Local Unimodal Sampling

By
Magnus Erik Hvass Pedersen, Andrew John Chipperfield
Hvass Laboratories Technical Report no. HL0801, 2008

Abstract

One of the simplest ways of optimizing a problem without knowledge of its gradient, is to iteratively and randomly pick a sequence of candidate solutions that improve on the fitness. This paper shows mathematically that to exhaust the optimum of a problem, the search-range from which the candidate solutions are picked, must be decreased during an optimization run. A simple technique for doing this is presented, and is demonstrated empirically to yield rapid convergence on a simple, yet representative unimodal optimization problem.

Keywords: Numerical optimization, stochastic, direct search, convergence.

1 Introduction

Optimization methods that do not explicitly use the gradient of the function to be optimized, but which are merely guided by a fitness measure of the candidate solutions, are desirable because the gradient is often laborious and sometimes even impossible to derive. These general purpose optimization methods are often known as Direct Search or Black-Box methods.

The simplest way of optimizing a problem in this direct manner, is to pick candidate solutions completely at random. But the number of possible solutions increases exponentially with an increasing number of optimization variables. This phenomenon is known as the Curse of Dimensionality [1, preface p. ix].

The random search for optimal solutions can be improved by localizing the sampling around the best known solution. This is what is meant by Local Sampling, and was introduced for real-valued optimization problems by Rastrigin [2] and Matyas [3], with the difference being that Rastrigin used a fixed stepsize, where as Matyas used a fixed size search-range. It was later realized by a number of researchers that both the Rastrigin and Matyas approaches needed to have their stepsize (or search-range) adapted during optimization [4][5][6][7][8][9]. And Schrack and Choit [10] found that the optimal stepsize can actually be approximated by a simple exponential decrease during an optimization run, although their formula for computing the decrease factor was somewhat complicated.

It is customary to try and prove that an optimization method converges to either a global or local optimum. Examples of such proofs for local sampling methods are found in [3][8][11], which all show that convergence to any given region surrounding the optimum, will occur with probability one. But the proofs show this in the extreme limit case; which can also be used to show inevitable convergence for completely random sampling of all possible solutions. Without an estimate on the number of iterations that are required, a convergence proof

is therefore of no use at all. Such an estimate was derived by Dorea [12], but was criticized through empirical experiments by Sarma [13], who found the convergence estimates to be inconsistent with observed performance on two real-world problems.

This paper uses probability theory to show that local sampling with a fixed search-range, does not converge to an optimum in a reasonable amount of time. The proof is also a limit-proof and demonstrates that when it comes to the extreme cases of probabilities, there are often two interpretations: One that is overly optimistic (namely that convergence always occurs), and one that is more realistic (that it will take a vast amount of time to converge to the optimum).

This paper also presents an extension to basic local sampling, providing a simple way of adapting the search-range. The method is especially related to the ones in [6][7][10], but differs significantly in that it remains a simple method throughout, while ensuring its search-range adaptation also corresponds well to the dimensionality of the search-space. The rapid convergence of this new method is not proven mathematically, but is rather demonstrated empirically.

The paper is structured as follows. Section 2 describes local sampling with a fixed search-range, which is shown in section 3 to be unable to deplete an optimum. Section 4 then presents a simple extension to local sampling which adapts the search-range during optimization, and section 5 gives an empirical demonstration that this enables the method to deplete an optimum. Section 6 gives the conclusive remarks.

2 Local Sampling

The essential idea of using localized stochastic sampling when searching for the minimum of some fitness function $f : \mathbb{R}^n \to \mathbb{R}$, is to choose a random point $\vec{y} \in \mathbb{R}^n$ from the neighbourhood of the current position $\vec{x}$ in the search-space, and move to the new position in case of fitness improvement. That is, let the new potential position $\vec{y}$ be defined as:

$$\vec{y} = \vec{x} + \vec{a}$$

with $\vec{a}$ chosen randomly and uniformly as follows: $\vec{a} \sim U(-\vec{d}, \vec{d})$, where $\vec{d}$ is the search-range. Then if $f(\vec{y}) < f(\vec{x})$ update the position: $\vec{x} \leftarrow \vec{y}$. Repeat this process a number of times.

