x_k)-f_{\mu_k}(x^*) \\
&\overset{(2.12)}{\geq} f(x_k)-f^*-\frac{\mu_k}{2}L_1 n.
\end{aligned}
$$

Since $\mu_k = \tilde{\mu} = C_4\sqrt{|\tilde{f}(x_k;\xi_{k'})|}$, we have

$$
\mathbb{E}[\langle s_{\tilde{\mu}}, x_k-x^*\rangle] = \mathbb{E}_{\xi_{k'}}[\mathbb{E}_{u_k,\xi_k,\xi_{k-1}}[\langle s_{\tilde{\mu}}, x_k-x^*\rangle]] \geq f(x_k)-f^*-\frac{C_4^2 L_1 n}{2}|f(x_k)|.
$$

$\square$

We are now ready to show the convergence of Algorithm 1, with $\mu_k = \tilde{\mu}$, for the minimization of a function (5.1) with bounded multiplicative noise.

Theorem 5.7. *Let Assumptions 5.1 and 5.2 hold. Let the sequence $\{x_k\}_{k\geq 0}$ be generated by Algorithm 1 with the smoothing parameter μ_k being*

$$
\mu_k = \tilde{\mu} = C_4\sqrt{|\tilde{f}(x;\xi_{k'})|}
$$

and the fixed step length set to $h_k = h = \frac{1}{4L_1(n+4)}$ for all k. Let M be an upper bound on the average of the historical absolute values of noise-free function evaluations; that is,

$$
M \geq \frac{1}{N+1}\sum_{k=0}^{N}|\phi_k| = \frac{1}{N+1}\left(|f(x_0)|+\sum_{k=1}^{N}\mathbb{E}_{\mathcal{U}_{k-1},\mathcal{P}_{k-1}}[|f(x_k)|]\right).
$$

Then, for any $N\geq 0$ we have

$$
\frac{1}{N+1}\sum_{k=0}^{N}(\phi_k-f^*) \leq \frac{4L_1(n+4)}{N+1}\|x_0-x^*\|^2 + 4L_1(n+4)\left(C_7 M+C_8\right), \tag{5.5}
$$

where $C_7 = \frac{C_4^2 n}{4(n+4)} + \frac{C_5}{16L_1^2(n+4)^2}$ and $C_8 = \frac{C_6}{16L_1^2(n+4)^2}$.

14

Proof. Let $r_k = \|x_k - x^*\|$. First,

$$\begin{aligned} r_{k+1}^2 &= \|x_k - h_k s_{\tilde{\mu}} - x^*\|^2 \\ &= r_k^2 - 2h_k\langle s_{\tilde{\mu}}, x_k - x^*\rangle + h_k^2\|s_{\tilde{\mu}}\|^2. \end{aligned}$$

$\mathbb{E}[\langle s_{\tilde{\mu}}, x_k - x^*\rangle]$ and $\mathbb{E}[\|s_{\tilde{\mu}}\|^2]$ are derived in Lemma 5.6 and Lemma 5.5, respectively. Hence, incorporating (2.6), we derive

$$\begin{aligned} \mathbb{E}\left[r_{k+1}^2\right] &\leq r_k^2 - 2h_k(f(x_k) - f^* - \frac{C_4^2 L_1 n}{2}|f(x_k)|) + h_k^2[2(n+4)\|\nabla f(x_k)\|^2 + C_5|f(x_k)| + C_6] \\ &\leq r_k^2 - 2h_k(1 - 2h_k L_1(n+4))(f(x_k) - f^*) + (h_k C_4^2 L_1 n + h_k^2 C_5)|f(x_k)| + h_k^2 C_6. \end{aligned}$$

Let $h_k = \frac{1}{4L_1(n+4)}$. Then, taking the expectation with respect to $\mathcal{U}_k = \{u_1, \cdots, u_k\}$ and $\mathcal{P}_k = \{\xi_0, \xi_0', \xi_1, \xi_{1'}, \cdots, \xi_k\}$ yields

$$\mathbb{E}_{\mathcal{U}_k, \mathcal{P}_k}\left[r_{k+1}^2\right] \leq \mathbb{E}_{\mathcal{U}_{k-1}, \mathcal{P}_{k-1}}\left[r_k^2\right] - \frac{\phi_k - f^*}{4L_1(n+4)} + C_7|\phi_k| + C_8.$$

