UNIVERSITY OF DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes and Systems by means of Computational Intelligence Methods

Lower bounds for hit-and-run optimization

Jens Jägersküpper

No. CI-229/07

Technical Report ISSN 1433-3325 April 2007

Secretary of the SFB 531 · University of Dortmund · Dept. of Computer Science/LS 2
44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the University of Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

Lower bounds for hit-and-run optimization

Jens Jägersküpper*

Universität Dortmund, Informatik 2, 44221 Dortmund, Germany
JJ@Ls2.cs.uni-dortmund.de

Abstract. "Hit-and-run is fast and fun" to generate a random point in a high dimensional convex set K (Lovász/Vempala, MSR-TR-2003-05). More precisely, the hit-and-run random walk mixes fast independently of where it is started inside the convex set (as opposed to the ball-walk, which requires a warm start). To hit-and-run from a point $\boldsymbol{x} \in \mathbb{R}^n$, a line L through $\boldsymbol{x}$ is randomly chosen (uniformly over all directions). Subsequently, the walk's next point is sampled from $L \cap K$ using a membership oracle which tells us whether a point lies in K or not.

Here the focus is on black-box optimization, however, where the function $f\colon \mathbb{R}^n \to \mathbb{R}$ to be minimized is given as an oracle, namely a black box for f-evaluations. We obtain in an obvious way a direct-search method when we substitute the f-oracle for the K-membership oracle to do a line search over L, and we are interested in how fast such a hit-and-run search heuristic converges to the optimum point $\boldsymbol{x}^*$ in the search space $\mathbb{R}^n$.

We prove that, even under the assumption of perfect line search, the search converges (at best) linearly at an expected rate which is larger than $1-1/n$. This implies a lower bound of $0.5\,n$ on the expected number of line searches necessary to halve the approximation error. Moreover, we show that $0.4\,n$ line searches suffice to halve the approximation error only with an exponentially small probability of $\exp(-\Omega(n^{1/3}))$. Since each line search requires at least one query to the f-oracle, the lower bounds obtained hold also for the number of f-evaluations.

1 Introduction

Finding an optimum of a given function $f\colon S \to \mathbb{R}$ is one of the fundamental problems—in theory as well as in practice. The search space S can be discrete or continuous, like $\mathbb{N}$ or $\mathbb{R}$. If S has more than one dimension, it may also be a mixture, like it is the case for optimization tasks that are so-called mixed-integer programs. Here the optimization in "high-dimensional" Euclidean space is considered, i. e., the search space is $\mathbb{R}^n$. What "high-dimensional" means is usually anything but well defined. A particular 10-dimensional problem in practice may already be considered "high-dimensional" by the one who tries to solve it. Here the crucial aspect is how the optimization time scales with the dimensionality of the search space $\mathbb{R}^n$, i. e., we consider the optimization time as a function of n. In other words, here we are interested in what happens when the dimensionality

* supported by the German Research Foundation (DFG) through the collaborative research center "Computational Intelligence" (SFB 531)

of the search space gets higher and higher. This viewpoint is typical for analyses in computer science. Unfortunately, it seems that optimization in continuous search spaces is not one of the core topics in computer science. Rather it lies in the domain of operations research and mathematical programming. There, however, focusing on how the optimization time scales with the search space's dimension seems rather uncommon. Usually, the performance of an optimization method is described by means of convergence theory. As an example, let us take a closer look at "Q-linear convergence" (we drop the "Q" in the following): Let $\boldsymbol{x}^*$ denote the optimum search point of a unimodal function and $\boldsymbol{x}^{[k]}$ the approximate solution after k optimization steps. Then we have

$$\frac{\mathrm{dist}(\boldsymbol{x}^*, \boldsymbol{x}^{[k+1]})}{\mathrm{dist}(\boldsymbol{x}^*, \boldsymbol{x}^{[k]})} \to r \in \mathbb{R}_{<1} \quad \text{as} \quad k \to \infty$$

where $\mathrm{dist}(\cdot, \cdot)$ denotes some distance measure, most commonly the Euclidean distance between two points (when considering convergence towards $\boldsymbol{x}^*$ in the search space $\mathbb{R}^n$, as we do here), or the absolute difference in function value (when considering convergence towards the optimum function value in the objective space). Apparently, there seems to be no connection to n, the dimension of the search space. Yet only if r is an absolute constant, there is actual independence of n. In general, however, the *convergence rate* r depends on n. When we are interested in, say, the number of steps necessary to halve the approximation error (given by the distance from $\boldsymbol{x}^*$), the order of this number with respect to n precisely depends on how r depends on n. For instance, if $r = 1 - 0.5/n$, we need $\Theta(n)$ steps; if $r = 1 - 0.5/n^2$, we need $\Theta(n^2)$ steps, and if $r = 1 - 2^{-n}$, we need $2^{\Theta(n)}$ steps. For any fixed dimension, however, in any of the three cases we actually have linear convergence. Thus, the order of convergence tells us something about the "speed" of the optimization, but in general nothing about the n-dependence of the number of steps necessary to ensure a certain approximation error (unless r is an absolute constant, then it takes a constant number of steps to halve the distance from $\boldsymbol{x}^*$ independently of n). So, in case of linear convergence, we want to know how the convergence rate depends on the dimensionality of the search space.

