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

General Lower Bounds for Randomized Direct Search with Isotropic Sampling

Jens Jägersküpper

No. CI-233/07

Technical Report ISSN 1433-3325 July 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.

General Lower Bounds for Randomized Direct Search with Isotropic Sampling

Jens Jägersküpper [1]
Universität Dortmund, Informatik 2, 44221 Dortmund, Germany

Abstract

The focus is on a certain class of randomized direct-search methods for optimization in (high-dimensional) Euclidean space, namely for minimization of a function $f\colon \mathbb{R}^n \to \mathbb{R}$, where f is given by an oracle, i. e. a black box for f-evaluations. The iterative methods under consideration generate a sequence of candidate solutions, where potential candidate solutions are generated by adding an isotropically distributed vector to the current candidate solution (possibly several times, to then choose one of these samples to become the next in the sequence of candidate solutions). This class of randomized direct-search methods covers in particular several *evolutionary algorithms.*

Lower bounds on the number of samples (i. e. queries to the f-oracle) are proved which are necessary to enable such a method to reduce the approximation error in the search space. The lower bounds to be presented do not only hold in expectation, but they are such that runtimes below these bounds are observed only with an exponentially small probability (in the search space dimension n). To derive such strong bounds, an appealingly simple, but nevertheless powerful method is applied: We think of the guided/directed random search as a selected fragment of a purely/obliviously random search. Interestingly, the lower bounds so obtained turn out to be tight (up to an absolute constant).

Key words: heuristic optimization, direct search, random search, probabilistic analysis, theory

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 $\mathbb{R}$, or it may also be a mixture (if S has more than one dimension). 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 of the search space gets higher and higher. This viewpoint is typical for analyses in computer science. It seems that optimization in continuous search spaces is not one of the core topics in computer science, though. In the domain of operations research and mathematical programming, however, focusing on how the optimization time scales with the search space's dimension seems rather uncommon. Usually, the performance of an iterative 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 i iterations. Then we have

Email address: JJ@Ls2.cs.uni-dortmund.de (Jens Jägersküpper).

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

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

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 iterations necessary to halve the approximation error (given by the Euclidean 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. 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 of the objective function, the gradient and the Hessian, 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, direct-search heuristics like simulated annealing, (randomized) local search, tabu search, 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 direct-search methods are the only option (unless the computer simulations allow for automatic/algorithmic differentiation). As the approximation of the gradient usually (i. e. forward or symmetric finite differences) involves n or even $2n$ f-evaluations, a single optimization step of a classical zeroth order-method is computationally expensive, in particular if f is given implicitly by simulations/experiments. In practical optimization this is often the case, and randomized search heuristics that abandon gradient approximation 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 neglected.

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 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 due to $\mathcal{F}$'s inherent black-box complexity rather than not due to the algorithm's simpleness. As mentioned above, the situation for heuristic optimization in continuous search spaces is different, especially with respect to randomized methods. The results to be presented here contribute to this emerging field of optimization theory.

2. The Framework for the Randomized Methods

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.; cf. Nocedal and Wright (2006, Ch. 3) for instance). As the approximation of the gradient usually costs at least 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 1950s/60s randomized search methods were proposed, for instance the so-called *evolution strategy* by Rechenberg (1965) and Schwefel (1965). In these early days, however, the focus was on whether or not (and, if so, under which conditions) a search heuristic would converge to a global optimum, cf. Rastrigin (1963) for instance. Here we are interested in how fast (w. r. t. the number of f-evaluations) a heuristic can approach the optimum point in the search space in principle, which can be considered a best-case scenario. Mainly, we want to answer this question: How many f-evaluations are necessary to halve the approximation error in the search space, i. e. to halve the Euclidean distance from the optimum point. As we consider randomized heuristics, this number is actually a random variable, and its distribution depends on various factors. Therefore, we consider the following framework of search heuristics: For a given initialization of the candidate solution $\boldsymbol{x} \in \mathbb{R}^n$, the number λ of samples per iteration, and the step-length parameter $\sigma \in \mathbb{R}_{>0}$, the following loop is performed until stopping is requested (externally):

(i) FOR $k := 1$ TO λ DO
Create a new search point $\boldsymbol{y}_{[k]} := \boldsymbol{x} + \boldsymbol{m} \in \mathbb{R}^n$, where the displacement vector $\boldsymbol{m}$ is drawn over $\mathbb{R}^n$ according to an isotropic distribution that depends only on σ.

(ii) Evaluate $f(\boldsymbol{y}_{[1]}), \ldots, f(\boldsymbol{y}_{[\lambda]})$ and decide which point $\boldsymbol{x}' \in \{\boldsymbol{x}, \boldsymbol{y}_{[1]}, \ldots, \boldsymbol{y}_{[\lambda]}\}$ becomes the next candidate solution; set $\boldsymbol{x} := \boldsymbol{x}'$.

