Exploring Algorithms for Derivative Free Optimization

Livvi Bechtold, Elise Reed

May 3, 2018

1 Introduction

The main goal of this report is to explore different algorithms for derivative-free optimization. We focus most heavily on the Step-size Approximation in Randomized Search (STARS) algorithm presented by Ruobing Chen and Stefan M. Wild in 2015. Two other derivative-free methods are then compared to STARS. The paper is organized as follows.

First, an overview of the purpose and first methods of derivative-free optimization is given. The need for derivative-free optimization is described and how early ideas for generic optimization methods were improved to become useful in today's applications. One such algorithm is presented later. This section also includes the change in direction to stochastic methods and gives the basic framework of two of the other algorithms presented in the following section.

In Sections 3 and 4 three published derivative-free methods are reviewed. For each algorithm, its pseudo-code is given and details about its implementation are explained. The first two algorithms, STARS and RG, use approximation to the derivative in a random direction. The last algorithm presented, RP, also uses a random direction but instead does a line search in that direction. Following these descriptions, a number of numerical experiments comparing the algorithms previously described are presented.

In [1], one experiment presented compares STARS and RG on the Nesterov test problem for the case of uniform random noise, measured as accuracy vs. number of function evaluations. First in Section 5 is the reproduction of the results presented by Chen and Wild (this ensures accuracy of implementation of the algorithms).Then these experiments are also run using a different test problem. This ensures the behavior presented by Nesterov or Chen and Wild are not specific to their chosen test problem.

2 Overview of Derivative-Free Optimization

The need for derivative-free methods in optimization has become more pronounced over the last decade or so as models have become more complicated and computational methods have changed. A common situation in applications is to have a black-box representation of the objective function (e.g. the solution to a PDE) with no practical derivative information available. Another example is when the objective function is nonsmooth - in this case the

necessary derivatives may not exist. In general, even when a function is differentiable, it may be intractable to calculate or evaluate the derivative. These issues motivate the rather new field of derivative-free optimization.

Derivative-free algorithms have mostly been designed for unconstrained problems, which is the type of problem we are exploring in this paper. For this reason, we will restrict our attention to unconstrained optimization problems of the form

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

and are interested in algorithms that converge globally to a stationary point. Algorithms that do generate sequences of iterates that approach stationary points asymptotically will share a general framework. Of this general framework the only part that is relevant to the algorithms we explore is how they choose the step-size. We expound on how this step-size is chosen when we detail each algorithm.

Derivative-free methods were some of the first explored in the field of optimization. The difficulty in their theoretical analysis prevented much progress, and analysis that was completed showed the methods were far less efficient than more mainstream optimization methods. One line of improvements was made by considering *random* derivative-free schemes. The initial idea was to, given a point x, sample the nearby area for a point y and evaluate the function there. If $f(y) < f(x)$ then move to point y, and repeat the process. While this scheme was found to be very inefficient, with proper modifications can be made to speed up the method. For example, the Random Pursuit [5] algorithm combines this random sampling with a line search.

A different line of improvements, first introduced by [4] in 1987, instead uses a random direction u to calculate a finite difference approximation for the directional derivative of the objective function. This mimics derivative-based methods such as gradient descent with an object (i.e. the directional derivative) that behaves much nicer and is easier to calculate than the gradient. The scheme takes the form

$$x_{k+1} = x_k - h_k \frac{f(x_k + \mu_k u) - f(x_k)}{\mu_k} u$$

where h_k is a step-size, μ_k is a so-called 'smoothing' step-size and $\mu_k \to 0$, and u is a random direction. The difficulty of course is determining which parameters h_k, μ_k will force the scheme to converge. One of the first people to determine these parameters and provide a detailed convergence analysis for this type of method was Yurii Nesterov in 2011. It is compared to similar algorithm Step-size Approximation in Randomized Search [1] that was introduced in 2015.

3 Step-size Approximation in Random Search (STARS)

3.1 Introduction to Gaussian Smoothing

Nesterov was the first to recognize that schemes like those introduced by Polyak, above, allow the algorithm to rely on a Gaussian-smoothed version of the objective function

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

where $\mu > 0$ is the smoothing step-size described above and $u \neq 0$ is a random direction with each component drawn from $\mathcal{N}(0,1)$, so each component is i.i.d. We will also define g_μ, the forward finite-difference approximation of the derivative of $f(x)$ in the direction u.

