ISSO Project Report: Conditions and Modifications for Convergence of Random Search under Noisy Measurements

Yu-Hsin Lee

April 2024

Abstract

Random search is known for its simplicity yet effectiveness in a wide range of problems, particularly in global optimization. The general idea of it is simple: from a domain, the estimated parameter is repeatedly selected and accepted, if a lower new loss function is obtained, sometimes according to some probability. However, it is an algorithm more suited for optimization without noise: convergence with noisy measurements is greatly limited. In this project, the conditions under which random search would achieve global optimum with noisy measurements will be consolidated, together with the possible modifications to the algorithm to increase its effectiveness in this area.

Introduction

Random search is a simple zero-order search algorithm for noise-free loss measurements that only uses loss function $L(\theta)$ values. In *Introduction to Stochastic Simulation* [Spa03], three different random search algorithms A, B and C are proposed. Algorithm A, or simple random search, comes in the simplest form: according to a distribution, randomly generate $\hat{\theta}$ estimates and calculate their loss, updating $\hat{\theta}$ if the new loss is lower than before. Algorithm B and C, or localized (enhanced) random search, are similar algorithms that restrict random search to the neighborhood of current $\hat{\theta}$.

Minimizing the loss function can be represented as finding the set:

$$\Theta^* = \text{arg min}_{\theta\in\Theta} L(\theta) = \{\theta^* \in \Theta : L(\theta^*) \leq L(\theta) \quad \forall\theta \in \Theta\}$$

In the case of local optimization, we find this set over a localized domain, i.e., the vicinity or neighborhood of the local minimum θ^*, and the domain Θ in this case would be $(\theta^* - \varepsilon, \theta^* + \varepsilon)$; but for global optimization, we aim to find this set over the entire domain of the problem setting. In other words, we want to find the minimum of all local minima $\theta_1^*, \cdots, \theta_n^*$.

Random search has been shown to converge to the global optimum under a few conditions on the loss function, albeit more restrictive in conditions for localized random search algorithms like B and C. Below I describe a version of algorithm A that is used in most of the applications in the project. All code can be found at: Github

Algorithm A

Step 1: Set $k = 0$. Choose an initial value $\hat{\theta}_k \in \Theta$.

Step 2: Generate a new independent value $\theta_{new} \in \Theta$, according to a uniform distribution on Θ. If $L(\theta_{new}) < L(\hat{\theta}_k)$, set $\hat{\theta}_{k+1} = \theta_{new}$; else set $\hat{\theta}_{k+1} = \hat{\theta}_k$.

Step 3: Repeat step 2-3 until maximum number of iterations.

Non-convergence of Random Search

It is easy to show the non-convergence of random search algorithms for noisy measurements. Consider the domain $[0, 1]$ and the noisy measurement $y(\theta) = L(\theta) + \varepsilon(\theta)$, where $L(\theta) = \frac{\theta}{2}$ and $\varepsilon(\theta) \sim \text{Uniform}(-\theta, \theta)$ (Figure 1). In this case, the errors are not identically

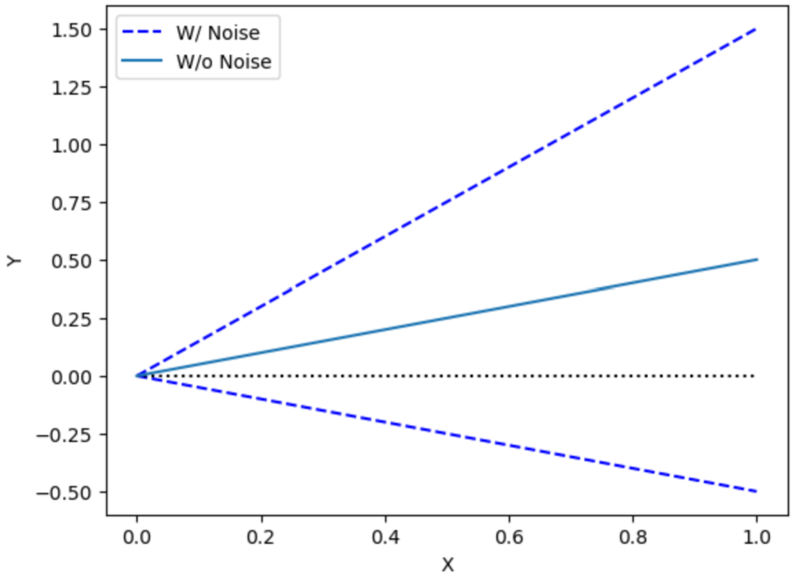

Figure 1: Example of Non-Convergence: $y(\theta) = \theta/2 + \text{Uniform}(-\theta, \theta)$

distributed. The minimum of the loss function is obtained at $\theta^* = 0$ where $L(\theta^*) = 0$, but the range of the noisy loss function extends beyond the minimum of the original. If random search chooses any point below the black dotted line $Y = 0$, then random search will not converge to θ^*; in fact, it converges to $\hat{\theta}^* = 1$ with minimum loss of -0.5.

It is important to note that the type of noise is crucial to the convergence or non-convergence of random search. Devroye and Krzyzak (2002) proposed some conditions for the noise to ensure convergence, and we will discuss it below.