(iii) Decide whether to increase, or to decrease, or to keep σ unchanged; adapt σ accordingly.

(iv) GOTO (i).

We are interested in how fast $\boldsymbol{x}$ can approach the optimum point $\boldsymbol{x}^*$. Therefore, we let "$\boldsymbol{x}^{[i]}$" denote the candidate solution *after* the ith iteration of the loop (so that "$\boldsymbol{x}^{[0]}$" denotes the initial search point). "$\sigma^{[i]}$" denotes the step-length parameter that is used *in* the ith iteration.

Concerning the generation of the samples in instruction (i), we formally need a mapping from $\mathbb{R}_{>0}$ into the set of isotropic distributions which tells us (given a specific σ) which isotropic distribution is to be used for the generation of the displacement vector $\boldsymbol{m}$. This mapping is fixed – just as λ, the number of samples per step. Note that a distribution over $\mathbb{R}^n$ is isotropic if it is invariant w. r. t. orthonormal transformations, i. e., a vector $\boldsymbol{m}$ is isotropically distributed over $\mathbb{R}^n$ if (and only if) for any fixed orthogonal matrix $\boldsymbol{R} \in \mathbb{R}^{n \times n}$, $\boldsymbol{Rm}$ follows the same distribution as $\boldsymbol{m}$, in short notation: $\boldsymbol{Rm} \sim \boldsymbol{m}$.

The selection of the next candidate solution $\boldsymbol{x}'$ in instruction (ii) may depend on the complete history of the optimization process, namely, in the ith iteration on the sequence $(\boldsymbol{x}^{[0]}, f(\boldsymbol{x}^{[0]})), \ldots, (\boldsymbol{x}^{[i-1]}, f(\boldsymbol{x}^{[i-1]}))$ associated with the trajectory of candidate solutions and also on all the discarded samples (including their f-values). Concerning the adaptation of σ in instruction (iii), the decision (whether to increase, or to decrease, or to keep σ unchanged) may depend on the complete history of the optimization process as well. The decision, however, must result in one of the following three outcomes: "increase", "decrease", or "keep." Depending solely on this outcome, σ is updated—possibly in a randomized manner. For instance, the σ-adaptation may be such that, if "increase" is the outcome, then σ is multiplied by a factor which is uniformly chosen from the interval $[1, 2]$. Naturally, the σ-adaptation need not necessarily be adaptive to the course of the optimization. It could follow a fixed schedule; σ could be kept fixed, or it could be multiplied by $1 - 0.1/n$ after each iteration, just for instance. Changing σ adaptively to the optimization process seems to make sense, though. How to do the σ-adaptation does obviously (also) depend on the family of isotropic distributions (parameterized in σ) which is used to randomly generate the displacement vector $\boldsymbol{m}$. In the original evolution strategy by Rechenberg/Schwefel, for instance, each component of $\boldsymbol{m}$ is i. i. d. according to a zero-mean normal distribution with variance σ^2. (It is easily checked that this

results in $\boldsymbol{m}$ being isotropically distributed.) Actually, sampling λ search points in each iteration i according to a normal distribution with mean $\boldsymbol{x}^{[i-1]}$ was already proposed by Brooks (1958)—without being specific about how to choose/adapt the variance. Rechenberg and Schwefel focused on how to update σ adaptively to the course of the optimization (*1/5-success-rule* resp. so-called σ*-self-adaptation*).

Also maintaining a set of μ candidate solutions (rather than only one), each with an associated σ, was proposed. Then, at the beginning of each iteration before instruction (i), one element $(\boldsymbol{x},\sigma)$ is chosen from the set of μ candidate solutions according to some rule; for instance uniformly at random. Moreover, in instruction (ii) μ candidate solutions must be chosen for the next iteration from the $\mu+\lambda$ search points (each with its associated σ). This overall selection makes it different from μ parallel/independent runs (which can also be realized by an appropriate rule). Before this modification will be discussed and analyzed, we focus on the framework given above in which a single candidate solution is iteratively optimized.

3. General Lower Bound

The aim is to prove a lower bound on the number of function evaluations (which equals λ times the number of iterations) which are necessary to reduce the approximation error in the search space, i. e. the Euclidean distance from the optimum search point $\boldsymbol{x}^*$ (unique by assumption). Namely, we will focus on how long it takes to halve the approximation error—even and in particular in the best case. In other words, the lower bound must hold for any algorithm that fits our framework as well as for any function to be optimized (with a unique optimum).

Obviously, the way how the sampling is done is crucial for a lower bound. Therefore, we consider isotropic sampling, and we will focus on some interesting facts about isotropic distributions in the following.