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

Lemma 2.1 in [1] states the following results about g_μ and f_μ for a normally distributed Gaussian vector $u \in \mathbb{R}^n$,

a) If f is convex, then $f_\mu(x) \geq f(x) \; \forall \; x \in \mathbb{R}^n$.

b) If f is convex and continuously differentiable with a gradient that is Lipschitz continuous with Lipschitz constant L_1, then $|f_\mu(x) - f(x)| \leq \frac{\mu^2}{2} L_1 n \; \forall \; x \in \mathbb{R}^n$.

c) If f is differentiable at x, then $\mathbb{E}_u[g_\mu(x)] = \nabla f_\mu(x)$.

The results given above motivate the utilization of random directions in an optimization algorithm. Random directions to approximate the directional derivative of the noisy objective function are used in STARS and RG, below. RP on the other hand has a form much more similar to the first derivative-free method presented in 2, where we check if a new function value is smaller than the current function value. This is done by choosing a random search direction to perform a line search in and then evaluating the function.

3.2 The Algorithm

Algorithm 1: STARS

1. Input initial point x_1, iteration limit N, and step-sizes h_k. Evaluate $\tilde{f}(x_1; \xi_0)$. Set $k = 1$.
2. Generate random Gaussian vector u_k, compute smoothing step-size μ_k.
3. Evaluate $\tilde{f}(x_k + \mu_k u_k; \xi_k)$.
4. Call the stochastic gradient free oracle

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

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

The Step-size Approximation in Randomized Search (STARS) method was introduced by Ruobing Chen and Stefan M. Wild in 2015, and is a variation on the general random derivative-free method introduced in Section 2. In this method, a noise-adjusted smoothing step-size is used in order to minimize the least-squares error between the true directional

derivative of the noisy function and its finite-difference approximation. The directional derivative will always be taken in a random direction u, where each component of u is drawn independently from $\mathcal{N}(0,1)$ (Assumption 3.1).

When using this method, it assumed that the objective function is continuously differentiable and convex. Its also assumed that the gradient is Lipschitz continuous with Lipschitz constant L_1, although direct evaluation of the derivative is impossible. When a function has these three properties we say it satisfies Assumption 4.1 STARS is equipped to handle both additive and multiplicative noise, but for simplicity of comparison with other algorithms only additive noise will be used in this paper. It is assumed that this noise has zero mean and bounded variance. For our current discussion, this noise will be a random number ξ_i drawn uniformly from $[-3\sigma, 3\sigma]$. In Section 5 stochastic noise with different distributions is explored.

It is shown in [1] that when the above assumptions are met (convex, continuously differentiable, etc.), there is an optimal smoothing step-size

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

for which the expectation of the least-squares error

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

between the finite-difference approximation and the directional derivative is no more than $\sqrt{2}L_1\sqrt{n(n+6)^3}$. This step-size is independent of x as long as the variance of the noise is constant.

3.3 Convergence Analysis for Additive Noise

Lemma 3.1. *Let assumptions 3.1, 4.1, and 4.2 hold. If the smoothing stepsize μ_k is set to the contant μ^* from 4.2, then Algorithm 1 genearates steps satisfying*

$$\mathbb{E}[\|S_{\mu_k}\|^2] \leq 2(n+4)\|\nabla f(x_k)\|^2 + C_2 \tag{1}$$

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

Theorem 3.2. *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*

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

Using lemma 3.1 and theorem 3.2 as shown above, Chen and Wild also present a convergence analysis for the case of additive noise. They show that, for target accuracy ϵ, Algorithm 1 will converge for noise with variance σ satisfying $\sigma \leq \frac{5\epsilon}{6\sqrt{2}(n+4)}$. It is further shown

that this accuracy is met in $N = \frac{8(n+4)L_1R^2}{\epsilon}$, where we can notice that $N \sim \mathcal{O}\left(\frac{n}{\epsilon}L_1R^2\right)$. Then if we are given an optimization problem that has bounded absolute noise, of variance σ_a^2, then the STARS algorithm gives a best possible accuracy of

$$\epsilon_{pred} \geq \frac{6\sqrt{2}\sigma_a(n+4)}{5}, \tag{3}$$

then a solution to our problem can be obtained in $\mathcal{O}\left(\frac{n}{\epsilon_{pred}}L_1R^2\right)$.