1 Devroye and Krzyzak (2002) [DK02]

Devroye and Krzyzak proposed an idea of ”stable” noise to ensure convergence. Suppose we observe $y_n(\theta_n) = L(\theta_n) + \varepsilon_n$, where ε_i's are iid random variables distributed as ε. Then, the noise will be called stable if for all $\delta > 0$,

$$\text{lim}_{x\downarrow-\infty}\frac{G(x-\delta)}{G(x)} = 0$$

where G is the distribution function of ε, and $0/0$ is considered as zero. A sufficient condition for stability is that G has a density g and

$$\lim_{x \downarrow -\infty} \frac{g(x)}{G(x)} = \infty$$

As for examples of such distribution functions, they stated that if G does not have a left tail (i.e. $G(x_0) = 0$ for some $x_0 > -\infty$), then the noise is stable. Normal noise and exponential noise are stable, but Laplace noise is not. If ε is stable, then $L(\theta_n^*) \to L(\theta^*)$ in probability.

Furthermore, they stated that G is strongly stable if the minimal order statistic of $\varepsilon_1, \ldots, \varepsilon_n$ is strongly stable, i.e., there exists a sequence of numbers a_n such that $\varepsilon_n - a_n \to 0$ almost surely as $n \to \infty$. If ε is strongly stable, then $L(\theta_n^*) \to L(\theta^*)$ almost surely.

Brief Analysis

Proof can be found in the paper, but here is my interpretation: Since the errors do not have a left tail, there is a negligible probability that a noisy measurement could be outside a set neighborhood of values around the *true* loss function. Because the errors are iid, this neighborhood of values around the minimum would still contain the minimum of the *noisy* loss function, hence random search would eventually converge to the minimum in probability.

Application

For the rest of the project: consider the loss function $L(\theta) = 0.10(1 - u_1)^2 + 0.90(1 - 0.5u_1u_2 - u_2)^2 + 1$ where $\theta = (u_1, u_2)$. The minimum of this function occurs at $\theta^* = (1, 2/3)$, with $L(\theta^*) = 1$. Now suppose we observe noisy measurements $y(\theta_k) = L(\theta_k) + \varepsilon_k$ with iid ε_k following Normal$(0, 1)$ (Figure 2), Exponential(1) (Figure 3) and Laplace$(0, 1)$ distributions (Figure 4). Setting $\hat{\theta}_0$ as $[2, 2]$, noisy measurements are used in algorithm A, while the plots are obtained taking the mean of the true loss function differences and

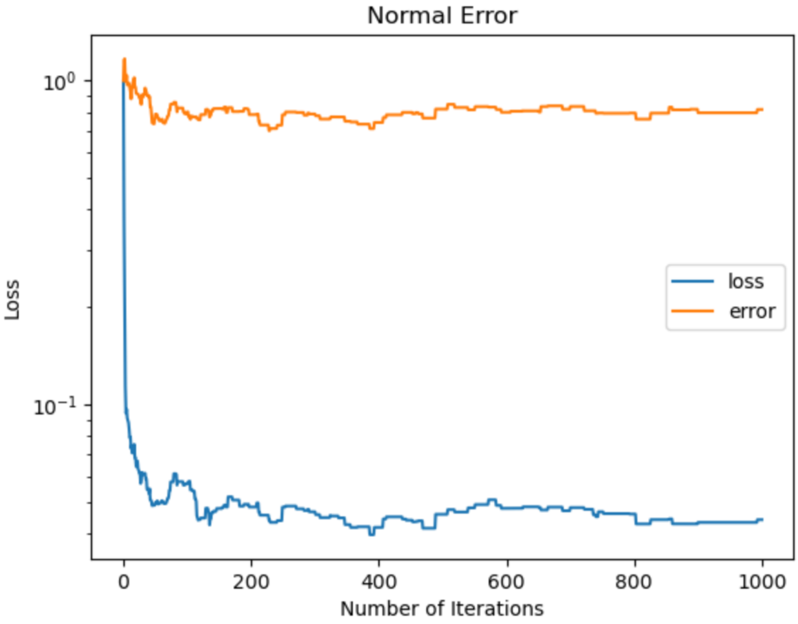

Figure 2: Stable Error: Normal Distribution

norm of parameter estimate errors (vs the minimum) over 40 replicates, then normalized.

Note that it is proven that the loss function converges (as shown in the normal and exponential errors), but not necessarily the parameter estimate $\hat{\theta}$. We observe that for the Laplace distributed error, the loss does not seem to converge. The convergence for normal error is not completely convincing, however, we would see clear convergence later on using algorithm B instead.

Modifications to Random Search for Noisy Measurements

As with any kind of algorithm, changes can be made to cater to specific problems. However, in order to do this, tradeoffs have to be made, often in terms of time and cost. We first look at two algorithms proposed in Spall (2003), Alexander et al. (2005), then Gelfand and Mitter (1989), in addition to a new proposed algorithm; and finally, briefly touch upon Yakowitz and Fisher (1973), Zhigljavsky and Pinter (1991).