Proposition 1 *Let $\boldsymbol{u}$ be uniformly distributed over the unit hyper-sphere. A vector $\boldsymbol{x}$ is isotropically distributed over $\mathbb{R}^n$ if and only if there exists a non-negative random variable L, independent of $\boldsymbol{u}$, such that $\boldsymbol{x} \sim L \cdot \boldsymbol{u}$.*

A proof can be found, e. g., in Fang, Kotz, and Ng (1990, Sec. 2.1). The independence of the length distribution and the direction is crucial. It enables us to assume that the length of $\boldsymbol{m}$ is picked before $\boldsymbol{m}$'s (uniformly distributed) direction is picked (or vice versa), which will be very useful in the analysis. Obviously, adding two independently isotropically distributed vectors results in an isotropically distributed vector. Moreover:

Lemma 2 *Given isotropically distributed vectors $\boldsymbol{m}^{[k]} \sim L^{[k]} \cdot \boldsymbol{u}^{[k]}$ such that the directions $\boldsymbol{u}^{[k]}$ are independent (i. e., the lengths $L^{[k]}$ need not necessarily be independent). Then $\sum_k \boldsymbol{m}^{[k]}$ is isotropically distributed.*

PROOF. We consider two such vectors. Let $\boldsymbol{R}$ denote an arbitrary but fixed orthogonal matrix. Then
$\boldsymbol{R}(L^{[1]}\boldsymbol{u}^{[1]} + L^{[2]}\boldsymbol{u}^{[2]}) \sim \boldsymbol{R}(L^{[1]}\boldsymbol{u}^{[1]}) + \boldsymbol{R}(L^{[2]}\boldsymbol{u}^{[2]})$
(just because matrix multiplication is distributive). Since $L^{[i]}$ and $\boldsymbol{u}^{[i]}$ are independent, respectively, and since $\boldsymbol{R}\boldsymbol{u}^{[1]} \sim \boldsymbol{u}^{[1]}$ independently of $\boldsymbol{R}\boldsymbol{u}^{[2]} \sim \boldsymbol{u}^{[2]}$ (because the $\boldsymbol{u}^{[i]}$ are independently uniformly distributed over the unit hyper-sphere, respectively), we have

$$\begin{aligned}\boldsymbol{R}(L^{[1]}\boldsymbol{u}^{[1]}) + \boldsymbol{R}(L^{[2]}\boldsymbol{u}^{[2]}) &\sim L^{[1]}\boldsymbol{R}\boldsymbol{u}^{[1]} + L^{[2]}\boldsymbol{R}\boldsymbol{u}^{[2]} \\ &\sim L^{[1]}\boldsymbol{u}^{[1]} + L^{[2]}\boldsymbol{u}^{[2]}.\end{aligned}$$

All in all, $\boldsymbol{R}(L^{[1]}\boldsymbol{u}^{[1]} + L^{[2]}\boldsymbol{u}^{[2]}) \sim L^{[1]}\boldsymbol{u}^{[1]} + L^{[2]}\boldsymbol{u}^{[2]}$, precisely matching the definition of isotropy. □

In particular, the best-case distribution of $|\boldsymbol{m}|$ will be interesting—in dependence on the current candidate solution $\boldsymbol{x}$ and the location of the optimum $\boldsymbol{x}^*$, of course. Actually, due to the uniformity of the samples' directions, we can restrict ourselves to the distance $d^{[i]}$ between $\boldsymbol{x}^{[i]}$ and $\boldsymbol{x}^*$ after each iteration i. So, we are interested in how long it necessarily takes until $d^{[i]} \leq d^{[0]}/2$ (for the first time). Naturally, one might ask for the chance to halve the approximation error with a single isotropic sample. Therefore consider the hyper-plane H containing the current candidate solution $\boldsymbol{x}$ $(\neq \boldsymbol{x}^*)$ that is orthogonal to the line passing through $\boldsymbol{x}$ and $\boldsymbol{x}^*$. Assume that the isotropically distributed vector $\boldsymbol{m}$ happens to have the positive length ℓ. Then $\boldsymbol{y} := \boldsymbol{x} + \boldsymbol{m}$ is uniformly distributed upon the hyper-sphere centered at $\boldsymbol{x}$ with radius ℓ. Now consider the random variable

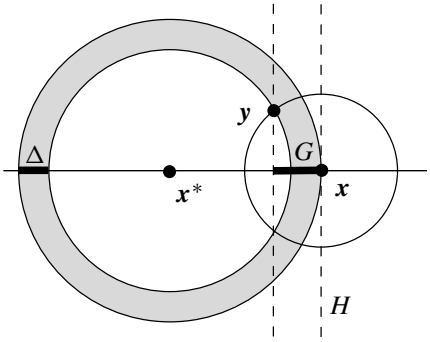
$$G_\ell(\boldsymbol{y}) := \begin{cases} \operatorname{dist}(\boldsymbol{y}, H) & \text{if } \boldsymbol{y} \text{ lies in the half-space} \\ & \text{w. r. t. } H \text{ containing } \boldsymbol{x}^* \\ -\operatorname{dist}(\boldsymbol{y}, H) & \text{otherwise} \end{cases}$$

which corresponds to the *signed distance* of the sample $\boldsymbol{x} + \boldsymbol{m}$ from the hyper-plane H (given $|\boldsymbol{m}| = \ell$). Obviously, the support of G_ℓ is $[-\ell, \ell]$. For $n \geq 4$ the density at g equals $(1-(g/\ell)^2)^{(n-3)/2}/(\ell \cdot \Psi)$, where