Methods for solving optimization problems in continuous domains, essentially $S = \mathbb{R}^n$, are usually classified into first-order, second-order, and zeroth-order methods, depending on whether they utilize the gradient (the first derivative) of the objective function, the gradient and the Hessian (the second derivative), or neither of both. A zeroth-order method is also called *derivative-free* or *direct search method*. Newton's method is a classical second-order method; first-order methods can be (sub)classified into Quasi-Newton, steepest descent, and conjugate gradient methods. Classical zeroth-order methods try to approximate the gradient and to then plug this estimate into a first-order method. Finally, amongst the modern zeroth-order methods, randomized search heuristics like simulated annealing and evolutionary algorithms come into play, which are supposed general-purpose search heuristics.

When information about the gradient is not available, for instance if f relates to a property of some workpiece and is given by computer simulations or even

by real-world experiments, then first-order (and also second-order) methods just cannot by applied. As the approximation of the gradient usually involves $\Omega(n)$ f-evaluations, a single optimization step of a classical zeroth order-method is computationally expensive, in particular if f is given implicitly by simulations. In practical optimization, especially in mechanical engineering, this is often the case, and particularly in this field randomized search heuristics (especially evolutionary algorithms) are becoming more and more popular. However, the enthusiasm in practical optimization heuristics has led to an unclear variety of very sophisticated and problem-specific algorithms. Unfortunately, from a theoretical point of view, the development of such algorithms is solely driven by practical success, whereas the aspect of a theoretical analysis is left aside.

In such situations f is given to the optimization algorithm as an oracle for f-evaluations (zeroth-order oracle) and the cost of the optimization (the runtime) is defined as the number of queries to this oracle, and we are in the so-called *black-box optimization* scenario. Nemirovsky and Yudin (1983, p. 333) state (w. r. t. optimization in continuous search spaces) in their book *Problem Complexity and Method Efficiency in Optimization:* "From a practical point of view this situation would seem to be more typical. At the same time it is objectively more complicated and it has been studied in a far less extend than the one [with first-order oracles/methods] considered earlier." After more than two decades there still seems to be some truth in their statement, though to a smaller extent. For discrete black-box optimization, a complexity theory has been successfully started, cf. Droste, Jansen, and Wegener (2006). Lower bounds on the number of f-evaluations (the *black-box complexity*) are proved with respect to classes of functions when an arbitrary(!) optimization heuristic knows about the class $\mathcal{F}$ of functions to which f belongs, but nothing about f itself. The benefits of such results are obvious: They can prove that an allegedly poor performance of an apparently simple black-box algorithm on f is not due to the algorithm's simpleness, but due to $\mathcal{F}$'s inherent black-box complexity. As mentioned above, the situation for heuristic optimization in continuous search spaces is different, especially with respect to randomized/stochastic methods. The results to be presented here contribute to this less-developed but emerging field of optimization theory.

2 The Framework for the Randomized Methods under Consideration

As already noted above, classical zeroth-order methods (i. e. black-box optimizers) for continuous search spaces usually try to approximate the gradient of the function f to be minimized at the current search point $\boldsymbol{x}$. Subsequently, a line search along gradient direction is performed to find the next search point, which replaces $\boldsymbol{x}$. Usually, the line search aims at locating the best (with respect to the f-value) point on the line through $\boldsymbol{x}$, and various strategies for how to do the line search exist (Armijo/Goldstein, Powell/Wolfe, etc.). As the approximation of the gradient usually involves $\Omega(n)$ f-evaluations, and as the (approximate) gradient's direction may significantly differ from the direction pointing directly to the

optimum $\boldsymbol{x}^*$ anyway (cf. ill-conditioned quadratics), more and more direct search heuristics have been proposed which abandon gradient approximation. Among the first and most prominent ones are the pattern search by Hooke and Jeeves (1961) and the (downhill) simplex method by Nelder and Mead (1965); cf. Kolda, Lewis, and Torczon (2004) for a comprehensive review. Surprisingly, also already in the 1960s a randomized direct search method was proposed, namely the so-called *evolution strategy* by Rechenberg (1965) and Schwefel (1965). For some obscure reason, however, there has been resentment against randomized algorithms in these early years. This started to change with the randomization of quicksort and randomized testing for primality. At the latest by the time when Dyer, Frieze, and Kannan (1989) came up with a randomized approximation algorithm for the computation of the volume of a convex body in high dimensional space, the (possible) benefits of randomization has won recognition. Though the polynomial expected runtime of this algorithm was not very practical, it showed in principle the power of randomization since for any deterministic algorithm there is a convex set for which the relative error is $n^{\Omega(n)}$ after any polynomial number of steps. At the core of this algorithm was a random walk on a (sufficiently fine) lattice. This algorithm was further improved, in particular by substituting the so-called *ball walk* for the original lattice walk. One step of this ball walk consists in uniformly choosing a point from the hyper-ball of radius δ around the current point. If this point lies in the convex set, then it becomes the next point of the walk. Apparently, one has to choose the parameter δ appropriately. Moreover, when the ball walk is started very close to the corner of a hypercube, just for instance, it may need an exponential number of steps to leave this corner, making a so-called warm start necessary (i. e. a preprocessing). As recently shown by Lovász and Vempala (2006), using the *hit-and-run walk* instead of the ball walk avoids these two issues. Hit-and-run mixes fast even when started close to the boundary of the convex set, and moreover, no "step size" needs to be appropriately predefined. Also an optimization algorithm based on random walks in covex sets has been proposed (Bertsimas and Vempala, 2004).