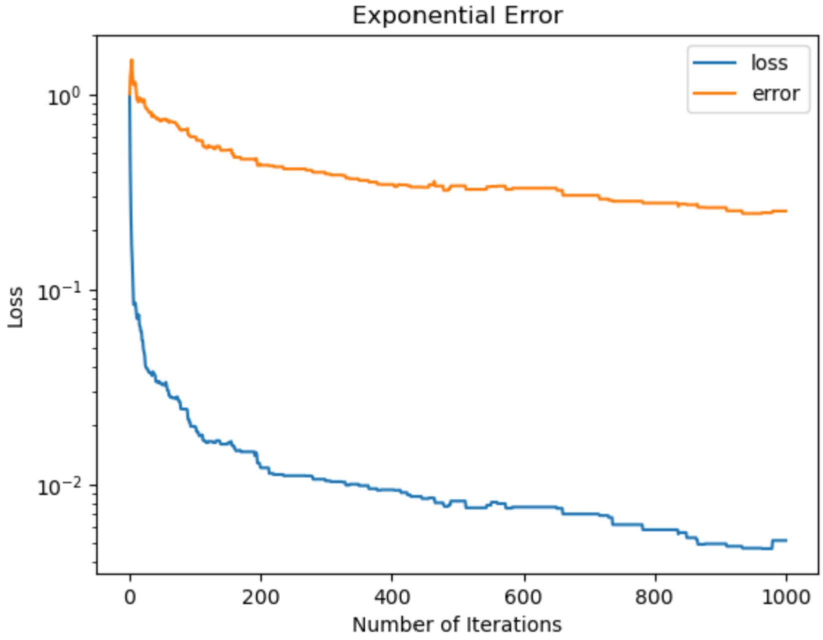

Figure 3: Stable Error: Exponential Distribution

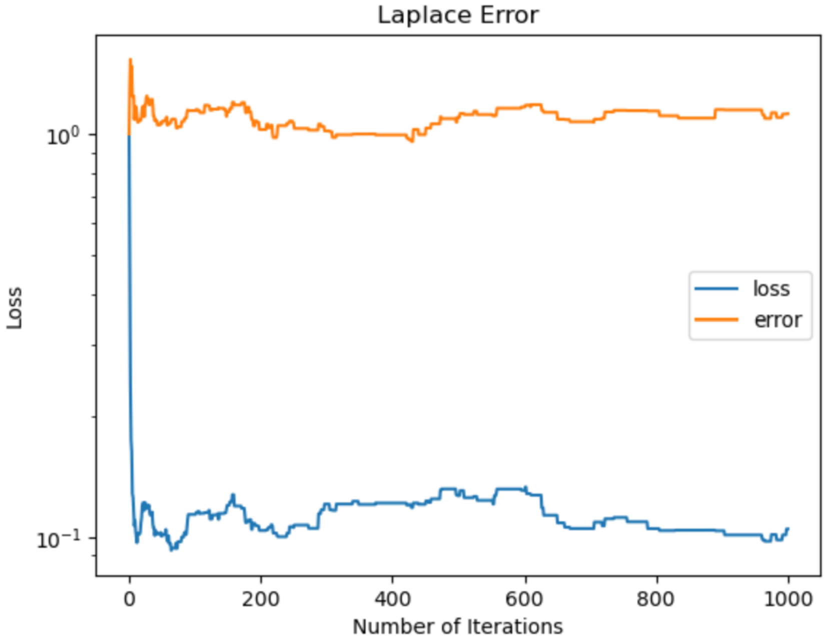

Figure 4: Unstable Error: Laplace Distribution

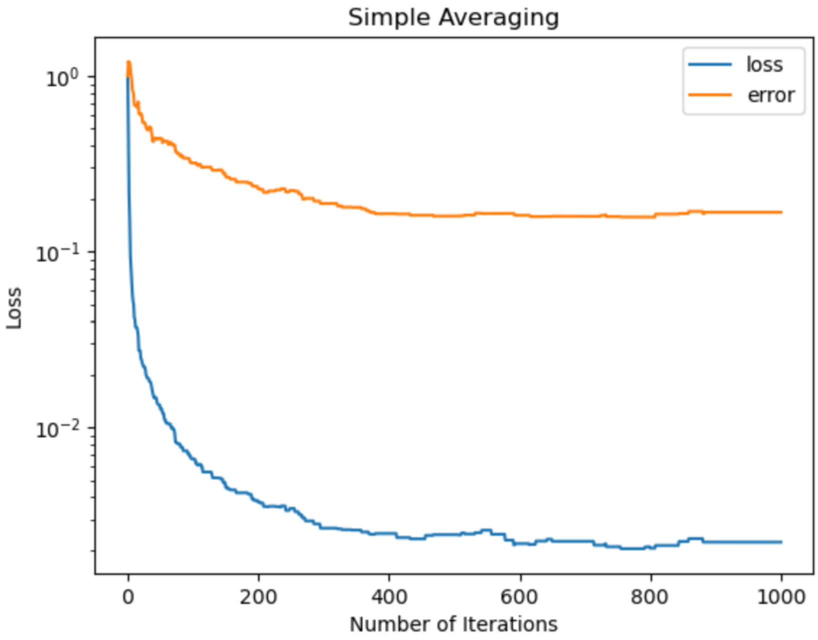

Figure 5: Simple Averaging: Normal Distribution

2 Simple Averaging [Spa03]

Under iid noise with bounded mean, We can collect several evaluations of the noisy loss measurement $y(\theta)$ at each value of θ and average these values. If the number of evaluations at each value is large enough, we can treat the average value as a perfect measurement, if the noise has mean 0. Otherwise, iid noise will still allow us to get to the minimum, as the noisy loss would simply be shifted upwards or downwards.