$\Psi = \int_{-1}^{1}(1-x^2)^{(n-3)/2}\,\mathrm{d}x$ (normalization), cf., e. g., Jägersküpper (2007a). Actually, for a distance of $d := \mathrm{dist}(\boldsymbol{x},\boldsymbol{x}^*)$ from the optimum we are interested in the random variable

$$\Delta_{d,\ell}(\boldsymbol{y}) := d - \mathrm{dist}(\boldsymbol{y},\boldsymbol{x}^*)$$

which corresponds to spatial gain towards the optimum $\boldsymbol{x}^*$ (given $|\boldsymbol{m}| = \ell$). The support of the random variable Δ is $[-\ell, \min\{\ell, 2d-\ell\}]$.



Simple geometry reveals (for any $\boldsymbol{y}$ with distance ℓ from $\boldsymbol{x}^*$) the interrelation

$$G_\ell(\boldsymbol{y}) = \Delta_{d,\ell}(\boldsymbol{y}) + \frac{\ell^2 - (\Delta_{d,\ell}(\boldsymbol{y}))^2}{2d}.$$

For halving the approximation error, i. e., $\Delta_{d,\ell} = d/2$, we obtain a corresponding distance from H of $\gamma := d\cdot 3/8 + \ell^2/(2d)$, where necessarily $\ell \in [d/2, d\cdot 3/2]$ because for $\ell < d/2$ as well as for $\ell > d\cdot 3/2$ a spatial gain of $d/2$ towards $\boldsymbol{x}^*$ is precluded. For such ℓ we thus have $\mathsf{P}\{\Delta_{d,\ell} \geq d/2\} = \mathsf{P}\{G_\ell \geq \gamma\}$, and the question is for which ℓ this probability is actually maximum. Since $\mathsf{P}\{G_\ell \geq \gamma\} = \int_{\gamma/\ell}^{1}(1-x^2)^{(n-3)/2}\,\mathrm{d}x \big/ \Psi$, we are interested in the ℓ for which $\gamma/\ell = \frac{3d}{8\ell} + \frac{\ell}{2d}$ is minimum. It is easily checked that this is actually the case for $\ell^* = d\sqrt{3/4}$. For this optimal length ℓ^*, the necessary signed distance from H solves to $\gamma^* := d\cdot 3/4$, so that $\gamma^*/\ell^* = \sqrt{3/4}$. Hence, (for $n \geq 4$)

$$\begin{aligned}
\max_\ell \mathsf{P}\{\Delta_{d,\ell} \geq d/2\} = \mathsf{P}\{G_{\ell^*} \geq \gamma^*\} &= \int_{\sqrt{3/4}}^{1}(1-x^2)^{(n-3)/2}\,\mathrm{d}x \Big/ \Psi \\
&< (1-\sqrt{3/4})\cdot(1-3/4)^{(n-3)/2}/\,\Psi \\
&< 0.134\cdot 2^{-(n-3)}/\,\Psi \\
&< 2^{-n}\cdot 0.43\sqrt{n-1}
\end{aligned}$$

since $\Psi = \sqrt{\pi}\cdot\Gamma((n-1)/2)/\,\Gamma(n/2) \geq \sqrt{\pi 2/(n-1)}$, where Γ denotes the well-known gamma function. Since this upper bound on the success probability holds when $|\boldsymbol{m}|$ is concentrated on the optimal length ℓ^*, it actually holds for any isotropic distribution of $\boldsymbol{m}$ so that we obtain the following result.

Lemma 3 *Let $\boldsymbol{x}^* \in \mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{x}^*\}$. Let $\boldsymbol{m}$ be (arbitrarily) isotropically distributed over $\mathbb{R}^n$. Then $\mathsf{P}\{\mathrm{dist}(\boldsymbol{x}+\boldsymbol{m},\boldsymbol{x}^*) \leq \mathrm{dist}(\boldsymbol{x},\boldsymbol{x}^*)/2\} < 2^{-n}\cdot 0.43\sqrt{n-1}$ for $n \geq 4$.*

Though it is no surprise that the chance of halving the approximation error with a single isotropic sample drops when the dimensionality increases, we now know a concrete (exponentially small) upper bound on that probability. And indeed, this upper bound will enable us to also obtain an upper bound on the success probability within multiple iterations of an optimization heuristic that fits our framework.