As already noted in the abstract, to hit-and-run from a point $\boldsymbol{x} \in \mathbb{R}^n$ within a convex set $K \subset \mathbb{R}^n$, a line L through $\boldsymbol{x}$ is randomly chosen (uniformly over all directions). Subsequently, the next point (to replace $\boldsymbol{x}$) is sampled from $L \cap K$ (as uniformly as possible) using a membership oracle which tells us whether a sample from L lies in K or not. As also already noted in the abstract, we obtain in an obvious way a hit-and-run direct-search method for black-box optimization of $f\colon \mathbb{R}^n \to \mathbb{R}$ when we substitute the f-oracle for the K-membership oracle. Thus, the framework of the heuristics for black-box optimization we consider is as follows: For a given initialization of $\boldsymbol{x} \in \mathbb{R}^n$ the following loop is performed:

1. Randomly choose a line L through $\boldsymbol{x}$ (uniformly over all directions).
2. By some kind of a line search (using the f-oracle), find a point $\boldsymbol{x}' \in L$.
3. Set $\boldsymbol{x} := \boldsymbol{x}'$ and GOTO 1 (unless stopping is requested; then output $\boldsymbol{x}$).

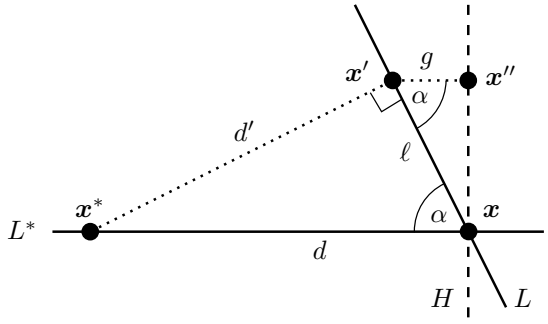
Naturally, we are interested in how fast such a heuristic converges to the optimum point $\boldsymbol{x}^* \in \mathbb{R}^n$ (we assume that there is a unique global optimum), in particular:

How fast can it converge in principle? That is, we are interested in a general lower bound which is universal for the class of hit-and-run heuristics.

Note that there are *no* assumptions on how the line search is performed. In particular, for the line search in the ith iteration, the algorithm may use *all* the information gathered from *all* the samples drawn during the preceding $i-1$ line searches. Naturally, in each step the choice of how to do the line search may depend on the actual direction of L. All in all, a large variety of adaptive strategies for black-box optimization with unlimited memory is covered by our framework.

3 General Lower Bound

Since any reasonable line-search strategy implies at least one query to the f-oracle, in our scenario the number of f-evaluations is bounded below by the number of line searches. Thus, we focus on the number of line searches in the following and aim at a general lower bound. Therefore, we need an upper bound on the gain of a single line search. We consider the best case: When we want the heuristic to approach the unique optimum point $\boldsymbol{x}^*$ as fast as possible, we may optimistically assume that $\boldsymbol{x}'$ was chosen from the line L such that distance between $\boldsymbol{x}'$ and $\boldsymbol{x}^*$ is minimum. Call this a *perfect line search*. The situation is depicted in the figure below.



It is well known that the distance between $\boldsymbol{x}^*$ and $\boldsymbol{x}'$ is minimum when $\boldsymbol{x}' \in L \supset \{\boldsymbol{x}\}$ is such that the line passing through $\boldsymbol{x}'$ and $\boldsymbol{x}^*$ is perpendicular to the line L (given that $\boldsymbol{x}^* \notin L$, which is the case with probability one, unlike already $\boldsymbol{x}$ coincides with the optimum point $\boldsymbol{x}^*$, because L's direction is chosen uniformly over all directions).

Let $d := \mathrm{dist}(\boldsymbol{x}, \boldsymbol{x}^*)$ denote the current approximation error in the search space and let $d' := \mathrm{dist}(\boldsymbol{x}', \boldsymbol{x}^*)$. Furthermore, let L^* denote the line through $\boldsymbol{x}$ and $\boldsymbol{x}^*$. Now consider the hyper-plane H which contains $\boldsymbol{x}$ and is perpendicular to L^*. Let $\boldsymbol{x}'' := \arg\min_{\boldsymbol{y} \in H} \mathrm{dist}(\boldsymbol{x}', \boldsymbol{y})$ denote the unique point in H with smallest distance from $\boldsymbol{x}'$. Then the angle α between L and L^* equals the angle between L and the line through $\boldsymbol{x}'$ and $\boldsymbol{x}''$ (which is parallel to L^* since it is perpendicular to H just as L^*). Consequently, we have

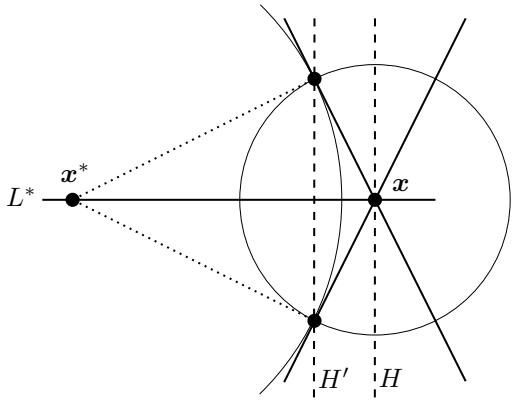
$$d' = d \cdot \sin\alpha \quad \text{and} \quad \mathrm{dist}(\boldsymbol{x}', H) = \mathrm{dist}(\boldsymbol{x}', \boldsymbol{x}) \cdot \cos\alpha.$$