2.1 Application

Consider the same noisy loss function and same setup (algorithm A) as before. At each step, we take the mean of 1000 noisy loss measurements, for Normal$(0, 1)$ (Figure 5), Weibull$(1, 5)$ (Figure 6) with a non-zero $\Gamma(1 + 1/5)$ mean and Laplace$(0, 1)$ (Figure 7) with heavy tails.

We note that in this case, both $L(\hat{\theta})$ and $\hat{\theta}$ converge, comparing to Figures 2-4. In particular, for averaging with Laplace error, both converge in spite of its heavy tails (Devroye and Krzyzak). The downside of the averaging method is the cost: for Laplace

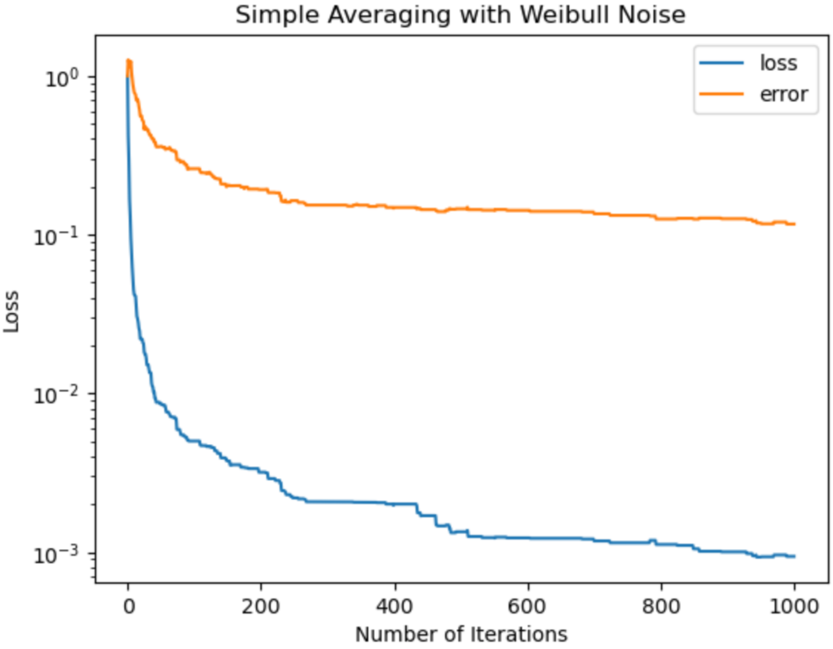

Figure 6: Simple Averaging: Weibull Distribution

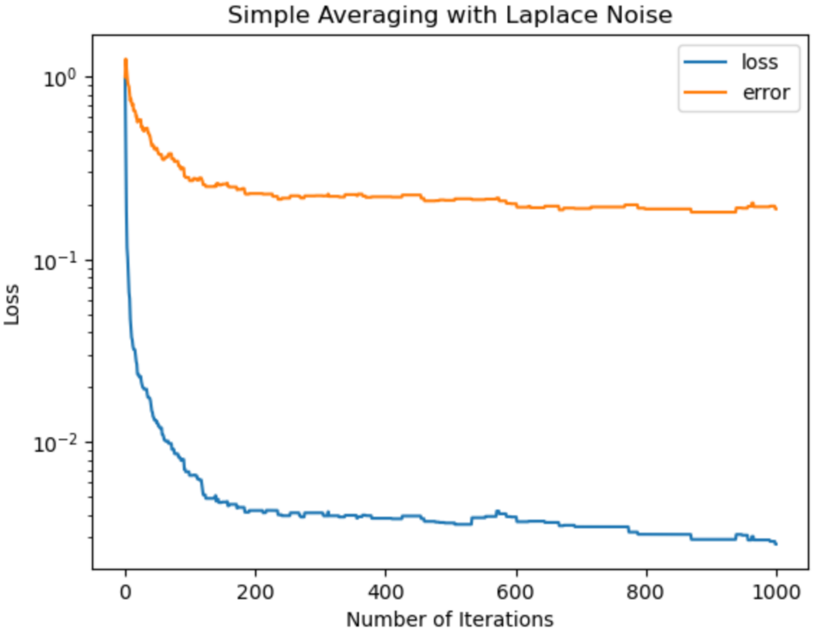

Figure 7: Simple Averaging: Laplace Distribution

error, $\hat{\theta}$ does not converge with 10 measurements at each step; a lot more measurements are needed for convergence.

3 Alexander et al. (2005) [ABC$^+$05]

In the same vein, Alexander et al. proposed what they call ”Increasing sample-size acceptance-rejection search” (ISSARS) given a sequence of positive integers N_k. δ is some sampling distribution.

Step 1: Set $k = 0$. Generate $\hat{\theta}_k \sim \delta$

Step 2: Generate a point $z \sim \delta$

Step 3: Take samples of noisy measurements y of size N_k at $\hat{\theta}_0$ and z, and compute the means $M_{\hat{\theta}_k}$ and M_z.

Step 4: If $M_{\hat{\theta}_k} < M_z$, then set $\theta_{k+1} = z$. Otherwise set $\theta_{k+1} = \theta_k$.

Step 5: Increment k and return to step 2.