The idea behind this bound is the "curse of dimensionality" in $\mathbb{R}^n$. Therefore, consider for a moment k independent samples, for each of which even the optimal isotropic distribution may be assumed. Then for $k := \mathrm{e}^{0.69n}$, the probability that at least one of these samples halves the approximation error is bounded above by $k\cdot 2^{-n}\cdot 0.43\sqrt{n-1} = \mathrm{e}^{0.69n}\cdot\mathrm{e}^{-n\cdot\ln 2}\cdot 0.43\sqrt{n-1} < \mathrm{e}^{-0.003n+\ln\sqrt{n}} = \mathrm{e}^{-\Omega(n)}$. In other words, even an exponential number of $\mathrm{e}^{0.69n}$ samples (with optimal isotropic distribution) result in an exponentially small success probability of $\mathrm{e}^{-\Omega(n)}$.

Naturally, a reasonable heuristic does not sample around the initial candidate solution $\boldsymbol{x}^{[0]}$ all the time, but tries to iteratively approach the optimum using the information gathered by evaluating the function f to be optimized at the sampled points and selecting the most promising one of the $1+\lambda$ points in each iteration. Thus, the search is guided by the selection in instruction (ii) (unless f is a. e. constant). This selection, however, merely means that search paths that do not seem promising are no longer followed (pruned). One may easily imagine that also these search paths would be followed (in addition to the promising ones, of course).

In the following, we modify the search algorithm such that we end up with a search procedure which is independent of the function to be optimized. Consider the original algorithm after initialization, i. e., an initial search point $\boldsymbol{x}^{[0]}$ and an initial $\sigma^{[0]}$ are given. In the first step, λ points are sampled, each by adding an isotropically distributed vector (the distribution of which depends solely on $\sigma^{[0]}$) to $\boldsymbol{x}^{[0]}$. In the modified algorithm, we now do *not* select one of the $1+\lambda$

points, yet keep all $1+\lambda$ search points in the set $P^{[1]}$, where $P^{[0]} := \{(\boldsymbol{x}^{[0]}, \sigma^{[0]})\}$. At the end of the first iteration σ may be up- or down-scaled—depending on the f-values of $\boldsymbol{x}^{[0]}$ and the λ samples. Thus, to also get rid of this f-dependency, in the second step of the modified algorithm for each of the $1+\lambda$ points in $P^{[1]}$ three samples a drawn: one without changing σ, one with an up-scaled σ, and one with a down-scaled σ. Again we keep all $(1+\lambda)\cdot 3\lambda$ newly generated points (each associated with the σ that was used to sample this search point). Consequently, we have $(1+\lambda)+(1+\lambda)\cdot 3\lambda = (1+\lambda)(1+3\lambda)$ search points after the second step in the set $P^{[2]}$. Repeating this procedure, after i iterations the set $P^{[i]}$ is generated containing

$$(1+\lambda)(1+3\lambda)^{i-1} \leq (1+3\lambda)^i = \mathrm{e}^{i\cdot\ln(1+3\lambda)}$$

search points (each with an associated σ). The crucial point is that $P^{[i]}$ is built without any dependency on the function f to be optimized, and that any trajectory of the original algorithm emerges at least as probable in this modified search procedure. More formally: Let $S \subset \mathbb{R}^n$ denote an arbitrary Borel set. Then the probability that $P^{[i]}$ hits S (i. e., $S \cap P^{[i]} \neq \emptyset$) is an upper bound on the probability that the search point $\boldsymbol{x}^{[i]}$ generated within i iterations by the original algorithm is in S, namely, $\forall i \in \mathbb{N}\colon \mathsf{P}\{\boldsymbol{x}^{[i]} \in S\} \leq \mathsf{P}\{P^{[i]} \cap S \neq \emptyset\}$. This is readily proved by induction on the number of steps; it is crucial that the initialization is done in the same way for both search procedures, of course.

Since each search point $\boldsymbol{x} \in P^{[i]}$ is generated by successively adding i isotropically distributed vectors to the initial search point, Lemma 2 tells us that $\boldsymbol{x}$ is indeed isotropically distributed around the initial search point $\boldsymbol{x}^{[0]}$. We do not know the (distribution of the) distance between $\boldsymbol{x}$ and $\boldsymbol{x}^{[0]}$, yet this does not matter—as we may assume the best case.