4 Two Other Algorithms

4.1 Random Search for Smooth Optimization (RG)

Algorithm 2: RG

1. Input initial point x_0 and iteration limit N. Fix step-size $h = \frac{1}{4(n+4)L_1}$ and compute smoothing step-size μ_k, based on target accuracy $\epsilon = 2^{-16}$. Set $k = 1$.
2. Generate random Gaussian vector u_k.
3. Evaluate $\tilde{f}(x_k; \xi_k)$ and $\tilde{f}(x_k + \mu_k u_k; \xi_k)$.
4. Call the 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 $x_{k+1} = x\_hs_\mu(x_k; u_k, \xi_k)$. Update $k = k + 1$. Return to Step 2.

The Random Search for Smooth Optimization (RG) method was introduced by Yurii Nesterov in 2011, and is the one of the first variations on the general random derivative-free method introduced above. Nesterov was the first to introduce complexity bounds and convergence analysis for these random zero-order methods, zero-order meaning we evaluate only the function and not any of its derivatives. In the RG method, the smoothing step-size is chosen based on a target accuracy, the dimension on the problem, and the Lipschitz constant of the objective function. This is the main difference from STARS, as RG also uses Gaussian random directions in its gradient-free oracle.

When using this method, it is assumed that the objective function is continuously differentiable and convex. In addition, it is also assumed that the gradient is Lipschitz continuous with Lipschitz constant L_1. The convergence rates given in [3] are found using an objective function without noise. In Section 5 we explore how RG performs with a noisy objective function. As with our STARS tests, this noise will be additive and generated from various distributions.

It is shown in [3] that choosing the smoothing step-size

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

allows the problem to be solved within the target accuracy ϵ. Further, Nesterov shows this target accuracy can be achieved in $N \sim \mathcal{O}\left(\frac{n}{\epsilon}L_1R^2\right)$ iterations. Notice this is the same convergence rate achieved by STARS when using a noisy objective function. This indicates that the smoothing step-size chosen in STARS allows the algorithm to perform as well on noisy functions as RG does on functions without noise.

4.2 Random Pursuit (RP)

Algorithm 3: RP

1. Input initial point x_0, iteration limit N, and line search accuracy $\mu = 0.0025$. Set $k = 1$.
2. Generate random Gaussian vector u_k.
3. Set $x_{k+1} = x_k + \text{LS}_{\text{APPROX}_\mu}(x_k, u_k) \cdot u_k$. Update $k = k + 1$. Return to Step 2.

$$\text{LS}(x, u) \in \arg\min_{h \in \mathbb{R}} f(x + hu)$$

$$\text{LS}(x, u) - \mu \leq \text{LS}_{\text{APPROX}_\mu}(x_k, u_k) \leq \text{LS}(x, u) + \mu$$

The Random Pursuit (RP) method was introduced by S. U. Stich, C. L. Muller, and B. Gartner in 2012. This method differs most from the other two in that instead of calling a stochastic gradient-free oracle, a line search oracle is called. Thus a random direction must still be generated, but now we will move along the function in that direction directly instead of the taking the derivative in that direction.

When using this method, it is assumed that the objective function is differentiable and convex. Furthermore, it should have bounded curvature in the following sense: $f(y) - f(x) - \langle \nabla f(x), y - x \rangle \leq \frac{L_1}{2}||x - y||^2$ for all $x, y \in \mathbb{R}^n$. I observe that the constant L_1 is only needed for theoretical results and is in fact not needed for implementation. This is a clear benefit over STARS and RG as they require knowledge of the Lipschitz constant of the gradient of the objective function, which can be difficult if not impossible to calculate and may not be easy to estimate. The convergence results given in [5] are found using an objective function without noise. In Section 5 the performance of RP with an objective function with additive noise of various distributions is tested.

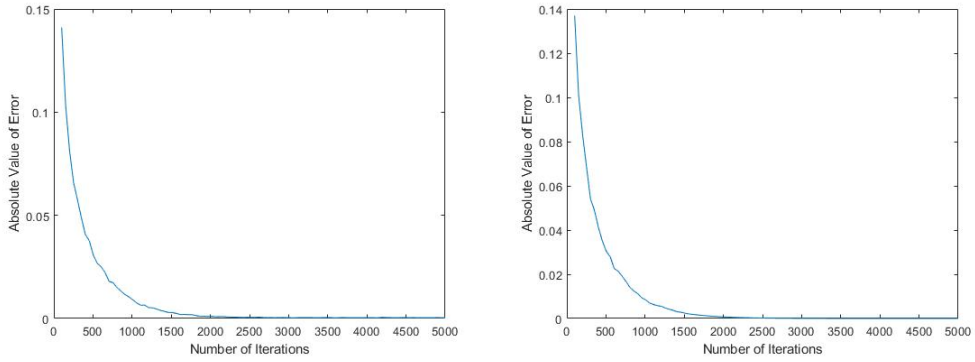
It is shown in [5] that a target accuracy ϵ can be achieved in $N \sim \mathcal{O}(\frac{n}{\epsilon})$ iterations. This occurs when the line search oracle mentioned above uses target accuracy $\mu = .0025$. Notice that this on the order of the same number of iterations as RG and STARS.