Let $g := \mathrm{dist}(\boldsymbol{x}', \boldsymbol{x}'')$ denote the distance of $\boldsymbol{x}'$ from H, and $\ell := \mathrm{dist}(\boldsymbol{x}', \boldsymbol{x})$ so that we have $g/\ell = \cos\alpha \; (= \ell/d)$. Since $d'/d = \sin\alpha = \sqrt{1-(\cos\alpha)^2}$, we obtain

$$\frac{d'}{d} = \sqrt{1-(g/\ell)^2}, \tag{1}$$

which ranges in $[0,1]$ since $g \in [0,\ell]$. Thus, instead of focusing on the distribution of $\sin\alpha$ when L is chosen uniformly over all directions, we can focus on the ratio g/ℓ and concentrate on the distribution of this *relative distance* of $\boldsymbol{x}'$ from the hyper-plane H (namely, relative to the distance of $\boldsymbol{x}'$ from $\boldsymbol{x}$). (It will shortly become clear why this makes sense.)

In two dimensions, like in the figure above, for any fixed $d' \in (0,d)$ there are exactly two (different) lines through $\boldsymbol{x}$ with distance d' from the optimum point $\boldsymbol{x}^*$. (Note that by fixing d' we also fixed ℓ and g.) In three or more dimensions, however, there is an infinite number of such lines. In three dimensions they form a double cone with its apex at $\boldsymbol{x}$, and all points of this cone with an actual distance of d' from $\boldsymbol{x}^*$ (namely all $\boldsymbol{x}'$) form a circle. This circle lies in a plane which is parallel to H (a plane in three dimensions). In general, i. e. in $n \geq 3$ dimensions, the potential points $\boldsymbol{x}'$ form the set $S := \{\boldsymbol{x}' \in \mathbb{R}^n \mid \mathrm{dist}(\boldsymbol{x}', \boldsymbol{x}^*) = d' \text{ and } \mathrm{dist}(\boldsymbol{x}', \boldsymbol{x}) = \ell\}$, which is an $(n-1)$-sphere since S is the intersection of two hyper-spheres, namely of the hyper-sphere with radius d' centered at $\boldsymbol{x}^*$ and the hyper-sphere with radius ℓ centered at $\boldsymbol{x}$. Moreover, S lies in the hyper-plane H' which is parallel to H such that it has distance g from H and distance $d-g$ from $\boldsymbol{x}^*$. The situation is depicted below, where the left sphere consists of all points with distance d' from the optimum point $\boldsymbol{x}^*$, and the right sphere consists of all points with distance ℓ from our current approximate solution $\boldsymbol{x}$.



Recall that we fixed $d' \in (0,d)$ for the above discussion, and that this implies fixed values for ℓ and $g = \mathrm{dist}(H', H)$. Now consider a randomly chosen line L through $\boldsymbol{x}$ (uniform over all directions). According to our construction, if L penetrates the $(n-1)$-sphere $S \subset H'$, then the perfect line search on L yields a point with a distance of exactly d' from $\boldsymbol{x}^*$. Now, if L lies inside the double cone, i. e., L penetrates the open $(n-1)$-ball the missing boundary of which is S, then the perfect line search yields a point with a distance smaller than d' from $\boldsymbol{x}^*$. If

L lies outside the double cone (except for passing through the apex $\boldsymbol{x}$, of course), then the perfect line search yields a point with a distance larger than d' from $\boldsymbol{x}^*$. Thus, we are interested in the probability p that L is chosen such that it lies inside the cone. Namely, p is the probability that the perfect line search yields a point with a distance of less than d' from $\boldsymbol{x}^*$.

Now, how can we actually pick a line through $\boldsymbol{x}$ such that its direction is uniformly random? We pick uniformly at random a point $\boldsymbol{y}$ from/over the unit hyper-sphere centered at $\boldsymbol{x}$ and choose L as the line through $\boldsymbol{y}$ and $\boldsymbol{x}$. From this point of view, the perfect line search yields a point with a distance of exactly d' from $\boldsymbol{x}^*$ if $\boldsymbol{y}$'s distance from H is exactly g/ℓ; a point with a distance smaller than d' from $\boldsymbol{x}^*$ if $\boldsymbol{y}$'s distance from H is larger than g/ℓ; and a point with a distance larger than d' from $\boldsymbol{x}^*$ if $\boldsymbol{y}$'s distance from H is smaller than g/ℓ.