We choose the target set S as the hyper-ball containing all search points with a distance of at most half the initial distance from $\boldsymbol{x}^*$. Since $\mathsf{P}\{\boldsymbol{x} \in S\} < 2^{-n}\sqrt{n}$ for each $\boldsymbol{x} \in P^{[i]}$ according to Lemma 3, by the union bound

$$\begin{aligned}\mathsf{P}\{P^{[i]} \cap S \neq \emptyset\} &< \#P^{[i]} \cdot 2^{-n}\sqrt{n} \\ &\leq \mathrm{e}^{\ln(1+3\lambda)\cdot i - n\ln 2 + \ln\sqrt{n}}\end{aligned}$$

for $n \geq 4$. Then choosing $i := 0.69n/\ln(1+3\lambda)$ finally yields an upper bound of $\mathrm{e}^{-0.003n+\ln\sqrt{n}} = \mathrm{e}^{-\Omega(n)}$ on the probability that after $0.69n/\ln(1+3\lambda)$ steps $P^{[i]}$ contains a search point that lies in S (note that $P^{[i]} \supset P^{[i-1]} \supset \cdots \supset P^{[0]}$). In other words, more than $0.69n/\ln(1+3\lambda)$ steps are necessary with probability $1-\mathrm{e}^{-\Omega(n)}$ to halve the approximation error. Since adding up a polynomial number of error probabilities each of which is $\mathrm{e}^{-\Omega(n)}$ results in a total error probability that is still $\mathrm{e}^{-\Omega(n)}$ (union bound again), we obtain the following lower-bound result:

Theorem 4 *Let a heuristic that fits our framework optimize some function $f\colon \mathbb{R}^n \to \mathbb{R}$, and let $\boldsymbol{x}^* \in \mathbb{R}^n$ be some fixed point (for instance the/an optimum). Let $b : \mathbb{N} \to \mathbb{N}$ such that $b(n) = \mathrm{poly}(n)$. Then, for $n \geq 4$ and given that $d := \mathrm{dist}(\boldsymbol{x}^{[0]}, \boldsymbol{x}^*) > 0$, with probability $1-\mathrm{e}^{-\Omega(n)}$ the number i of iterations until $\mathrm{dist}(\boldsymbol{x}^{[i]}, \boldsymbol{x}^*) \leq d/2^{b(n)}$ (for the first time) is larger than $b(n)\cdot 0.69n/\ln(1+3\lambda)$.*

Since $2^{-b} = \varepsilon$ for $b = \ln(1/\varepsilon)/\ln(2)$, we directly obtain:

Corollary 5 *In the setting of the preceding theorem, more than $\lambda\cdot 0.99n\cdot\ln(1/\varepsilon)/\ln(1+3\lambda)$ samples are necessary with probability $1-\mathrm{e}^{-\Omega(n)}$ until the approximation error is at most an ε-fraction of the initial one, where $2 \leq 1/\varepsilon = 2^{\mathrm{poly}(n)}$.*

Note that, since $1-\mathrm{e}^{-\Omega(n)} = \Omega(1)$, this directly implies a lower bound of $\Omega(n\cdot\lambda/\ln(1+\lambda))$ on the *expected* number of samples necessary to halve the approximation error. Further note that the λ samples in an iteration are a. s. mutually distinct if $|\boldsymbol{m}| > 0$ a. s., so that a. s. λ f-evaluations are indeed necessary. (A setting in which $\mathsf{P}\{|\boldsymbol{m}| = 0\} > 0$ does not make much sense.)

Optimizing a Set of μ Candidate Solutions

We now turn our attention to the modification of the framework described in the last paragraph of Section 2: A set of μ candidate solutions is maintained (each with an associated σ), where $\mu = \mathrm{poly}(n)$ may depend on the search space dimension n but is kept fixed during a run (just as the number λ of samples per iteration). In each iteration one of the μ candidate solutions is chosen as the basis for the λ samples. Let p denote an a priori *upper* bound (as small as possible, though) on the probability that a particular candidate solution of the μ is selected (when disregarding the other $\mu-1$) to become the basis. For instance, if (in each iteration) one of the μ candidate solutions is picked uniformly at random, we can choose $p := 1/\mu$, and if it is chosen rank-proportionally at random, we can choose $p := \mu/\sum_{r=1}^{\mu} r = 2/(\mu+1)$. If the one with the best f-value

is chosen, however, we must choose $p := 1$ (as for any deterministic rule). Note that necessarily $p \geq 1/\mu$.

Again we analyze a modified algorithm in which we keep all sampled points (each associated with the σ used for its sampling). In each iteration i, a subset of $\kappa^{[i]} := \lceil p \cdot \#P^{[i-1]} \rceil$ elements is chosen uniformly at random from $P^{[i-1]}$, where $P^{[0]}$ consists of the μ initial candidate solutions of the original algorithm. In the modified algorithm, for each of these κ search points, 3λ samples are drawn in the same way as for the optimization of a single candidate solution (cf. above). For each element in $P^{[i-1]}$ the a priori probability to become a basis of sampling in the ith iteration equals $\kappa^{[i]}/\#P^{[i-1]} = \lceil p \cdot \#P^{[i-1]} \rceil / \#P^{[i-1]} \geq p$. Recall that p is an *upper* bound for original process on the a priori probability that a particular of the μ candidate solutions is chosen as the basis in an iteration. Thus, the size of P grows as follows: $\#P^{[i]} = \#P^{[i-1]} + 3\lambda \cdot \lceil p \cdot \#P^{[i-1]} \rceil$.