Summing these inequalities over $k = 0, \cdots, N$ and dividing by $N + 1$, we get

$$\frac{1}{N+1}\sum_{k=0}^N (\phi_k - f^*) \leq \frac{4L_1(n+4)}{N+1}\|x_0 - x^*\|^2 + 4L_1(n+4)(C_7 M + C_8).$$

$\square$

The bound (5.5) is valid also for $\hat{\phi}_N = \mathbb{E}_{\mathcal{U}_{k-1}, \mathcal{P}_{k-1}}[f(\hat{x}_N)]$, where $\hat{x}_N = \arg\min_x\{f(x) : x \in \{x_0, \cdots, x_N\}\}$. In this case,

$$\begin{aligned} \mathbb{E}_{\mathcal{U}_{k-1}, \mathcal{P}_{k-1}}[f(\hat{x}_N)] - f^* &\leq \mathbb{E}_{\mathcal{U}_{k-1}, \mathcal{P}_{k-1}}\left[\frac{1}{N+1}\sum_{k=0}^N(\phi_k - f^*)\right] \\ &\leq \frac{4L_1(n+4)}{N+1}\|x_0 - x^*\|^2 + 4L_1(n+4)(C_7 M + C_8). \quad (5.6) \end{aligned}$$

Let us collect and simplify the constants C_7 and C_8. First, $C_8 = \frac{C_6}{16L_1^2(n+4)^2} = \frac{3L_0^2\sigma_r^2}{16L_1^2}$. Second, since

$$\begin{aligned} C_5 &= \frac{1}{2}C_4^2 L_1^2(n+6)^3 + (1+b)L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3} \\ &= 2L_1\sigma_r\sqrt{\frac{1}{1+3\sigma_r^2}}\sqrt{n(n+6)^3} + (1+b)L_1\sigma_r\sqrt{(1+3\sigma_r^2)n(n+6)^3} \\ &\leq (b+3)L_1\sigma_r\sqrt{1+3\sigma_r^2}\sqrt{n(n+6)^3}, \end{aligned}$$

where the last inequality holds because $\frac{1}{1+3\sigma_r^2} \leq 1 \leq 1 + 3\sigma_r^2$, we can derive

$$\begin{aligned} C_7 &= \frac{C_4^2 n}{4(n+4)} + \frac{C_5}{16L_1^2(n+4)^2} \\ &\leq \frac{1}{L_1}\sqrt{\frac{\sigma_r^2}{1+3\sigma_r^2}} \cdot \frac{n}{n+4}\sqrt{\frac{n}{(n+6)^3}} + \frac{(b+3)\sigma_r\sqrt{1+3\sigma_r^2}}{16L_1} \cdot \frac{\sqrt{n(n+6)^3}}{(n+4)^2} \\ &\leq \frac{\sigma_r\sqrt{1+3\sigma_r^2}}{L_1}[g_2(n) + (b+3)g_3(n)], \end{aligned}$$

15

where $g_2(n) = \frac{n}{n+4}\sqrt{\frac{n}{(n+6)^3}}$, $g_3 = \frac{\sqrt{n(n+6)^3}}{16(n+4)^2}$, and the last inequality again utilizes $\frac{1}{1+3\sigma_r^2} \leq 1 \leq 1 + 3\sigma_r^2$. It can be shown that $g_2'(n) < 0$ for all $n \geq 8$ and $g_2'(n) > 0$ for all $n \leq 7$, thus $g_2(n) \leq \max\{g(7), g(8)\} = \max\{0.0359, 0.0360\} \leq \frac{3}{64}$. Similarly, one can prove that $g_3'(12) = 0$, $g_3'(n) < 0$ for all $n > 12$, and $g_3'(n) > 0$ for all $n < 12$, which indicates $g_3(n) \leq g_3(12) \approx 0.0646 \leq \frac{3}{32}$. Hence,

$$C_7 \leq \frac{3(2b+7)\sigma_r\sqrt{1+3\sigma_r^2}}{64L_1} \leq \frac{3\sqrt{3}(2b+7)(\sigma_r^2 + \frac{1}{6})}{64L_1},$$

where the last inequality holds because $\sigma_r\sqrt{\frac{1}{3} + \sigma_r^2} \leq \sigma_r^2 + \frac{1}{6}$.