They stated that if the error variances σ_θ^2 are uniformly bounded, the sequence of points generated by ISSARS converges to the global optimum with probability one.

3.1 Brief Analysis

As the number of iterations increases, the number of noisy measurements increases. We can afford to have a cruder estimate of initial loss, but as we get closer to the minimum, we need a more accurate estimate of the true loss.

3.2 Application

Under Normal$(0, 1)$ (Figure 8) errors, and increasing $N_k = \{1, 2, 3, \ldots\}$, ISSARS applied to algorithm A achieves convergence in $L(\hat{\theta})$ and $\hat{\theta}$, similar to the simple averaging method in Section 2. ISSARS has considerably slower convergence, however, and this is likely due to the selection of N_k. The sequence N_k is considerably difficult to tune (a guideline

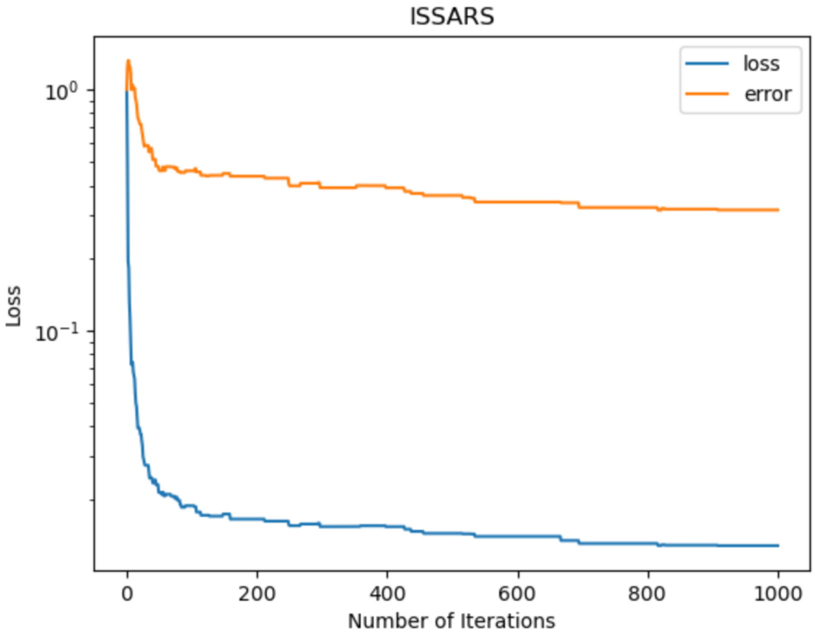

Figure 8: ISSARS

was not provided in the paper): it is hard to balance between the cost (the number of measurements) and the rate of convergence.

4 Constant Threshold [Spa03]

Another method is to allow the algorithm to resist changing the best current estimate unless the new loss is significantly lower. In other words, we accept the new estimate only if $y(\theta_{new}(k+1)) < y(\hat{\theta_k}) + \tau_k$ for $\tau_k > 0$. Spall suggested that τ_k can be viewed as the number of standard deviations in the measurement noise that the new measurement must improve on the old measurement before θ will be changed, particularly under approximate normal distribution. Thus, a good guideline to choosing τ_k, is the number of $\sigma-$improvements we desire.

4.1 Application

Using Normal$(0, 1)$ (Figure 9) errors, we set τ_k as 1, for a one-sigma improvement in measured loss value. We note that convergence is noticeably slower: the higher τ_k is, the more conservative the acceptance rule is.

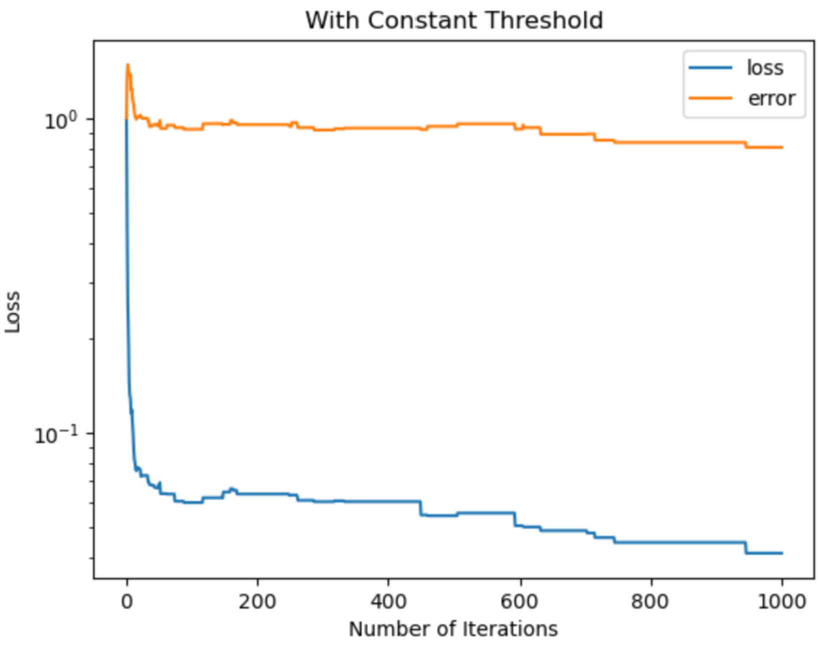

Figure 9: Constant Threshold

Simulated Annealing Inspirations

Simulated Annealing (SAN) is another global optimization method that is very similar to random search, except it accepts the new $\hat{\theta}$ estimate based on the Metropolis criterion. It can be shown that as the temperature $T \to 0$, SAN becomes random search. Like random search, there is limited global convergence theory for SAN with noisy measurements. Below, we will be looking into a modification for the SAN algorithm that is particularly similar to the constant threshold method, and apply it onto random search.