In other words, we can consider the random variable $R := d'/d$ as a function of the random variable G defined as $\boldsymbol{y}$'s distance from the hyper-plane H, where the point $\boldsymbol{y}$, is chosen uniformly over the unit hyper-sphere centered at $\boldsymbol{x}$. Namely, we have $R = \sqrt{1 - G^2}$, cf. Equation 1 on the facing page. (Note that the distribution of $\boldsymbol{y}$ over $\mathbb{R}^n$ is spherically symmetric; more precisely, it is *isotropic*, i. e. invariant w. r. t. orthonormal transformations.) For $n \geq 4$ the density function of G's distribution over $[0, 1]$ is given by $(1 - x^2)^{(n-3)/2}/\Psi$ (Jägersküpper, 2003), where $\Psi = \int_0^1 (1 - x^2)^{(n-3)/2} \, \mathrm{d}x$ (normalization) and the value of this integral is $\Psi = \sqrt{\pi/4} \cdot \Gamma(n/2 - 1/2)/\Gamma(n/2) = \sqrt{\pi/n}/2 + \Theta(n^{-3/2})$, where "$\Gamma$" denotes the well-known gamma function. Consequently, $\boldsymbol{y}$'s expected distance from H equals $\int_0^1 x \cdot (1 - x^2)^{(n-3)/2} \, \mathrm{d}x/\Psi = (n - 1)^{-1}/\Psi$ which turns out to be $\sqrt{2/\pi}/\sqrt{n} + \Theta(n^{-3/2})$. That is, $\boldsymbol{y}$'s expected distance from H is about $0.8/\sqrt{n}$. This might appear bewildering (at first) since this implies that, as the search space's dimensionality increases, the expected distance from H tends to zero—although $\boldsymbol{y}$'s distance form $\boldsymbol{x}$ is fixed to one and H is hit with zero probability. However, noting that H is an affine subspace with dimension $n-1$ (i. e. codimension 1), it may become more plausible that getting far away from H becomes less and less probable as n increases. It might help even more to recall that an n-hypercube with unit diameter (longest diagonal) has edges of length $1/\sqrt{n}$.

So, what does this help? Naturally, $\mathsf{E}[G]$ does not tell us much about $\mathsf{E}[R] = \mathsf{E}\big[\sqrt{1 - G^2}\big]$, the expectation in which we are actually interested. We can easily compute it, though:

$$\mathsf{E}[R] = \int_0^1 \sqrt{1 - x^2} \cdot (1 - x^2)^{(n-3)/2}/\Psi \, \mathrm{d}x = \int_0^1 (1 - x^2)^{n/2-1} \, \mathrm{d}x/\Psi.$$

Since $\int_0^1 (1 - x^2)^{n/2-1} \, \mathrm{d}x = \sqrt{\pi/4} \cdot \Gamma(n/2)/\Gamma(n/2 + 1/2)$, we obtain

$$\mathsf{E}[R] = \frac{\sqrt{\pi/4} \cdot \Gamma(n/2)/\Gamma(n/2 + 1/2)}{\sqrt{\pi/4} \cdot \Gamma(n/2 - 1/2)/\Gamma(n/2)} = \frac{\Gamma(n/2) \cdot \Gamma(n/2)}{\Gamma(n/2 - 1/2) \cdot \Gamma(n/2 + 1/2)}. \quad (2)$$

Using $\Gamma(n/2 + 1/2) = \Gamma(n/2 - 1/2) \cdot (n/2 - 1/2)$, we have

$$\mathsf{E}[R] = \frac{n - 1}{2} \cdot \left(\frac{\Gamma(n/2)}{\Gamma(n/2 + 1/2)} \right)^2,$$

and since $\Gamma(n/2+1/2)/\Gamma(n/2) < \sqrt{n/2}$, we obtain the following lower bound on the expected factor by which the approximation error is reduced in each step:

$$\mathsf{E}[R] > \frac{n-1}{2} \cdot \frac{2}{n} = 1 - \frac{1}{n}.$$

This lower bound holds for perfect line search and, as a consequence, also for any other line-search strategy. Thus this bound is universal for the class of hit-and-run direct-search methods.

To see how good this general lower bound on $\mathsf{E}[R]$ actually is, an upper bound on $\mathsf{E}[R]$ under the assumption of perfect line search would be nice. Using that $\Gamma(n) = (n-1)!$, $\Gamma(n/2) = (n-2)!! \cdot \sqrt{\pi}/2^{(n-1)/2}$, and $\Gamma(k+1/2) = (2k-1)!! \cdot \sqrt{\pi}/2^k$ (where $k!!$ is defined as $2 \cdot 4 \cdot 6 \cdots k$ for even k, and as $3 \cdot 5 \cdots k$ for odd k), the right-hand side of Equation 2 on the previous page can be estimated as follows:

$$\mathsf{E}[R] < \frac{2n-1}{2n} = 1 - \frac{1}{2n}.$$

In other words, for perfect line search, the expected factor by which the approximation error is reduced (in each step) is smaller than $1-0.5/n$. This shows that our general lower bound of $1-1/n$ on $\mathsf{E}[R]$ is actually pretty tight. All in all, we have proved the following result:

Theorem 1. *Consider the optimization of a function $f\colon \mathbb{R}^n \to \mathbb{R}$ with a unique optimum point $\boldsymbol{x}^*$. Then we have for $n \geq 4$:*

The (hypothetical) hit-and-run direct-search method which performs a perfect line search in each step converges linearly to $\boldsymbol{x}^*$ at an expected rate of $1-\beta/n$, where $0.5 < \beta < 1$ (and β may depend on n).

Independently of how a hit-and-run direct-search method performs the line searches, the expected factor by which the approximation error (i. e. the distance from $\boldsymbol{x}^*$) is reduced is larger than $1-1/n$ in each step. That is, if (at all) a hit-and-run direct-search method converges towards $\boldsymbol{x}^*$, then at best linearly at an expected rate larger than $1-1/n$.