With C_7 and C_8 simplified, (5.6) can be used to establish an accuracy ϵ for $\hat{\phi}_N$; that is, $\hat{\phi}_N - f^* \leq \epsilon$, can be achieved in $\mathcal{O}\left(\frac{n}{\epsilon}L_1R^2\right)$ iterations, provided the variance of the relative noise σ_r^2 satisfies

$$4L_1(n+4)(C_7M + C_8) \leq \frac{1}{2}C_9(\sigma_r^2 + \frac{1}{6})(n+4) \leq \frac{\epsilon}{2},$$

where $C_9 = \frac{3\sqrt{3}}{8}(2b+7)M + \frac{3L_0^2}{2L_1}$, that is,

$$\sigma_r^2 \leq \frac{\epsilon}{C_9(n+4)} - \frac{1}{6}. \tag{5.7}$$

The bound in (5.7) may be cause for concern since the upper bound may only be positive for larger values of ϵ. Rearranging the terms explicitly shows that the additive term $\frac{1}{6}$ is a limiting factor for the best accuracy that can be ensured by this bound:

$$\epsilon_{\text{pred}} \geq C_9(\sigma_r^2 + \frac{1}{6})(n+4). \tag{5.8}$$

6 Numerical Experiments

We perform three types of numerical studies. Since our convergence rate analysis guarantees only that the means converge, we first test how much variability the performance of STARS show from one run to another. Second, we study the convergence behavior of STARS in both the absolute noise and multiplicative noise cases and examine these results relative to the bounds established in our analysis. Then, we compare STARS with four other randomized zero-order methods to highlight what is gained by using an adaptive smoothing stepsize.

6.1 Performance Variability

We first examine the variability of the performance of STARS relative to that of Nesterov's RG algorithm [15], which is summarized in Algorithm 2. One can observe that RG and STARS have identical algorithmic updates except for the choice of the smoothing stepsize μ_k. Whereas STARS takes into account the noise level, RG calculates the smoothing stepsize based on the target accuracy ϵ in addition to the problem dimension and Lipschitz constant,

$$\mu = \frac{5}{3(n+4)}\sqrt{\frac{\epsilon}{2L_1}}. \tag{6.1}$$

16

Algorithm 2 (RG: Random Search for Smooth Optimization)

1: Choose initial point x_0 and iteration limit N. Fix step length $h_k = h = \frac{1}{4(n+4)L_1}$ and compute smoothing stepsize μ_k based on $\epsilon = 2^{-16}$. Set $k \leftarrow 1$.

2: Generate a random Gaussian vector u_k.

3: Evaluate the function values $\tilde{f}(x_k; \xi_k)$ and $\tilde{f}(x_k + \mu_k u_k; \xi_k)$.

4: Call the random stochastic gradient-free oracle

$$s_\mu(x_k; u_k, \xi_k) = \frac{\tilde{f}(x_k + \mu_k u_k; \xi_k) - \tilde{f}(x_k; \xi_k)}{\mu_k} u_k.$$

5: Set $x_{k+1} = x_k - h_k s_\mu(x_k; u_k, \xi_k)$, update $k \leftarrow k + 1$, and return to Step 2.

MATLAB implementations of both RG and STARS are tested on a smooth convex function with random noise added in both additive and multiplicative forms. In our tests, we use uniform random noise, with ν generated uniformly from the interval $[-\sqrt{3}\sigma, \sqrt{3}\sigma]$ by using MATLAB's random number generator rand. This choice ensures that ν has zero mean and bounded variance σ^2 in both the additive ($\sigma_a = \sigma$) and multiplicative cases ($\sigma_r = \sigma$) and that Assumptions 4.2 and 5.2 hold, provided that $\sigma < 3^{-1/2}$.

We use Nesterov's smooth function as introduced in [15]:

$$f_1(x) = \frac{1}{2}(x^{(1)})^2 + \frac{1}{2}\sum_{i=1}^{n-1}(x^{(i+1)} - x^i)^2 + \frac{1}{2}(x^{(n)})^2 - x^{(1)}, \tag{6.2}$$

where $x^{(i)}$ denotes the ith component of the vector $x \in \mathbb{R}^n$. The starting point specified for this problem is the vector of zeros, $x_0 = \mathbf{0}$. The optimal solution is

$$x^{*(i)} = 1 - \frac{i}{n+1}, \ i = 1, \cdots, n; \quad f(x^*) = -\frac{n}{2(n+1)}.$$