3 Convergence Analysis

Keeping the search-range $\vec{d}$ fixed throughout the optimization run, will now be shown mathematically to decrease the probability of fitness improvement, the closer we get to the optimum of some fitness function.

3.1 Sphere Fitness Function

The Sphere fitness function is well suited for mathematical analysis of local sampling methods because it allows for particularly graceful mathematical deductions. Furthermore, as it is perhaps the simplest unimodal and continuous optimization problem available, it strongly suggests that an optimization method having trouble optimizing the Sphere function, will also have trouble with other, more complex optimization problems. For these reasons the Sphere function has also been used by other researchers for analyzing convergence aspects of optimization methods, see for example [2][3][11]. The Sphere fitness function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that is to be minimized, is formally defined as:

$$f(\vec{x}) = \sum_{i=1}^{n} x_i^2 \tag{1}$$

Note that the global optimum is situated in $\vec{x} = 0$, and as this is the only local optimum as well, the Sphere fitness function is unimodal. The function can be written in terms of the dot-product of vector $\vec{x}$ with itself, using basic identities of linear algebra:

$$f(\vec{x}) = \sum_{i=1}^{n} x_i^2 = \vec{x} \cdot \vec{x} = \|\vec{x}\|^2 \tag{2}$$

This means that for $\vec{y}$ to improve on the fitness of $\vec{x}$ the following must hold:

$$f(\vec{y}) < f(\vec{x}) \;\Leftrightarrow\; \|\vec{y}\| < \|\vec{x}\| \tag{3}$$

In other words, for $\vec{y}$ to improve on $\vec{x}$ its norm must be smaller.

3.2 Single-Dimensional Case

To understand the underlying idea of the proof of non-convergence for local sampling with a fixed search-range, consider the single-dimensional Sphere function:

$$f(x) = x^2,\ x \in \mathbb{R}$$

where local sampling will calculate the agent's potential new position as $y = x+a$, with $a \sim U(-d, d)$ for some sampling range delimiter $d > 0$. The condition for y to improve on the fitness of x is simply the one-dimensional case of Eq.(3): $f(y) < f(x) \;\Leftrightarrow\; |y| < |x|$. So a sample y taken from the range $(-|x|, |x|)$ will result in an improvement over x, but since y is actually taken from the range $(x - d, x + d)$, it means the probability of finding such an improved position for y, is given by the amount of overlap between these two ranges. For the one-dimensional case, this can be expressed as:

$$\Pr\left[f(y) < f(x)\right] = \begin{cases} |x|/d & \text{, if } |x| < d/2 \\ 1/2 & \text{, else} \end{cases}$$

Initially, when x is far from the global optimum, the probability of finding an improved position by localized random sampling, is always $1/2$. This is