5 Gelfand and Mitter (1989) [GM89]

Gelfand and Mitter proposed a method introducing a random threshold for acceptance. The simplified version of their SAN algorithm is as follows.

Step 1: Set $k = 0$ Set an initial temperature T and initial estimate $\hat{\theta}_k$.

Step 2: Choose candidate θ_{new} according to some probability distribution.

Step 3: Let $\delta = y(\theta_{new}) - y(\hat{\theta}_k) + W_k$ for some real-valued random variable W_k. Accept θ_{new} and set $\hat{\theta}_{k+1} = \theta_{new}$ if $\delta \leq 0$; else accept with probability $\exp(-\delta/c_b T)$ if $\delta > 0$.

Step 4: Repeat steps 3-4 until end of iteration.

In their paper, they considered the case where $W_k \sim \text{Normal}(0, \sigma_k^2)$ with $\sigma_k^2 > 0$ and showed convergence.

5.1 Brief Analysis

Similar to the constant threshold method in section 4, the modification here uses a (normal) random variable threshold instead. Having a random threshold W_k gives more flexibility in the acceptance rule: if the threshold is negative, there is a higher probability of accepting values that have a higher measured loss, hence possibly exploring more of the other local minima. On the other hand, if the threshold is positive, the acceptance rule is more conservative, thus less exploration.

5.2 Application

In our version of random search there is 0 acceptance probability if the new measured loss is higher. However, perhaps the same concept could be applied to the acceptance probability if the new measured loss is lower. Since SAN has a decreasing temperature parameter that gradually reduces the acceptance probability, in random search, we can increase the variance of random threshold W_k to "decrease the acceptance probability". In other words, step 2 of algorithm A would be as follows: If $L(\theta_{new}) < L(\hat{\theta}_k) - |W_k|$, set $\hat{\theta}_{k+1} = \theta_{new}$ for some $W_k \sim \text{Normal}(0, \sigma_k^2)$ with increasing σ_k^2; else set $\hat{\theta}_{k+1} = \hat{\theta}_k$. The results are shown in Figure 10 with $\sigma_k^2 = 0.3 * 1.002^k$, rather similar to that of constant threshold. Note that as number of iterations increases, the variance will become very large, and the acceptance rule may become too conservative.

6 Proposed Algorithm: Taking the Minimum

We now look at algorithm B, a localized random search algorithm. The steps are as follows.

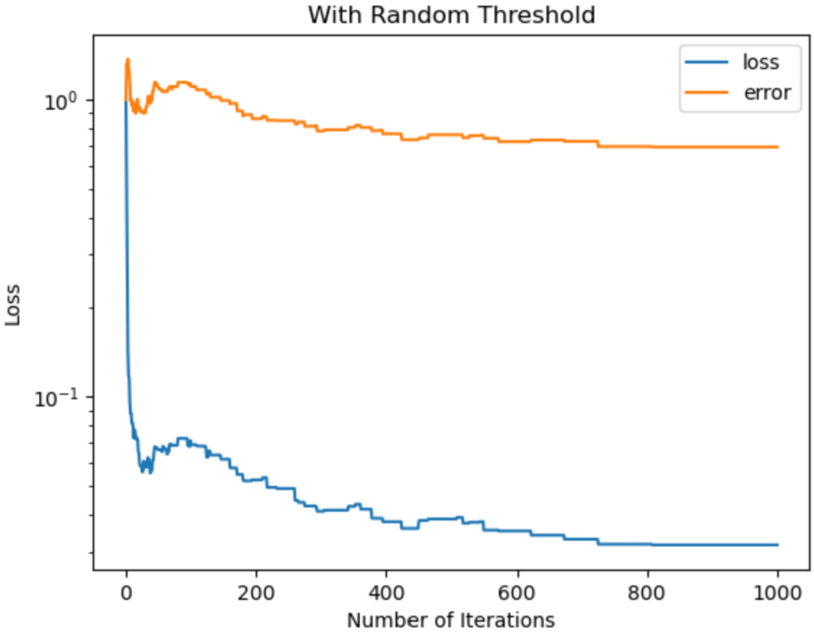

Figure 10: Random Threshold

Step 1: Set $k = 0$. Pick initial estimate $\hat{\theta}_k \in \Theta$.

Step 2: Generate an independent random vector $d_k \in \mathbf{R}^p$ and add it to $\hat{\theta}_k$, accept if $\theta_{new} = \hat{\theta}_k + d_k \in \Theta$, else repeat step 2 until satisfied.

Step 3: If $L(\theta_{new}) < L(\hat{\theta}_k)$, accept and set $\hat{\theta}_{k+1} = \theta_{new}$, else set $\hat{\theta}_{k+1} = \hat{\theta}_k$.

Step 4: Repeat steps 2-4 until maximum number of iterations.