The analytical values for the parameters (corresponding to Lipschitz constant for the gradient and the squared Euclidean distance between the starting point and optimal solution) are: $L_1 \leq 4$ and $R^2 = \|x_0 - x^*\|^2 \leq \frac{n+1}{3}$. Both methods were given the same parameter value (4.0) for L_1, but the smoothing stepsizes differ. Whereas RG always uses fixed stepsizes of the form (6.1), STARS uses fixed stepsizes of the form (4.2) in the absolute noise case and uses dynamic stepsizes calculated as (5.3) in the multiplicative noise case. To observe convergence over many random trials, we use a small problem dimension of $n = 8$; however, the behavior shown in Figure 6.1 is typical of the behavior that we observed in higher dimensions (but the $n = 8$ case requiring fewer function evaluations).

In Figure 6.1, we plot the accuracy achieved at each function evaluation, which is the true function value $f(x_k)$ minus the optimal function value $f(x^*)$. The median across 20 trials is plotted as a line; the shaded region denotes the best and worst trials; and the 25% and 75% quartiles are plotted as error bars. We observe that when the function is relatively smooth, as in Figure 6.1(a) when the additive noise is 10^{-6}, the methods exhibit
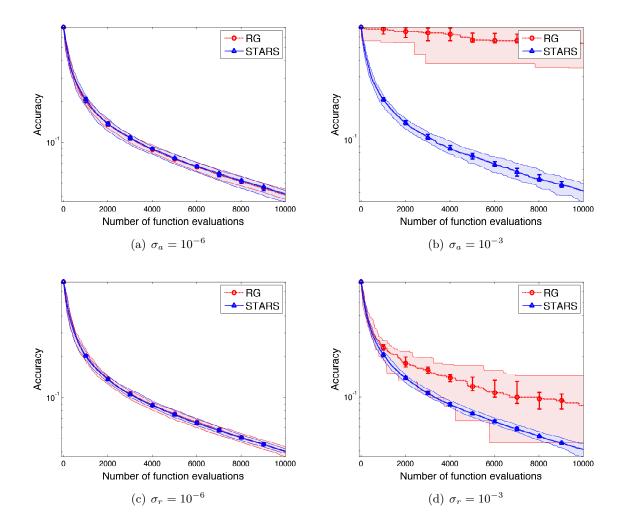
Figure 6.1: Median and quartile plots of achieved accuracy with respect to 20 random seeds when applying RG and STARS to the noisy f_1 function. Figures 6.1(a) and 6.1(b) show the additive noise case, while Figures 6.1(c) and 6.1(d) show the multiplicative noise case.

similar performance. As the function gets more noisy, however, as in Figure 6.1(b) when the additive noise becomes 10^{-4}, RG shows more fluctuations in performance resulting in large variance, whereas the performance STARS is almost the same as in the smoother case. The same noise-invariant behavior of STARS can be observed in the multiplicative case.

6.2   Convergence Behavior

We tested the convergence behavior of STARS with respect to dimension n and noise levels on the same smooth convex function f_1 with noise added in the same way as in Section 6.1. The results are summarized in Figure 6.2 , where (a) and (b) are for the additive case and (c) and (d) are for the multiplicative case. The horizontal axis marks the problem dimension
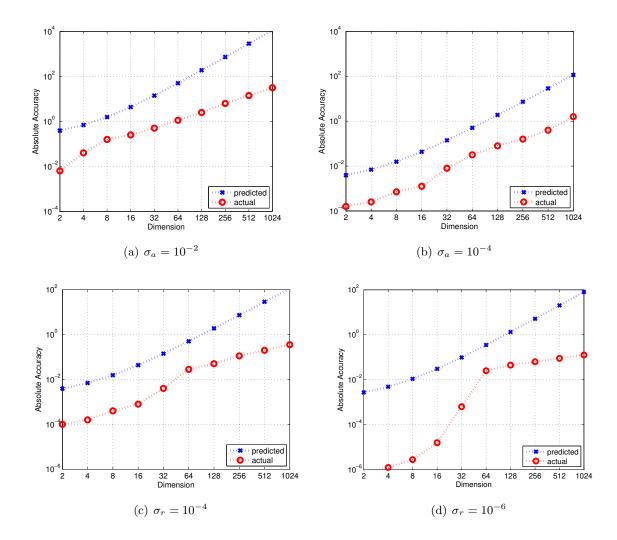
(a) $\sigma_a = 10^{-2}$