The result on the (expected) factor by which the approximation error is reduced directly implies a bound on the (expected) spatial gain towards the optimum point $\boldsymbol{x}^*$. Therefore, let $d^{[i]}$ denote the approximation error (i. e. the distance from $\boldsymbol{x}^*$) after the ith step, and let $d^{[0]}$ denote the initial approximation error. For a fixed $d^{[i-1]}$, let $\Delta^{[i]} := d^{[i]} - d^{[i-1]}$ be defined as the random variable corresponding to the spatial gain towards $\boldsymbol{x}^*$ in the ith step. Then the above theorem says that in general, i. e. for any hit-and-run direct-search method, $\mathsf{E}\big[\Delta^{[i]}\big] < d^{[i-i]}/n$ in each step i. Moreover, for perfect line search, in each step $\mathsf{E}\big[\Delta^{[i]}\big] = \beta_{(n)} \cdot d^{[i-1]}/n$ for some function $\beta\colon \mathbb{N} \to (0.5, 1)$.

Let us stick with perfect line search in the following. Then the approximation error is non-increasing, i. e., $d^{[0]} \geq d^{[1]} \geq d^{[2]} \ldots$ (actually, $d^{[i+1]} < d^{[i]}$ with probability one, since the randomly chosen line lies in H with zero probability). Thus, in each step $\Delta^{[i]} < d^{[i-1]}/n \leq d^{[0]}/n$, and consequently, the number of steps necessary for an expected total gain of at least $d^{[0]}/2$ is larger than

$(d^{[0]}/2)/(d^{[0]}/n) = n/2$. However, in general, maximizing the *expected total gain* of a fixed number of steps need not necessarily result in minimizing the *expected number of steps* to realize a specified gain (for instance, to halve the approximation error). Nevertheless, $n/2$ will turn out to be a lower bound on the expected number of steps which are necessary to halve the approximation error. The proof is a straight-forward application of the following lemma, which is a modification of Wald's equation.

Lemma 2. *Let $X_1, X_2, \ldots$ denote random variables with bounded range and S the random variable defined by $S = \min\{\, t \mid X_1 + \cdots + X_t \geq g\}$ for a given $g > 0$. Given that S is a stopping time (i. e., the event $\{S = t\}$ depends only on $X_1, \ldots, X_t$), if $\mathsf{E}[S] < \infty$ and $\mathsf{E}[X_i \mid S \geq i] \leq u \neq 0$ for $i \in \mathbb{N}$, then $\mathsf{E}[S] \geq g/u$.*

(A proof can be found, e. g., in Jägersküpper, 2007.) We let X_i denote $\Delta^{[i]}$ and choose $g := d^{[0]}/2$. As we have just seen, $0 \leq \Delta^{[i]} \leq d^{[0]}$, and since in our scenario "$S \geq i$" merely means that the approximation error has not been halved in the first $i-1$ steps, actually $\mathsf{E}\big[\Delta^{[i]} \mid S \geq i\big] < d^{[0]}/n =: u$. Finally, we note that S is in fact a stopping time so that $g/u = n/2$ is indeed a lower bound on the expected number of steps to halve the approximation error (unless $\mathsf{E}[S]$ was infinite, in which case we would not need to prove a lower bound anyway). Due to the linearity of expectation, the expected number of steps to halve the approximation error $b \in \mathbb{N}$ times is lower bounded by $(n/2) + (b-1) \cdot (n/2-1)$, where the rightmost "$-1$" emerges because the last step within a halving-phase is also (and must be counted as) the first step of the following halving-phase. Thus, we have just proved the following result.

Theorem 3. *Let a hit-and-run direct-search method optimize a function in $\mathbb{R}^n$, $n \geq 4$, with a unique optimum. Let $b\colon \mathbb{N} \to \mathbb{N}$. For perfect line search the expected number of steps until the approximation error in the search space is less than a $2^{-b(n)}$-fraction of the initial one is lower bounded by $b(n) \cdot n/2 - b(n) + 1$.*

Now that we know that at least $n/2$ steps are necessary *in expectation* to halve the approximation error, we would like to know whether there is a good chance of getting by with considerably fewer steps. In fact, we want to show that there is almost no chance of getting by with a little fewer steps. Actually, we are going to prove that $0.4\,n$ steps suffice to halve the approximation error only with an exponentially small probability. Therefore recall the following notions and notations, where X and Y denote random variables:

- X *stochastically dominates* Y, in short "$X \succ Y$," if (and only if) $\forall a \in \mathbb{R}$: $\mathsf{P}\{X \leq a\} \leq \mathsf{P}\{Y \leq a\}$.
- If $X \succ Y$ as well as $Y \succ X$, i. e., $\forall a \in \mathbb{R}$: $\mathsf{P}\{X \leq a\} = \mathsf{P}\{Y \leq a\}$, then X and Y are equidistributed and we write "$X \sim Y$."

Obviously, stochastic dominance is a transitive relation, and it is readily seen that, if $X \succ Y$ and $\mathsf{E}[X]$ exists, then $\mathsf{E}[Y] \leq \mathsf{E}[X]$.