5 Numerical Experiments

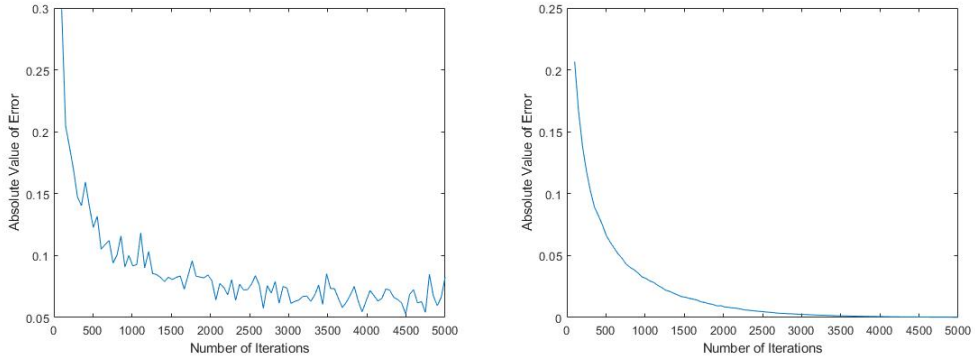
5.1 Nesterov Test Function

To ensure the algorithms are working correctly, our first goal was to reproduce the results given in Section 6.1 of [1]. Here, the authors run experiments to test the performance of STARS and RG in the sense of accuracy vs. number of function evaluations. This is done for additive noise drawn from a uniform distribution, using standard deviations 10^{-6} and 10^{-3}. They present the mean of 20 random seeds, and the same is done here. Note that number of iterations is used instead of number of function evaluations here for simplicity. Each trial uses N iterations, and each iteration uses 2 function evaluations. Therefore we test $N = 100$ iterations to $N = 5,000$. The same behavior of RG and STARS as was found in [1] is observed below.

STARS: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$

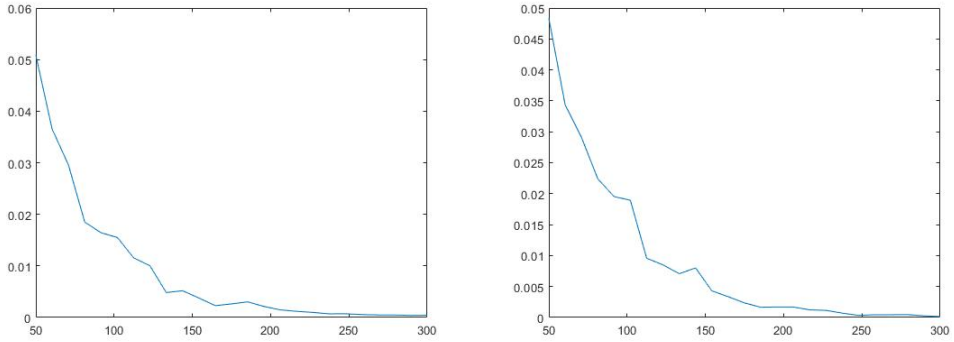


RG: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$



Since the Random Pursuit (RP) algorithm is also considered in this paper, the above analysis was completed for RP. It was seen in preliminary experiments that RP achieves its maximum accuracy in less iterations than RG or STARS. It should be noted, however, that the computations are much more expensive in RP: It takes 5-15 minutes on my machine to run 20 trials of STARS or RG for each N as given above, while it takes RP over 10 hours to do so. For this reason, RP will only be tested between $N = 50$ iterations to $N = 300$ iterations. This is due to the fact that it is difficult to know the number of function evaluations the line search oracle computes for each iteration.