(b) $\sigma_a = 10^{-4}$

(c) $\sigma_r = 10^{-4}$

(d) $\sigma_r = 10^{-6}$

Figure 6.2: Convergence behavior of STARS: absolute accuracy versus dimension n. Two absolute noise levels (a) and (b), and two relative noise levels (c) and (d) are presented.

and the vertical axis shows the absolute accuracy. Two types of absolute accuracy are plotted. First, ϵ_{pred} (in blue $\times$'s) is the best achievable accuracy given a certain noise level, computed by using (4.9) for the additive case and (5.7) for the multiplicative case. Second is the actual accuracy (in red circle) achieved by STARS after N iterations where N, calculated as in (4.8), is the number of iterations needed in theory to get ϵ_{pred}. Because of the stochastic nature of STARS, we perform 15 runs (each with a different random seed) of each test and report the averaged accuracy

$$\bar{\epsilon}_{\text{actual}} = \frac{1}{15} \sum_{i=1}^{15} \epsilon^i_{\text{actual}} = \frac{1}{15} \sum_{i=1}^{15} (f(x_N^i) - f^*). \tag{6.3}$$

We observe from Figure 6.2 that the solution obtained by STARS within the iteration limit N is more accurate than that predicted by the theoretical bounds. The difference be-

tween predicted and achieved accuracy is always over an order of magnitude and is relatively consistent for all dimensions we examined.

6.3   Illustrative Example

In this section, we provide a comparison between STARS and four other zero-order algorithms on noisy versions of (6.2) with $n = 8$. The methods we study all share a stochastic nature; that is, a random direction is generated at each iteration. Except for RP [19], which is designed for solving smooth convex functions, the rest are stochastic optimization algorithms. However, we still include RP in the comparison because of its similar algorithmic framework. The algorithms and their function-specific inputs are summarized in Table 6.1, where $\tilde{L}_1$ and $\tilde{\sigma}^2$ are, respectively, estimations of L_1 and σ^2 given a noisy function (details on how to estimate $\tilde{L}_1$ and $\tilde{\sigma}^2$ are discussed in Appendix). We now briefly introduce each of the tested algorithms; algorithmic and implementation details are given in the appendix.

Table 6.1: Relevant function parameters for different methods.

Method Abbreviation	Method Name	Parameters
STARS	Stepsize Approximation in Random Search	L_1, σ^2
SS	Random Search for Stochastic Optimization [15]	L_0, R^2
RSGF	Random Stochastic Gradient Free method [7]	$\tilde{L}_1, \tilde{\sigma}^2$
RP	Random Pursuit [19]	-
ES	(1+1)-Evolution Strategy [18]	-

The first zero-order method we include, named SS (Random Search for Stochastic Optimization), is proposed in [15] for solving (1.1). It assumes that $f \in \mathcal{C}^{0,0}(\mathbb{R}^n)$ is convex. The SS algorithm, summarized in Algorithm 3, shares the same algorithmic framework as STARS except for the choice of smoothing stepsize μ_k and the step length h_k. It is shown that the quantities μ_k and h_k can be chosen so that a solution for (1.1) such that $f(x_N) - f^* \leq \epsilon$ can be ensured by SS in $\mathcal{O}(n^2/\epsilon^2)$ iterations.

Another stochastic zero-order method that also shares an algorithmic framework similar to STARS is RSGF [7], which is summarized in Algorithm 4. RSGF targets the stochastic optimization objective function in (1.1), but the authors relax the convexity assumption and allow f to be nonconvex. However, it is assumed that $\tilde{f}(\cdot, \xi) \in \mathcal{C}^{1,1}(\mathbb{R}^n)$ almost surely, which implies that $f \in \mathcal{C}^{1,1}(\mathbb{R}^n)$. The authors show that the iteration complexity for RSGF finding an ϵ-accurate solution, (i.e., a point $\bar{x}$ such that $\mathbb{E}[\|\nabla f(\bar{x})\|] \leq \epsilon$) can be bounded by $\mathcal{O}(n/\epsilon^2)$. Since such a solution $\bar{x}$ satisfies $f(\bar{x}) - f^* \leq \epsilon$ when f is convex, this bound improves Nesterov's result in [15] by a factor n for convex stochastic optimization problems.