Theorem 4. *Let a hit-and-run direct-search method optimize a function in $\mathbb{R}^n$ with a unique optimum. Let $b\colon \mathbb{N} \to \mathbb{N}$ such that $b(n) = \mathrm{poly}(n)$. For perfect line search, with a very high probability of $1 - \exp(-\Omega(n^{1/3}))$ more than $b_{(n)} \cdot 0.4\,n$ steps are necessary until the approximation error is less than a $2^{-b(n)}$-fraction of the initial one.*

Proof. Assume that $\boldsymbol{x}^{[0]} \neq \boldsymbol{x}^*$. Because in each step perfect line search is performed, $\Delta^{[i]}/d^{[i-1]} \sim \Delta^{[j]}/d^{[j-1]}$ for $i, j \in \mathbb{N}$ (scale invariance) . Since moreover $d^{[0]} \geq d^{[1]} \geq d^{[2]} \ldots$, we have $\Delta^{[1]} \succ \Delta^{[2]} \succ \ldots$ for the single-step gains. Let $X_1, X_2, X_3, \ldots$ denote independent instances of the random variable $\Delta^{[1]}$. Then $\forall i \in \mathbb{N}\colon X_i \succ \Delta^{[i]}$, and hence $\sum_{i=1}^{k} \Delta^{[i]} \prec S_k := \sum_{i=1}^{k} X_i$. In less formal words: Adding up k independent instances of the random variable which corresponds the spatial gain in the first step results in a random variable (namely S_k) which stochastically dominates the random variable given by the total gain of the first k steps. The advantage of considering S_k instead of the "true" total gain of these steps is the following: S_k is the sum of independent random variables so that we can apply Hoeffding's bound. Namely, Hoeffding (1963, Theorem 2) tells us:

> Let $X_1, \ldots, X_k$ denote independent random variables, each with bounded range so that $a_i \leq X_i \leq b_i$ with $a_i < b_i$ for $i \in \{1, \ldots, k\}$. Let $S := X_1 + \cdots + X_k$. Then $\mathsf{P}\{S \geq \mathsf{E}[S] + x\} \leq \exp(-2x^2/\sum_{i=1}^{k}(b_i - a_i)^2)$ for any $x > 0$.

If the support of each random variable X_i is contained in $[a, b] \subset \mathbb{R}$, the upper bound becomes $\exp\left(-2 \cdot (x/(b-a))^2/k\right)$. So, let $k := 0.4n$ and $S := S_k$. Then $\mathsf{E}[S] = 0.4n \cdot \mathsf{E}\left[\Delta^{[1]}\right] \leq 0.4d^{[0]}$, and for the application of Hoeffding's bound we choose $x := 0.1d^{[0]}$, which yields an upper bound of $\exp(-0.05(d^{[0]}/(b-a))^2/n)$ on the probability that the approximation error is halved in $0.4n$ steps. We can choose $a := 0$ so that we obtain $\mathsf{P}\left\{X_1 + \cdots + X_k \geq d^{[0]}/2 \mid X_1, \ldots, X_k \leq b\right\} \leq \exp(-0.05(d^{[0]}/b)^2/n)$, where b is an upper bound on the gain towards the optimum point $\boldsymbol{x}^*$ in a step. Unfortunately, when substituting the trivial upper bound of $d^{[0]}$ for b, the upper bound on the probability becomes $\exp(-0.05/n)$, which tends to one as n grows. For $b := d^{[0]}/n^{2/3}$, however, we obtain (recall that k was chosen as $0.4n$)

$$\mathsf{P}\left\{X_1 + \cdots + X_k \geq d^{[0]}/2 \mid X_1, \ldots, X_k \leq d^{[0]}/n^{2/3}\right\} \leq \mathrm{e}^{-0.05\,n^{1/3}}.$$

Thus, if we can show that $\mathsf{P}\left\{X_i > d^{[0]}/n^{2/3}\right\} = \mathrm{e}^{-\Omega(n^{1/3})}$ in each of the $0.4n$ steps, we obtain (by an application of the union bound)

$$\mathsf{P}\left\{X_1 + \cdots + X_k \geq d^{[0]}/2\right\} \leq \mathrm{e}^{-0.05\,n^{1/3}} + 0.4n \cdot \mathrm{e}^{-\Omega(n^{1/3})} = \mathrm{e}^{-\Omega(n^{1/3})}.$$

Finally, by another application of the union bound, we obtain the theorem because $b(n) = \mathrm{poly}(n)$ implies $b(n) \cdot \mathrm{e}^{-\Omega(n^{1/3})} = \mathrm{e}^{-\Omega(n^{1/3})}$.

In other words, it remains to be shown that $\mathsf{P}\left\{\Delta^{[0]} > d^{[0]}/n^{2/3}\right\}$ is actually bounded above by $\mathrm{e}^{-\Omega(n^{1/3})}$. Therefore, recall Equation 1 on page 6. It tells

us that $d - d' = d \cdot (1 - \sqrt{1-(g/\ell)^2})$. As a consequence, $\mathsf{P}\{\Delta > d/n^{2/3}\}$ is equal to $\mathsf{P}\{1 - \sqrt{1-G^2} > 1/n^{2/3}\}$. Solving $1 - \sqrt{1-G^2} > 1/n^{2/3}$ for G yields $G > \sqrt{2/n^{2/3} + 1/n^{4/3}}$ so that that $\Delta > d/n^{2/3}$ actually implies $G > \sqrt{2}/n^{1/3}$. Since G's density is a non-increasing function in $[0,1]$,

$$\mathsf{P}\Big\{G > \sqrt{2}/n^{1/3}\Big\} = \int_{\sqrt{2}/n^{1/3}}^{1} (1-x^2)^{(n-3)/2}\,\mathrm{d}x < \left(1 - \frac{2}{n^{2/3}}\right)^{(n-3)/2}.$$

Since $(1-t/k)^k < \mathrm{e}^{-t}$ for $0 < t < k > 1$, we have $(1-2/n^{2/3})^{n^{2/3}} < \mathrm{e}^{-2}$, so that finally $\mathsf{P}\{\Delta > d/n^{2/3}\} < \mathsf{P}\{G > \sqrt{2}/n^{1/3}\} < \mathrm{e}^{-2\cdot((n-3)/2)/n^{2/3}} = \mathrm{e}^{-n^{1/3}+3/n^{2/3}}$. □

4 Discussion and Conclusion

Even though it is clear from intuition that the lower bounds presented in the two preceding theorems do not only hold for perfect line search but for any line-search strategy, they are formally proved only for perfect line search. Interestingly, we can easily show that our theorems hold independently of how the line searching is actually done: By induction over the number of steps i we show that the random variable which corresponds to the approximation error after i steps for a given line-search strategy stochastically dominates the random variable $d^{[i]}$ for perfect line search, which we considered in the proofs.