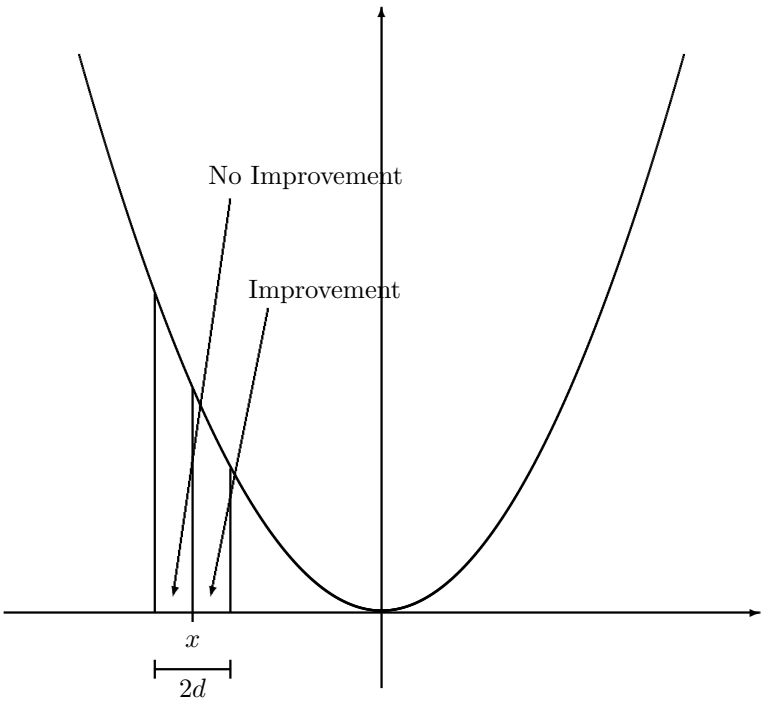

Figure 1: Local sampling of the Sphere function. When the current position x is far from the optimum, the probability of improvement is always 1/2.

demonstrated in figure 1. But as x approaches the global optimum, and when the sampling range d remains fixed, the probability of improvement approaches zero. This is demonstrated in figure 2.

However, as the probability of improvement is non-zero unless x is already situated in the optimum, an improved position y is inevitably discovered if enough samples are taken. So the method does eventually converge to the optimum. But convergence for an optimization method should never be proven using extreme limit-cases, without augmenting that proof with a good estimate on the computational effort required to obtain satisfactory results. In the case of local sampling with a fixed range, it was shown that the computational effort approaches infinity as the optimum is approached.

3.3 Multi-Dimensional Case

Since the Sphere function is separable, in the sense that the dimensions are independent of each other in their influence on the fitness measure f, the above argument can be repeated for each dimension in the multi-dimensional case, to show the same result applies there. But it is also possible to show it using set-volumes, which will be done now.

It follows from Eq.(2) that the vectors $\vec{x}$ evaluating to fitnesses less than

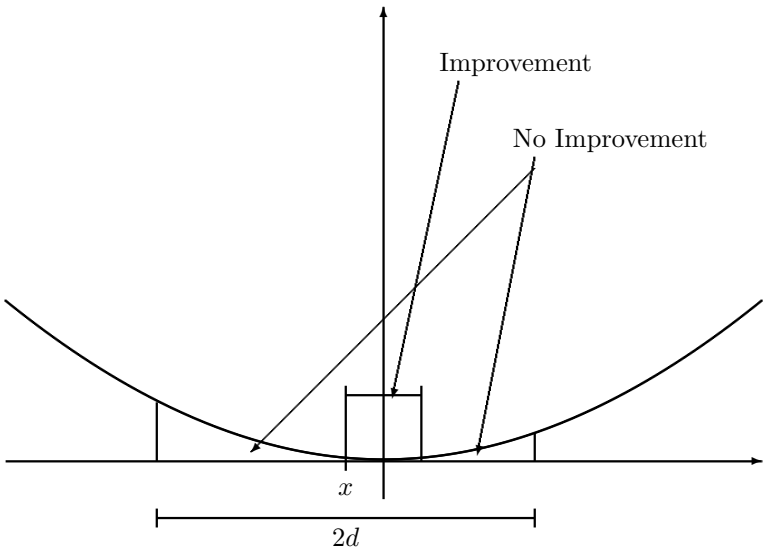

Figure 2: Local sampling of the Sphere function. When the current position x is close to the optimum, the probability of improvement is $2|x|/(2d) = |x|/d$. If d remains fixed, this probability will approach zero as x approaches the optimum situated in zero.

some fitness-value $F \geq 0$ must satisfy:

$$f(\vec{x}) \leq F \;\Leftrightarrow\; \|\vec{x}\|^2 \leq F \;\Leftrightarrow\; \|\vec{x}\| \leq \sqrt{F}$$

which designates a hypersphere of radius $\sqrt{F}$. Let us denote a general n-dimensional hypersphere with radius r by $S_n(r)$, thus defined as:

$$S_n(r) = \{\vec{z} : \|\vec{z}\| \leq r\}$$

with the vectors $\vec{x}$ being n-dimensional. This means positions having a fitness better than or equal to F, must satisfy:

$$f(\vec{x}) \leq F \;\Leftrightarrow\; \vec{x} \in S_n\left(\sqrt{F}\right) \tag{4}$$

Expressing this fitness requirement as set-membership, is convenient in the probability-theoretical analysis of the Sphere function below.