In contrast with the presented randomized approaches that work with a Gaussian vector u, we include an algorithm that samples from a uniform distribution on the unit hypersphere. Summarized in Algorithm 5, RP [19] is designed for unconstrained, smooth, convex optimization. It relaxes the requirement in [15] of approximating directional derivatives via

a suitable oracle. Instead, the sampling directions are chosen uniformly at random on the unit hypersphere, and the step lengths are determined by a line search oracle. This randomized method also requires only zeroth-order information about the objective function, but it does not need any function-specific parametrization. It was shown that RP meets the convergence rates of the standard steepest descent method up to a factor n.

Experimental studies of variants of $(1 + 1)$-Evolution Strategy (ES), first proposed by Schumer and Steiglitz [18], have shown their effectiveness in practice and their robustness in noisy environment. However, provable convergence rates are derived only for the simplest forms of ES on unimodal objective functions [5, 8, 9], such as sphere or ellipsoidal functions. The implementation we study is summarized in Algorithm 6; however, different variants of this scheme have been studied in [6].

We observe from Figure 6.3 that STARS outperforms the other four algorithms in terms of final accuracy in the solution. In both Figures 6.3(a) and 6.3(b), ES is the fastest algorithm among all in the beginning. However, ES stops progressing after a few iterations, whereas STARS keeps progressing to a more accurate solution. As the noise level increases from 10^{-5} to 10^{-1}, the performance of ES gradually worsens, similar to the other methods SS, RSGF, and RP. However, the noise-invariant property of STARS allows it to remain robust in these noisy environments.
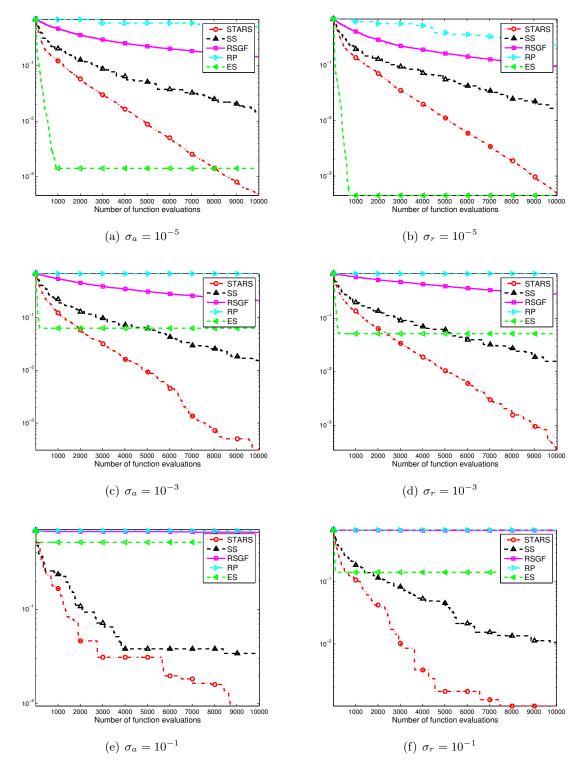
Acknowledgments

Figure 6.3: Trajectory plots of five zero-order methods in the additive and multiplicative noise settings. The vertical axis represents the true function value $f(x_k)$, and each line is the mean of 20 trials.

7 Appendix

In this appendix we describe the implementation details of the four zero-order methods tested in Table 6.1 and Section 6.3.

Random Search for Stochastic Optimization

Algorithm 3 (SS: Random Search for Stochastic Optimization)

1: Choose initial point x_0 and iteration limit N. Fix step length $h_k = h = \frac{R}{(n+4)(N+1)^{1/2}L_0}$ and smoothing stepsize $\mu_k = \mu = \frac{\epsilon}{2L_0 n^{1/2}}$. Set $k \leftarrow 1$.
2: Generate a random Gaussian vector u_k.
3: Evaluate the function values $\tilde{f}(x_k; \xi_k)$ and $\tilde{f}(x_k + \mu_k u_k; \xi_k)$.
4: Call the random stochastic gradient-free oracle

$$s_\mu(x_k; u_k, \xi_k) = \frac{\tilde{f}(x_k + \mu_k u_k; \xi_k) - \tilde{f}(x_k; \xi_k)}{\mu_k} u_k.$$

5: Set $x_{k+1} = x_k - h_k s_\mu(x_k; u_k, \xi_k)$, update $k \leftarrow k + 1$, and return to Step 2.