So, hit-and-run direct-search methods converge (at best and if at all) linearly with an expected rate larger than $1 - 1/n$. In simple words, the reason for this is that in high dimensions the randomly chosen direction is with a high probability "almost perpendicular" to the direction pointing directly towards the optimum point $\boldsymbol{x}^*$. For the further discussion, consider the simple toy problem of minimizing a quadratic form $\boldsymbol{x} \mapsto \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$, where the $n \times n$-matrix Q is positive definite. For this simple scenario, steepest descent converges at least linearly at a rate which is independent of the dimension n but which gets worse when the condition number of Q increases—when assuming a worst-case starting point. In the best case, however, steepest descent needs a single (perfect) line search to determine the optimum. Thus, for ill-conditioned quadratics, the performance of steepest decent heavily depends on the starting point. This is one reason why usually preconditioning is applied. Hypothetically assume for a moment the extreme of perfect preconditioning, so that $\boldsymbol{x}^\top \boldsymbol{I}\boldsymbol{x} = |\boldsymbol{x}|^2$ is to be minimized. Interestingly, the original evolution strategy from 1965 by Rechenberg/Schwefel (cf. the introduction), a very simple randomized direct-search method which belongs to the class of hit-and-run methods, needs $O(n)$ f-evaluations with very high probability to halve the approximation error in this scenario (Jägersküpper, 2003), showing that the very general lower bound obtained here can be met at least up to a constant factor. However, in this ideal scenario steepest descent needs a single (perfect) line search to find the optimum independently of the starting point. Now, as we consider black-box optimization, steepest descent must approximate the gradient. Even though the approximation of the gradient

may cost $\Theta(n)$ f-evaluations, a single line search in this approximate direction may yield a significantly larger gain towards the optimum—whereas a hit-and-run method needs at least $0.5n$ f-evaluations to halve the approximation error *in any case* (in expectation; $0.4n$ with very high probability). Thus, with a passable preconditioning, the approximation of the gradient should pay off—even though it costs a linear (in n) number of f-evaluations per step—so that it will likely be superior to hit-and-run into a random direction.

As it should have become clear from the discussion above, hit-and-run cannot be supposed to compete with methods which learn (and then utilize) second-order information like the well-known BFGS method or generalized conjugate gradient methods. Clearly, hit-and-run can make sense in real-world optimization when classical direct-search methods have turned out to fail, for instance, when the function to be optimized is highly multimodal such that gradient approximation is deceptive. However, as we have proved here, we should not expect such hit-and-run to be fast.

References

Bertsimas, D., Vempala, S. (2004): *Solving convex programs by random walks.* Journal of the ACM, 51(4):540–556.

Droste, S., Jansen, T., Wegener, I. (2006): *Upper and lower bounds for randomized search heuristics in black-box optimization.* Theory of Computing Systems, 39(4):525–544.

Dyer, M. E., Frieze, A. M., Kannan, R. (1989): *A random polynomial time algorithm for approximating the volume of convex bodies.* In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, 375–381.

Fogel, D. B. (editor) (1998): *Evolutionary Computation: The Fossil Record.* Wiley-IEEE Press.

Hoeffding, W. (1963): *Probability inequalities for sums of bounded random variables.* American Statistical Association Journal, 58(301):13–30.

Hooke, R., Jeeves, T. A. (1961): *"Direct search" solution of numerical and statistical problems.* Journal of the ACM, 8(2):212–229.

Jägersküpper, J. (2003): *Analysis of a simple evolutionary algorithm for minimization in Euclidean spaces.* In *Proc. 30th Int'l Colloquium on Automata, Languages and Programming (ICALP)*, vol. 2719 of *LNCS*, 1068–79, Springer.

Jägersküpper, J. (2007): *Algorithmic analysis of a basic evolutionary algorithm for continuous optimization.* Theoretical Computer Science, to appear, doi: 10.1016/j.tcs.2007.02.042.

Kolda, T. G., Lewis, R. M., Torczon, V. (2004): *Optimization by direct search: New perspectives on some classical and modern methods.* SIAM Review, 45(3):385–482.

Lovász, L., Vempala, S. (2006): *Hit-and-run from a corner.* SIAM Journal on Computing, 35(4):985–1005.

Nelder, J. A., Mead, R. (1965): *A simplex method for function minimization.* The Computer Journal, 7:308–313.

Nemirovsky, A. S., Yudin, D. B. (1983): *Problem Complexity and Method Efficiency in Optimization.* Wiley, New York.

Rechenberg, I. (1965): *Cybernetic solution path of an experimental problem.* Royal Aircraft Establishment, in Fogel (1998).

Schwefel, H.-P. (1965): *Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik.* Diploma thesis, Technische Universität Berlin.