Using the same noisy loss measurement with Normal$(0, 1)$ errors, and setting each term of d_k as iid Uniform$(-0.5, 0.5)$, we obtain Figure 11, with a significant improvement in convergence compared to algorithm A (Figure 2).

6.1 Taking the Minimum Analysis

Under iid noise with non-heavy tails, from section 1 we discussed the idea of minimum *true* loss ultimately having the minimum *measured* loss. Consider a simple loss function $y(\theta) = L(\theta) + \varepsilon$ where $L(\theta) = \theta^2$ in the domain $[-3, 3]$ and $\varepsilon \sim$ Normal$(0, 1)$ in Figure 12. Suppose algorithm B chooses $\theta = -1$, then, subject to noise, the measured loss could take any value between $[-2, 4]$ with a 0.997 probability. Suppose we observe $y(-1) = 1$,

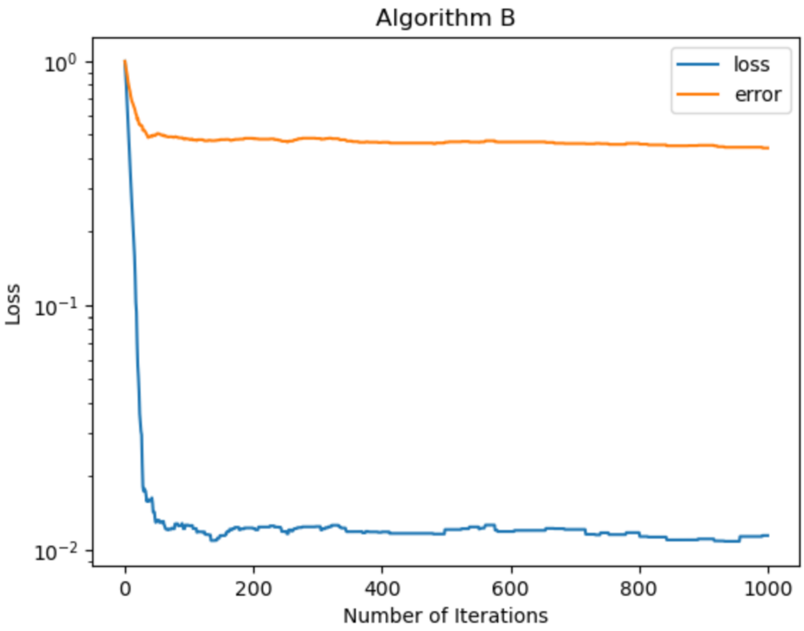

Figure 11: Algorithm B

then algorithm B would accept any value that lies below the horizontal green dotted line: it could possibly accept $\theta < -1$, which is inefficient in getting towards the minimum $\hat{\theta} = 0$. Hence, I propose that in algorithm B, we can take the minimum of a multiple measured losses at an estimate $\hat{\theta}$ at a small probability - to limit the cost as seen in the simple averaging algorithm.

6.2 Application

In this case, step 3 of algorithm becomes: With a certain probability p, take the minimum m_k of n measurements of $L(\theta_{new})$; else, with probability $1-p$, take one measurements of $m_k = L(\theta_{new})$. If $m_k < L(\hat{\theta}_k)$, accept and set $\hat{\theta}_{k+1} = \theta_{new}$, else set $\hat{\theta}_{k+1} = \hat{\theta}_k$. For the same noisy loss with Normal$(0,1)$ errors, I set $p = 0.05$ and $n = 50$ (Figure 13). We can observe a slightly faster convergence rate as compared to the original algorithm B in Figure 11. Convergence theory for this algorithm should be more restrictive than algorithm A, just like other localized random search algorithms such as B and C in *ISSO*.