In other words, (for the modified algorithm) in iteration i the number of elements in P grows by the factor

$$1 + 3\lambda \cdot \frac{\lceil p \cdot \#P^{[i-1]} \rceil}{\#P^{[i-1]}} < 1 + 3\lambda \cdot (p + 1/\#P^{[i-1]}) \leq 1 + 3\lambda \cdot 2p$$

since $p \geq 1/\mu$ anyway and $\#P^{[i-1]} \geq \mu$. Consequently, $\#P^{[i]} \leq \mu \cdot (1 + 6\lambda p)^i = \mathrm{e}^{i \cdot \ln(1+6\lambda p) + \ln \mu}$. Again by induction on the number of iterations, we readily see that any trajectory of search points emerges in the modified algorithm as least as probable as in the original algorithm. For any Borel set $S \subset \mathbb{R}^n$ and any $i \in \mathbb{N}$, the probability that the original algorithm generates a point in S within i iterations is at most $\mathsf{P}\{P^{[i]} \cap S \neq \emptyset\}$ (given the same initialization, of course). Moreover, any point in $P^{[i]}$ is isotropically distributed around one of the μ initial search points. As a consequence, the success probability of halving the approximation error within i iterations (i. e., $\min\{\mathrm{dist}(\boldsymbol{x}, \boldsymbol{x}^*) \mid \boldsymbol{x} \in P^{[i]}\} \leq \min\{\mathrm{dist}(\boldsymbol{x}, \boldsymbol{x}^*) \mid \boldsymbol{x} \in P^{[0]}\}/2$) is bounded above by

$$\#P^{[i]} \cdot 2^{-n} \sqrt{n} \leq \mathrm{e}^{i \cdot \ln(1+6\lambda p) + \ln \mu - n \ln 2 + \ln \sqrt{n}}.$$

Choosing $i := 0.69n/\ln(1 + 6\lambda p)$ yields a success probability of less than $\mathrm{e}^{-0.003n + \ln \mu + \ln \sqrt{n}} = \mathrm{e}^{-\Omega(n)}$ since $\mu = \mathrm{poly}(n)$. All in all, we have proved the following result:

Theorem 6 *Let a heuristic that fits our modified framework (with a set of μ candidate solutions) optimize some function $f : \mathbb{R}^n \to \mathbb{R}$, and let $\boldsymbol{x}^* \in \mathbb{R}^n$ be some fixed point (for instance the/an optimum). Let $b : \mathbb{N} \to \mathbb{N}$ such that $b(n) = \mathrm{poly}(n)$. Then, for $n \geq 4$ and given that each initial candidate solution has distance at least d from $\boldsymbol{x}^*$, with probability $1 - \mathrm{e}^{-\Omega(n)}$ more than $b(n) \cdot 0.69n/\ln(1 + 6p\lambda)$ iterations are necessary until a search point with a distance of at most $d/2^{b(n)}$ from $\boldsymbol{x}^*$ is generated (for the first time), where p is an upper bound on the a priori probability that in an iteration a particular of the μ candidate solutions is selected.*

Since $\ln(1 + x) \leq x$ (i. e., $1/\ln(1 + x) \geq 1/x$) for $x > 0$, we directly obtain the following.

Corollary 7 *The lower bound in the preceding theorem implies that more than $b(n) \cdot 0.115n/(p\lambda)$ samples are necessary; for uniform selection, namely $p = 1/\mu$, more than $b(n) \cdot 0.115n \cdot \mu/\lambda$ samples, and for rank-proportional selection, namely $p = 2/(\mu + 1)$, more than $b(n) \cdot 0.057n \cdot \mu/\lambda$ samples are necessary.*

Thus, in cases when $p = O(1/\mu)$ as n grows, the lower bound on the number of samples necessary to halve the approximation error is $\Omega(n \cdot \mu/\lambda)$ as n grows. To put it more concise: The (lower bound on the) number of samples necessary to halve the approximation error grows linearly in the number of candidate solutions which are maintained (in such cases). Since "$\ln(1 + x) \leq x$" is a good estimate only for small x, the lower bounds from the preceding corollary are good in particular when λ, μ, and the randomization of the selection rule are such that $\lambda p \to 0$ as n grows, implying that $\lambda = o(\mu)$ as n grows, which implies that μ grows with n.