Algorithm 3 provides the SS (Random Search for Stochastic Optimization) algorithm from [15].

Remark: ϵ is suggested to be 2^{-16} in the experiments in [15]. Our experiments in Section 6.3, however, show that this choice of ϵ forces SS to take small steps and thus SS does not converge at all in the noisy environment. Hence, we increase ϵ (to $\epsilon = 0.1$) to show that optimistically, SS will work if the stepsize is big enough. Although in the additive noise case one can recover STARS by appropriately setting this ϵ in SS, it is not possible in the multiplicative case because STARS takes dynamically adjusted smoothing stepsizes in this case.

Randomized Stochastic Gradient-Free Method

Algorithm 4 provides the RSGF (Randomized Stochastic Gradient-Free Method) algorithm from [7].

Remark: Although the convergence analysis of RSGF is based on knowledge of the constants $Ł_1$ and σ^2, the discussion in [7] on how to implement RSGF does not reply on these inputs. Because the authors solved a support vector machine problem and an inventory problem, both of which do not have known L_1 and σ^2 values, they provide details on how to estimate these parameters given a noisy function. Hence following [7], the parameter L_1 is estimated as the l_2 norm of the Hessian of the deterministic approximation of the noisy objective functions. This estimation is achieved by using a sample average approximation

Algorithm 4 (RSGF: Randomized Stochastic Gradient-Free Method)

1: Choose initial point x_0 and iteration limit N. Estimate L_1 and $\tilde{\sigma}^2$ of the noisy function $\tilde{f}$. Fix step length as

$$\gamma_k = \gamma = \frac{1}{\sqrt{n+4}} \min \left\{ \frac{1}{4L_1\sqrt{n+4}}, \frac{\tilde{D}}{\tilde{\sigma}\sqrt{N}} \right\},$$

where $\tilde{D} = (2f(x_0)/L_1)^{\frac{1}{2}}$. Fix $\mu_k = \mu = 0.0025$. Set $k \leftarrow 1$.

2: Generate a Gaussian vector u_k.

3: Evaluate the function values $\tilde{f}(x_k; \xi_k)$ and $\tilde{f}(x_k + \mu_k u_k; \xi_k)$.

4: Call the stochastic zero-order oracle

$$G_\mu(x_k; u_k, \xi_k) = \frac{\tilde{f}(x_k + \mu_k u_k; \xi_k) - \tilde{f}(x_k; \xi_k)}{\mu} u_k.$$

5: Set $x_{k+1} = x_k - \gamma_k G_\mu(x_k; u_k, \xi_k)$, update $k \leftarrow k+1$, and return to Step 2.

---

approach with 200 i.i.d. samples. Also, we compute the stochastic gradients of the objective functions at these randomly selected points and take the maximum variance of the stochastic gradients as an estimate of $\tilde{\sigma}^2$.

Random Pursuit

Algorithm 5 (RP: Random Pursuit)

1: Choose initial point x_0, iteration limit N, and line search accuracy $\mu = 0.0025$. Set $k \leftarrow 1$.

2: Choose a random Gaussian vector u_k.

3: Choose $x_{k+1} = x_k + \mathsf{LS}_{\mathsf{APPROX}_\mu}(x_k, u_k) \cdot u_k$, update $k \leftarrow k+1$, and return to Step 2.

Algorithm 5 provides the RP (Random Pursuit) algorithm from [19].

Remark: We follow the authors in [19] and use the built-in MATLAB routine fminunc.m as the approximate line search oracle.

$(1+1)$-Evolution Strategy

Algorithm 6 provides the ES ($(1+1)$-Evolution Strategy) algorithm from [18].

Remark: A problem-specific parameter required by Algorithm 6 is the initial stepsize σ_0, which is given in [19] for some of our test functions. The stepsize is multiplied by a factor $c_s = e^{1/3} > 1$ when the mutant's fitness is as good as the parent is and is otherwise