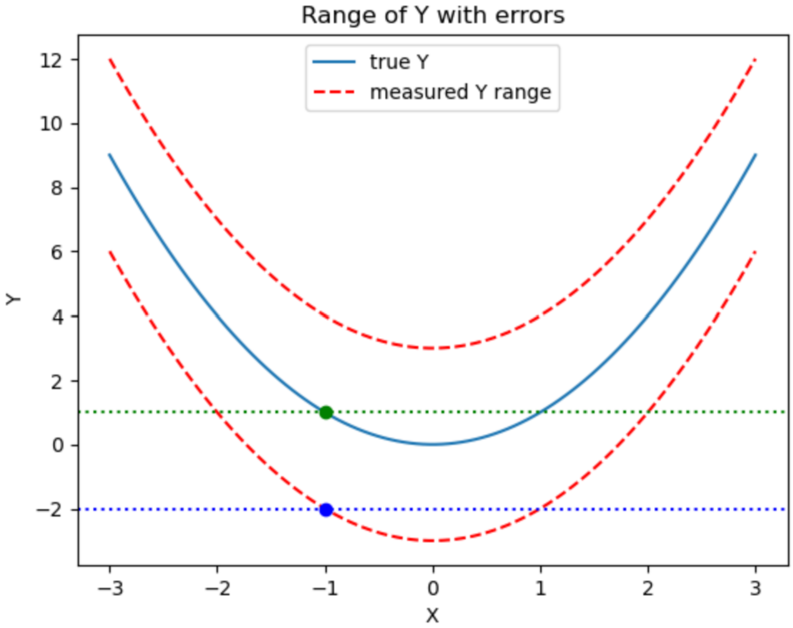

Figure 12: Simple Loss with Normal Noise

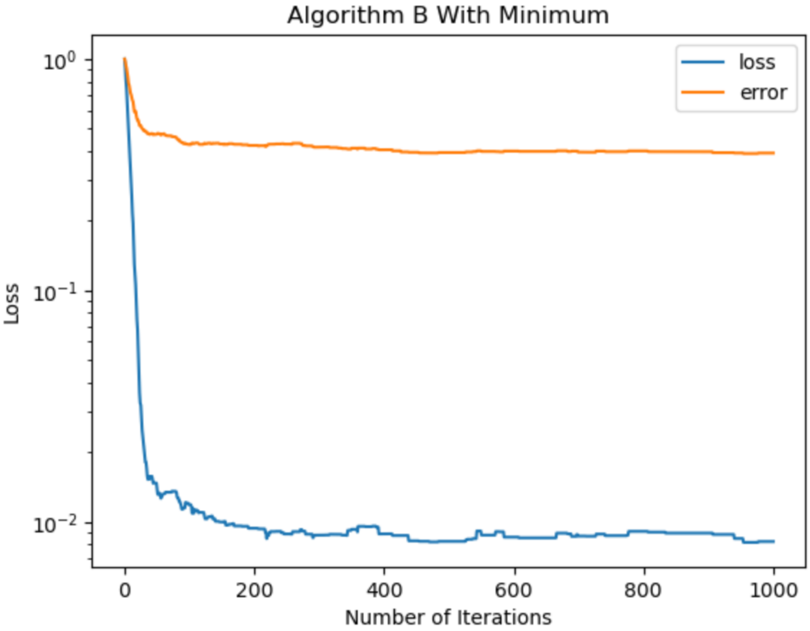

Figure 13: Algorithm B

7 Potential Future Research

Yakowitz and Fisher, Zhigljavsky and Pinter proposed rather complex conditions and modifications to the random search algorithm, which we would briefly summarize.

7.1 Yakowitz and Fisher (1973) [YF73]

Yakowitz and Fisher proposed that when measurement error does not depend on θ_n or $L(\theta_n)$, assuming $\varepsilon_1, \varepsilon_2, \ldots$ iid as random variable ε that is independent of $(\theta_1, L(\theta_1)), (\theta_2, L(\theta_2)), \ldots;$ then using search procedure S_4 (which they later define), with P-probability 1, as $n \to \infty$, and regardless of ε,

$$R(\theta_i, L) \to 0$$

where $R(\theta, L)$ is defined as the relative volume of points θ' such that $L(\theta') > L(\theta)$. Hence $L(\theta_i) \to L(\theta^*)$

7.2 Zhigljavsky and Pinter (1991) [ZP91]

Zhigljavsky and Pinter proposed the following algorithm.

Step 1: Choose a probability P_1 on feasible region Z.

Step 2: Obtain points $\theta_1^{(k)}, \ldots, \theta_{N_k}^{(k)}$ by sampling N_k times from the distribution P_k. Evaluate y at these points.

Step 3: According to a fixed (algorithm-dependent) rule construct a probability distribution P_{k+1} on Z.

Step 4: Check some appropriate stopping condition; if the algorithm is not terminated, then return to Step 2 (substituting $k+1$ for k).

Let y be continuous on the vicinity of a global minimizer θ^* of f, and assume that $\sum_{k=1}^{\infty} q_k = \infty$ for any $\theta \in Z$ and $\epsilon > 0$ where

$$q_k = q_k(\theta^*, \epsilon) = \text{vrai inf}_{\{\theta_1, \ldots, \theta_{k-1}\}} P_k(B(\epsilon))$$

Then for any $\delta > 0$ the sequence of random vectors $\theta_1, \theta_2, \dots$ generated by the algorithm with $N_k = 1\,(k = 1, 2, \dots)$ falls infinitely often into the set $A(\delta)$ with probability one, where A and B are balls around θ^*.

8 Conclusion

Noisy loss measurements hinder the effectiveness of random search, however, there are alterations we can make to the algorithm to ameliorate the problem slightly. Conditions for global convergence under noisy loss usually include restrictions on the mean, variance and tails of the noise distribution, in addition to being identically and independently distributed. Of course, in the real world, such conditions are difficult to attain; as such, other stochastic optimization methods are often preferred.

References

[ABC$^+$05] David L J Alexander, David W Bulger, James M Calvin, H Edwin Romeijn, and Ryan L Sherriff. Approximate implementations of pure random search in the presence of noise. *Journal of Global Optimization*, 31(4):601–612, April 2005.

[DK02] Luc Devroye and Adam Krzyzak. *Random Search Under Additive Noise*, pages 383–417. Springer US, New York, NY, 2002.

[GM89] S B Gelfand and S K Mitter. Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Applications*, 62(1):49–62, July 1989.

[Spa03] J C Spall. *Direct Methods for Stochastic Search*, pages 34–64. John Wiley Sons, Ltd, 2003.

[YF73] S.J Yakowitz and Lloyd Fisher. On sequential search for the maximum of an unknown function. *Journal of Mathematical Analysis and Applications*, 41(1):234–259, 1973.

[ZP91] Anatoly A. Zhigljavsky and J. Pintér. *Main Concepts and Approaches of Global Random Search*, pages 77–113. Springer Netherlands, Dordrecht, 1991.