RP: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$



5.2 Sphere Test Function

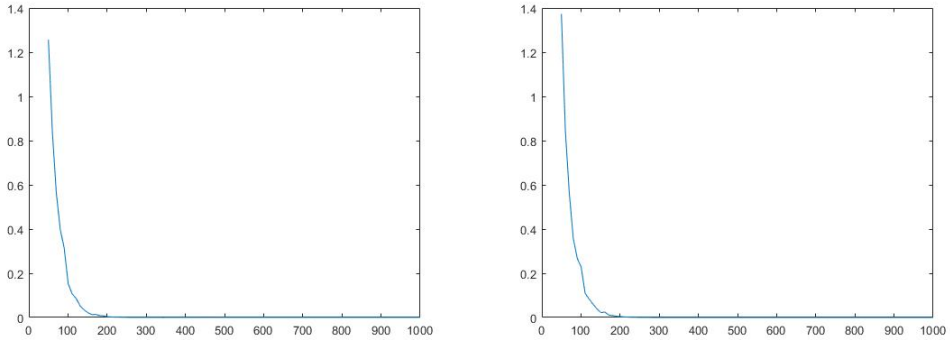
Recall now that we observed that in both papers presenting the RG and STARS algorithms, only the Nesterov test function is used. Therefore, it was thought we could benefit from knowing if these schemes perform well with other test functions as well. Unfortunately, it is difficult to find test functions that meet the assumptions needed for convergence of STARS or RG: the function needs to be continuously differentiable and convex. Further, the gradient must be Lipschitz continuous. One function found that is known [6] in optimization literature is the sphere function

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

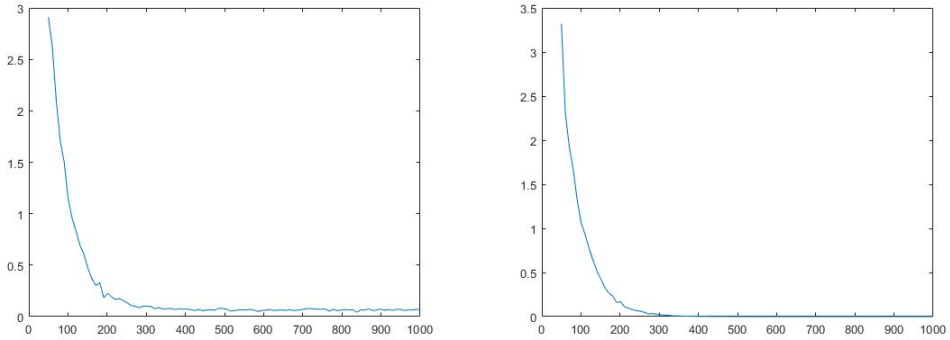
It is certainly continuously differentiable and convex. Further, the gradient is Lipschitz continuous and $L_1 = 4$ is an appropriate estimate to the its Lipschitz constant, as is used for the Nesterov function above. (The Lipschitz constant of this function is in fact 2, which shows that STARS and RG do not need a very accurate input for the Lipschitz constant. This is very advantageous as exact Lipschitz constants can be very difficult to calculate or even estimate.)

The behavior seen with this test function is very similar to the Nesterov function. It was found that the algorithms converge much more quickly (less iterations) for the sphere function as opposed to the Nesterov function. For this reason, the trials for STARS and RG on the sphere test function will only run number of iterations between $N = 50$ and $N = 1,000$ iterations.

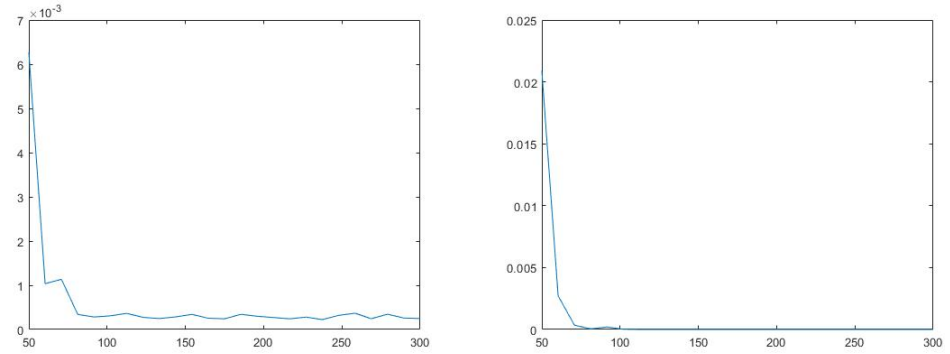
STARS: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$