---
Algorithm 6 (ES: $(1+1)$-Evolution Strategy)
___
1: Choose initial point x_0, initial stepsize σ_0, iteration limit N, and probability of improve-
   ment $p = 0.27$. Set $c_s = e^{\frac{1}{3}} \approx 1.3956$ and $c_f = c_s \cdot e^{\frac{-p}{1-p}} \approx 0.8840$. Set $k \leftarrow 1$.
2: Generate a random Gaussian vector u_k.
3: Evaluate the function values $\tilde{f}(x_k; \xi_k)$ and $\tilde{f}(x_k + \sigma_k u_k; \xi_k)$.
4: If $\tilde{f}(x_k + \sigma_k u_k; \xi_k) \leq \tilde{f}(x_k; \xi_k)$, then set $x_{k+1} = x_k + \sigma_k u_k$ and $\sigma_{k+1} = c_s \sigma_k$;
   Otherwise, set $x_{k+1} = x_k$ and $\sigma_{k+1} = c_f \sigma_k$.
5: Update $k \leftarrow k + 1$ and return to Step 2.
___

multiplied by $c_s \cdot e^{\frac{-p}{1-p}} < 1$, where p is the probability of improvement set to the value 0.27 suggested by Schumer and Steiglitz [18].

References

[1] M. A. ABRAMSON AND C. AUDET, *Convergence of mesh adaptive direct search to second-order stationary points*, SIAM Journal on Optimization, 17 (2006), pp. 606–619.

[2] M. A. ABRAMSON, C. AUDET, J. E. DENNIS, JR., AND S. LE DIGABEL, *OrthoMADS: A deterministic MADS instance with orthogonal directions*, SIAM Journal on Optimization, 20 (2009), pp. 948–966.

[3] ALEKH AGARWAL, DEAN P. FOSTER, DANIEL J. HSU, SHAM M. KAKADE, AND ALEXANDER RAKHLIN, *Stochastic convex optimization with bandit feedback*, in Advances in Neural Information Processing Systems 24, 2011, pp. 1035–1043.

[4] C. AUDET AND J. E. DENNIS, JR., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on Optimization, 17 (2006), pp. 188–217.

[5] A. AUGER, *Convergence results for the $(1, \lambda)$-SA-ES using the theory of φ-irreducible Markov chains*, Theoretical Computer Science, 334 (2005), pp. 35–69.

[6] HANS-GEORG BEYER AND HANS-PAUL SCHWEFEL, *Evolution strategies– A comprehensive introduction*, Natural Computing, 1 (2002), pp. 3–52.

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

[8] JENS JÄGERSKÜPPER, *How the (1+1)-ES using isotropic mutations minimizes positive definite quadratic forms*, Theoretical Computer Science, 361 (2006), pp. 38–56.

[9] M. JEBALIA, A. AUGER, AND N. HANSEN, *Log-linear convergence and divergence of the scale-invariant (1+1)-ES in noisy environments*, Algorithmica, 59 (2011), pp. 425–460.

[10] R. M. Lewis, V. Torczon, and M. Trosset, *Direct search methods: Then and now*, Journal of Computational and Applied Mathematics, 124 (2000), pp. 191–207.

[11] J. Matyas, *Random optimization*, Automation and Remote Control, 26 (1965), pp. 246–253.

[12] Jorge J. Moré and Stefan M. Wild, *Estimating computational noise*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1292–1314.

[13] Jorge J. Moré and Stefan M. Wild, *Estimating derivatives of noisy simulations*, ACM Transactions on Mathematical Software, 38 (2012), pp. 19:1–19:21.

[14] Jorge J. Moré and Stefan M. Wild, *Do you trust derivatives or differences?*, Journal of Computational Physics, 273 (2014), pp. 268–277.

[15] Yurii Nesterov, *Random gradient-free minimization of convex functions*, CORE Discussion Papers 2011001, Université Catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2011.

[16] B.T. Polyak, *Introduction to Optimization*, Optimization Software, 1987.

[17] Ben Recht, Kevin G. Jamieson, and Robert Nowak, *Query complexity of derivative-free optimization*, in Advances in Neural Information Processing Systems 25, 2012, pp. 2672–2680.

[18] M. Schumer and K. Steiglitz, *Adaptive step size random search*, IEEE Transactions on Automatic Control, 13 (1968), pp. 270–276.

[19] Sebastian U. Stich, Christian L. Müller, and Bernd Gärtner, *Optimization of convex functions with random pursuit*, SIAM Journal on Optimization, 23 (2013), pp. 1284–1309.

[20] V. Torczon, *On the convergence of the multidirectional search algorithm*, SIAM Journal on Optimization, 1 (1991), pp. 123–145.

[21] V. Torczon, *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, 7 (1997), pp. 1–25.