The volume of a hypersphere $|S_n(r)|$ is defined in [14, p. 1442] in terms of the constant s_n, which is the hyper-surface area of the n-dimensional hypersphere with unit radius. The hypersphere volume is then defined as:

$$|S_n(r)| = \frac{s_n \cdot r^n}{n} \tag{5}$$

Although [14, Eq.(7), p. 1442] provides an explicit formula for computing s_n, it is fairly complicated. Fortunately enough, the value s_n is not needed for the

analysis to follow, as the probability boundaries may be studied instead, which suffices for the conclusions that are sought.

The idea from the one-dimensional Sphere function can be extended directly to n-dimensional search-spaces. The relation from Eq.(3) is used again, and states the conditions under which some position $\vec{y}$ will improve on the Sphere function's fitness of $\vec{x}$; namely when the norm of $\vec{y}$ is less than the norm of $\vec{x}$. In the multi-dimensional local sampling used here, $\vec{y}$ is taken from the hypercube surrounding $\vec{x}$ and having sidelengths $2d$ (assuming the search-range d is identical for all dimensions). Let this n-dimensional hypercube be denoted by:

$$C_n(\vec{x}, d) = [x_1 - d,\, x_1 + d] \times \cdots \times [x_n - d,\, x_n + d]$$

So the probability of $\vec{y}$ improving on the fitness of $\vec{x}$, is the size of the intersection between this hypercube and the (open) hypersphere of radius $\|\vec{x}\|$, divided by the total size of the hypercube used for sampling, that is:

$$\Pr\left[f(\vec{y}) < f(\vec{x})\right] = \frac{|C_n(\vec{x}, d) \cap S_n(\|\vec{x}\|)|}{|C_n(\vec{x}, d)|} = \frac{|C_n(\vec{x}, d) \cap S_n(\|\vec{x}\|)|}{(2d)^n} \qquad (6)$$

where the last identity follows from the fact that $|C_n(\vec{x}, d)| = (2d)^n$

The exact value of Eq.(6) is not important, as we only need to note a few things. First note that the denominator $(2d)^n$ remains constant because the range delimiter d of the sampling hypercube is fixed. Second, note that the numerator approaches zero as $\vec{x}$ approaches the global optimum of the Sphere function, because $|S_n(\|\vec{x}\|)|$ and hence $|C_n(\vec{x}, d) \cap S_n(\|\vec{x}\|)|$ approach zero. So the probability in Eq.(6) approaches zero as we approach the global optimum of the Sphere function.

This finding is not so strange, because in other words, the hypercube used for sampling is of fixed size throughout the optimization run, while the hypersphere that holds the part of the search-space with improved fitnesses, decreases in size each time such an improvement is found. Thus we have proved a remarkable deficiency with localized sampling of the Sphere function. Namely that the closer we get to the optimum, the less likely it is to find any improved positions; provided the neighbourhood from which we sample has a fixed size.

4 Local Unimodal Sampling

An optimization method that was designed to overcome the above limitation is called Local Unimodal Sampling (LUS), and has been found to work for harder optimization problems as well.

4.1 Search-Range Decrease

The sampling done by the LUS method is identical to that in section 2, with the initial search-range $\vec{d}$ chosen as the full range of the search-space. But instead of keeping a fixed search-range through the entire optimization run,

the LUS method decreases the search-range exponentially when samples fail to improve on the fitness of the current position. The question is how much this decrease should be. Many optimization methods have used exponential decrease of the search-range, including the method known as Pattern Search by Hooke and Jeeves [15], the Luus-Jaakola method [7], and the method by Schrack and Choit [10]. The method by Fermi and Metropolis as described in [16] also uses exponential decrease of its search-range, and had success with halving the search-range one dimension at a time. To use the idea of halving the search-range when all n dimensions of the search-space are sampled simultaneously, as done in the LUS method, the halving would have to occur after n failures to improve the fitness, to yield a similar combined effect. This was proposed by Lawrence and Steiglitz [6] who would halve the search-range once for every n samples. However, it is possible to make the search-range decrease occur more smoothly, by multiplying the search-range $\vec{d}$ with a factor q for each failure to improve the fitness, instead of waiting until all n samples have been made:

$$\vec{d} \leftarrow q \cdot \vec{d}$$

To yield a halving of the search-range after n failures to improve the fitness, would then require for the decrease factor to be defined as: $q = (1/2)^{1/n}$. But experience has shown that some adjustment is usually necessary, meaning the decrease factor q should instead be defined as:

$$q = \left(\frac{1}{2}\right)^{\beta/n} \tag{7}$$

With $0 < \beta < 1$ causing slower decrease of the search-range, and $\beta > 1$ causing more rapid decrease. Note that applying this n times yields a search-range reduction of $q^n = (1/2)^{\beta}$, and for $\beta = 1$ this would mean a halving of the search-range for all dimensions. A value of $\beta = 1/3$ has been found to yield good results on a broad range of problems and is therefore used in the experiments below. The algorithm for the LUS optimization method is shown in figure 3.

5 Experimental Results

A mathematical proof of convergence for the LUS method will not be given here. Instead convergence within a reasonable number of iterations is demonstrated empirically, for the Sphere function with various dimensionalities. This is shown in figure 4 where fitness traces are averaged from 50 optimization runs of the LUS method, with the search-space being bounded to $x_i \in [-100, 100]$ for all dimensions i. And to increase the difficulty of optimizing the Sphere problem, the LUS method is initialized in only part of the search-space at the beginning of each optimization run, namely in the range $x_i \sim U(50, 100)$ for all i.

The fitness-axis in figure 4 is log-scaled, thus showing the LUS method has an exponential approach to the optimum of the Sphere fitness function, with exception of its early iterations. The fitness traces also show that the LUS method

- Initialize $\vec{x}$ to a random position in the search-space:

$$\vec{x} \sim U\left(\vec{b}_{lo}, \vec{b}_{up}\right)$$

  Where $\vec{b}_{lo}$ and $\vec{b}_{up}$ are the search-space boundaries.

- Set the initial search-range $\vec{d}$ to cover the entire search-space:

$$\vec{d} \leftarrow \vec{b}_{up} - \vec{b}_{lo}$$

- Until a termination criterion is met (e.g. a given number of fitness evaluations have been executed, observed fitness stagnates, or a fitness threshold is met), repeat:

  - Pick a random vector $\vec{a} \sim U\left(-\vec{d}, \vec{d}\right)$
  - Add this to the current position $\vec{x}$, to create the new potential position $\vec{y}$:

$$\vec{y} = \vec{x} + \vec{a}$$

  - If $(f(\vec{y}) < f(\vec{x}))$ then update the position:

$$\vec{x} \leftarrow \vec{y}$$

    Otherwise decrease the search-range by multiplication with the factor q from Eq.(7):

$$\vec{d} \leftarrow q \cdot \vec{d}$$

Figure 3: The LUS algorithm.

only requires a number of iterations that depends linearly on the dimensionality of the Sphere problem. This is an important issue, because the number of potential solutions that must be searched for an optimization problem increases exponentially with increasing dimensionality. So optimization methods which have a sub-exponential time-complexity with regard to the dimensionality of that problem, are therefore preferable because they can be executed within a reasonable amount of time.

Furthermore, due to the exponential decrease of search-range in the LUS method, a computer implementation using discrete floating point numbers will quickly underflow. It is therefore not even possible to have long optimization runs of the LUS method. But this rapidness of the LUS method is again considered an advantage.