RG: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$

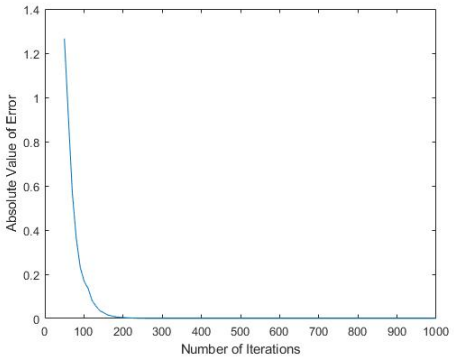


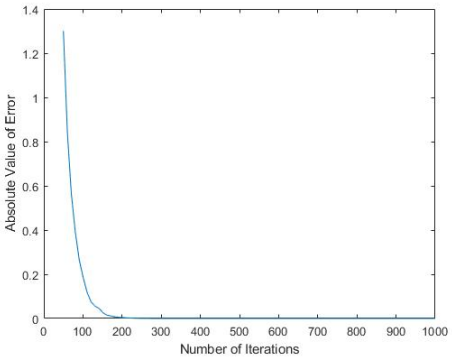
RP: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$



In fact, STARS performs so well with the sphere test function that we decided to explore how large of a variance it would still converge for. The results are seen below. In order to see the same smoothness in plotting accuracy vs. number of function evaluations, 100 random seeds were used instead of the 20 used above.

STARS: $\sigma_1 = 1$, $\sigma_2 = 10^{-1}$





6 Conclusion

To summarize, this paper has given an overview of derivative-free optimization, a review of three random derivative-free methods, and a results from various numerical experiments testing the performance of the algorithms in the sense of accuracy vs. number of iterations.

Derivative-free methods have become popular recently for a number of reasons. Some of these include lack of availability of a derivative and difficulty in evaluating a derivative. In order to combat these difficulties, zero-order schemes were explored. We focus on the idea of random search, and discuss developments for implementing random search with both line search and derivative estimation.

The specific algorithms explored in this paper were Stepsize Approximation in Randomized Search (STARS), Random Search for Smooth Optimization (RG), and Random Pursuit (RP). The first two use the technique of generating a random direction and then estimating the gradient via calculating the directional derivative of the noisy objective function in that random direction. STARS uses a noise-adjusted smoothing stepsize, while RG uses a smoothing stepsize for a targeted accuracy. RP on the other hand generates a random direction and then performs a line search in that direction. The new iterate is accepted in the value of the noisy objective function is smaller than that of the previous iterate, and rejected otherwise.

Section 5 was comprised of two sets of numerical experiments. The first of these is a reproduction of the results given by Chen and Wild in [1] in order to confirm the algorithms have been implemented accurately. Then, we expand the accuracy experiments given in [1] by considering a different test problem.

We have found that STARS, RG, and RP can converge when given an objective function contaminated with additive uniform noise. This is impressive as neither RG nor RP were designed to be used on noisy objective functions. It was also found that this behavior is not unique to the Nesterov test function used by Nesterov, and Chen and Wild. In fact, all three algorithms converged very quickly (in the sense of number of iterations) for the sphere test function. This is likely due to the simplicity of this test function. We were interested to see if STARS would still converge when we increased the variance past the theoretical

upper bound presented by Chen and Wild. Interestingly, STARS will converge for variance as large as one.

References

[1] Chen, Ruobing, and Stefan M. Wild. "Randomized Derivative-Free Optimization of Noisy Convex Functions". *U.S. Department of Energy, Office of Science*, 12 July 2015.

[2] "Introduction." *Introduction to Derivative-Free Optimization* , by Andrew R Conn et al., 2009, pp. 1-–12. MOS–SIAM Series on Optimization.

[3] Nesterov, Yurii, and Vladimir Spokoiny. "Random Gradient-Free Minimization of Convex Functions." *Foundations of Computational Mathematics*, vol. 17, no. 2, 2015, pp. 527—566., doi:10.1007/s10208–015–9296–2.

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

[5] Stich, S. U., et al. "Optimization of Convex Functions with Random Pursuit." *SIAM Journal on Optimization*, vol. 23, no. 2, 2013, pp. 1284—1309., doi:10.1137/110853613.

[6] Surjanovic, Sonja, and Derek Bingham. "Virtual Library of Simulation Experiments:" *Optimization Test Functions and Datasets*, Simon Fraser University, 2013, `www.sfu.ca/~ssurjano/optimization.html`.

[7] `https://stats.stackexchange.com/users/919/whuber`, whuber. "What Is the Variance of the Weighted Mixture of Two Gaussians?" *What Is the Variance of the Weighted Mixture of Two Gaussians?*, 20 May 2015, `stats.stackexchange.com/questions/16608/what-is-the-variance-of-the-weighted-mixture-of-two-gaussians?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa`.