4. Discussion and Conclusion

The lower bound of $0.69n/\ln(1+3\lambda)$ on the number of iterations necessary to halve the approximation error from Theorem 4 holds independently of how the next candidate solution $\boldsymbol{x}'$ is chosen in instruction (ii), and for $\lambda := 1$, this lower bound becomes $0.497n$. For instance, the search point with the best f-value could be chosen as the next candidate solution, yet also a Metropolis-like selection which accepts a worse sample with a probability of, say, 5% would also be covered. The reason for this is simple: In the modified search procedure, which is used in the analysis, all samples that are ever generated are kept in the (exponentially growing) set P anyway. As a consequence, also a simulated annealing-like selection, where the probability of accepting a worse sample as the next candidate solution depends on how much worse it is, would be covered. So, the selection

of the next candidate solution does not influence our lower bound. The adaptation of σ, however, is more critical in this respect. In the proof of the lower bound we used that at the end of each iteration there are exactly three alternatives for the adaptation, which may be called "increase", "keep", "decrease." We could allow more alternatives, though. If there were, say, **five** alternatives for the σ-adaptation, the lower bound on the number of iterations to halve the approximation error in the search space would become $0.69n/\ln(1+\mathbf{5}\lambda)$, and, when maintaining a set of μ candidate solutions, $0.69n/\ln(1+(2\cdot\mathbf{5})p\lambda)$, where p is an a priori bound on the probability that a particular of the μ is chosen in an iteration as the basis for the λ samples.

The lower bounds presented cover a very large class of randomized direct search methods—which use isotropic sampling. As, moreover, the analysis uses an exponentially growing set of samples, one may question the quality of the lower bounds presented. Actually, the lower bound from Theorem 4 is tight (up to a constant factor), and the one from Theorem 6 is tight (up to a constant factor) for $\lambda := 1$ and random selection (i. e., $p = 1/\mu$) at least.

Therefore, consider the scenario in which the function f is defined as the Euclidean distance from a fixed point $\boldsymbol{x}^* \in \mathbb{R}^n$, the unique global minimum. As mentioned in the introduction, in the so-called *(1+1) evolution strategy* by Rechenberg, in the ith iteration a single search point $\boldsymbol{y}$ is sampled according to a multivariate normal distribution with variance σ^2 and mean $\boldsymbol{x}^{[i-1]} \in \mathbb{R}^n$. If $f(\boldsymbol{y}) \leq f(\boldsymbol{x}^{[i-1]})$, then $\boldsymbol{y}$ becomes the next candidate solution, otherwise $\boldsymbol{x}^{[i]} := \boldsymbol{x}^{[i-1]}$. As shown by Jägersküpper (2007a), when the so-called *1/5-success-rule* proposed by Rechenberg is used for the σ-adaptation, then $O(b\cdot n)$ iterations/samples/f-evaluations suffice with a very high probability of $1-\mathrm{e}^{-\Omega(n^{1/3})}$ to reduce the approximation error below a 2^{-b}-fraction of the initial one—given an appropriate initialization of $\sigma^{[0]}$, of course. Jägersküpper (2007b, Ch. 5) generalizes this upper bound result: When λ i. i. d. such samples are drawn per iteration, $O(b\cdot n/\ln(1+\lambda))$ iterations suffice w. v. h. p. when a slightly modified σ-adaptation is used. Moreover, a so-called $(\mu+1)$ evolution strategy is investigated, which uses random selection among the μ candidate solutions, and indeed $O(b\cdot\mu\cdot n)$ iterations suffice w. v. h. p.

Thus, at least in the cases mentioned above, the lower bounds presented here are tight up to a constant factor. In particular, the lower bounds are such that fewer iterations/samples suffice only with an exponentially small (in n) probability. As we have seen, we trivially obtain lower bounds on the expected number of iterations/samples of the same order. Interestingly, the method by which we obtained these strong lower bounds seems quite simple. We thought of the directed/guided random search as a selected fragment of a non-guided/oblivious random search to bound the success probability from above. This idea may also work in other randomized optimization scenarios, possibly also in randomized combinatorial optimization.

References

Brooks, S. H. (1958): *A discussion of random methods for seeking maxima*. Operations Research, 6(2):244–251.

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.

Fang, K.-T., Kotz, S., Ng, K.-W. (1990): *Symmetric multivariate and related distributions*, vol. 36 of *Monographs on statistics and applied probability*. Chapman & Hall, London.

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

Jägersküpper, J. (2007a): *Algorithmic analysis of a basic evolutionary algorithm for continuous optimization*. Theoretical Computer Science, 379(3):329–347.

Jägersküpper, J. (2007b): *Probabilistic analysis of Evolution Strategies using isotropic mutations*. Dissertation, Universität Dortmund, Germany.

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.

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.

Nocedal, J., Wright, S. (2006): *Numerical Optimization*. Springer, 2nd edn.

Rastrigin, L. A. (1963): *The convergence of the random search method in extremal control of a many-parameter system*. Automation and Remote Control, 24:1337–42.

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

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