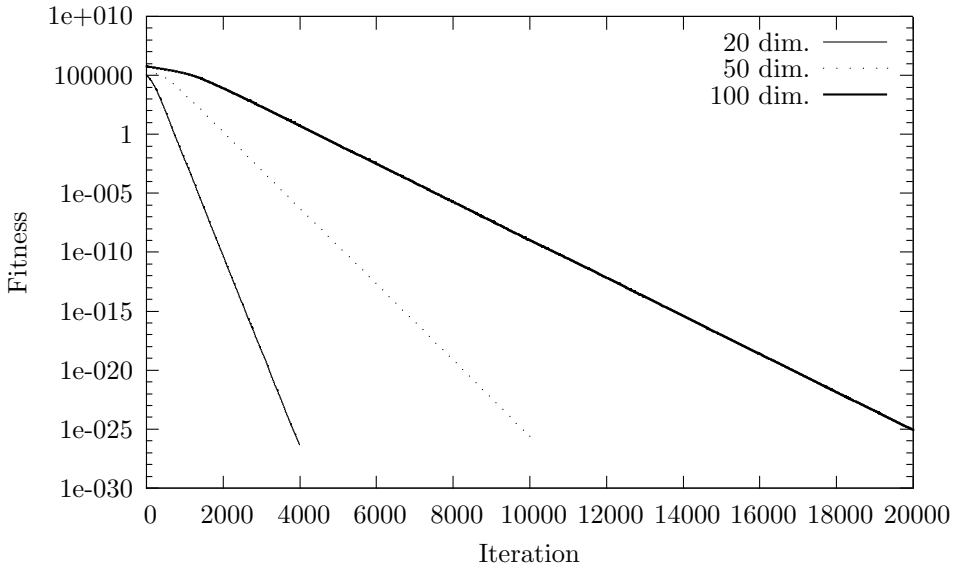

Figure 4: Fitness traces of using the LUS method to optimize the Sphere problem with various dimensionalities. Plot shows the average fitness obtained at each iteration of 50 optimization runs. Fitness axis is log-scaled.

6 Conclusion

This paper showed mathematically that an optimization method based on iterative, local sampling of the search-space, will not converge to the optimum of a simple unimodal problem, unless the sampling range is adapted during optimization.

An optimization method called the LUS method was then presented, which offers a simple way of adapting its search-range through an exponential decrease, and in a manner that is also fitting for the dimensionality of the optimization problem at hand. The LUS method was shown empirically to rapidly, and seemingly indefinitely, approach and deplete the optimum of the unimodal Sphere problem, for both smaller and larger dimensionalities.

References

[1] R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[2] L.A. Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automation and Remote Control*, 24(10):1337–1342, 1963.

[3] J. Matyas. Random optimization. *Automation and Remote Control*, 26(2):246–253, 1965.

[4] V.A. Mutseniyeks and L.A. Rastrigin. Extremal control of continuous multi-parameter systems by the method of random search. *Engineering Cybernetics*, 1:82–90, 1964.

[5] M.A. Schumer and K. Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13(3):270–276, 1968.

[6] J.P. Lawrence III and K. Steiglitz. Randomized pattern search. *IEEE Transactions on Computers*, C-21(4):382–385, 1972.

[7] R. Luus and T.H.I. Jaakola. Optimization by direct search and systematic reduction of the size of search region. *American Institute of Chemical Engineers Journal (AIChE)*, 19(4):760–766, 1973.

[8] N. Baba. Convergence of a random optimization method for constrained optimization problems. *Journal of Optimization Theory and Applications*, 33(4):451–461, 1981.

[9] F.J. Solis and R.J-B. Wets. Minimization by random search techniques. *Mathematics of Operation Research*, 6(1):19–30, 1981.

[10] G. Schrack and M. Choit. Optimized relative step size random searches. *Mathematical Programming*, 10(1):230–244, 1976.

[11] G.G. Nair. On the convergence of the LJ search method. *Journal of Optimization Theory and Applications*, 28(3):429–434, 1979.

[12] C.C.Y. Dorea. Expected number of steps of a random optimization method. *Journal of Optimization Theory and Applications*, 39(3):165–171, 1983.

[13] M.S. Sarma. On the convergence of the Baba and Dorea random optimization methods. *Journal of Optimization Theory and Applications*, 66(2):337–343, 1990.

[14] E.W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman & Hall/CRC, 2nd edition, 2002.

[15] R. Hooke and T.A. Jeeves. "Direct Search" solution of numerical and statistical problems. *Journal of the Association for Computing Machinery (ACM)*, 8(2):212–229, 1961.

[16] W.C. Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.