Handling Expensive Optimization with Large Noise

Rémi Coulom, Philippe Rolet, Nataliya Sokolovska, Olivier Teytaud

Handling Expensive Optimization with Large Noise

Rémi Coulom
Grappa, SequeL, Université
Charles-de-Gaulle,
Lille, France
Remi.Coulom@univ-
lille3.fr

Philippe Rolet
TAO-INRIA,LRI, UMR 8623
CNRS - Université Paris-Sud,
Orsay, France
rolet@lri.fr

Nataliya Sokolovska
TAO-INRIA,LRI, UMR 8623
CNRS - Université Paris-Sud,
Orsay, France
nataliya@lri.fr

Olivier Teytaud
TAO-INRIA,LRI, UMR 8623
CNRS - Université Paris-Sud,
Orsay, France
teytaud@lri.fr

ABSTRACT

This paper exhibits lower and upper bounds on runtimes for expensive noisy optimization problems. Runtimes are expressed in terms of number of fitness evaluations. Fitnesses considered are monotonic transformations of the *sphere* function. The analysis focuses on the common case of fitness functions quadratic in the distance to the optimum in the neighbourhood of this optimum—it is nonetheless also valid for any monotonic polynomial of degree $p > 2$. Upper bounds are derived via a bandit-based estimation of distribution algorithm that relies on *Bernstein races* called R-EDA. It is an evolutionary algorithm in the sense that it is based on selection, random mutations, with a distribution (updated at each iteration) for generating new individuals. It is known that the algorithm is consistent (i.e. it converges to the optimum asymptotically in the number of examples) even in non-differentiable cases. Here we show that: (i) if the variance of the noise decreases to 0 around the optimum, it can perform optimally for quadratic transformations of the norm to the optimum, (ii) otherwise, it provides a slower convergence rate than the one exhibited empirically by an algorithm called Quadratic Logistic Regression (QLR) based on surrogate models—although QLR requires a probabilistic prior on the fitness class.

Categories and Subject Descriptors

F.m [**Theory of Computation**]: Miscellaneous; I.2.8 [**Artificial Intelligence**]: Problem Solving, Search

General Terms

Theory

Keywords

Noisy optimization, Bernstein races

1. INTRODUCTION

The following work deals with expensive noisy optimization. Noisy means that the result of a fitness evaluation at a given point is a random variable, whose probability distribution depends only on the location of the point—this noise model will be detailed in Section 2, as well as the class of fitnesses that we address in the paper. Expensive means that each fitness call is considered costly: for example, evaluating the fitness of an individual might involve the building and testing of a prototype, or hours of simulations on a computer or on a grid. Therefore, an expensive optimization algorithm's performance is measured by the number of fitness calls required to find the optimum with a given precision, rather than considering the computational time required by the algorithm to function.

A practical example of such a framework is searching for the parameters of an algorithm that minimize its probability of failure: a fitness call is then a Bernouilli random variable, resulting for a given parameter vector in a success with probability p and in a failure with probability $1 - p$. Each fitness call implies running the algorithm, and as such is quite costly in time.

State of the art

Using evolutionary algorithms and Estimation of Distribution algorithms (EDAs) to deal with noisy fitnesses is a topic that has been substantially discussed in the literature. Notably, many question the idea that repeatedly evaluating the same points in order to average values and decrease noise variance is effective, as compared to, for instance, simply increasing the population size [13, 7, 14, 1, 3]. A brief survey can be found in [23], where it has been shown that averaging can be efficient when used in the framework of *multi-armed bandits* (following ideas of [18]) and *races*. Specifically, it is proved that an EDA using Bernstein races to choose the number of evaluations of a given point reaches an optimal convergence rate for some noise models.

When dealing with noisy optimization, it is important to distinguish cases in which the variance of the noise decreases to zero near the optimum—which we will refer to as the *small noise* assumption hereafter—and cases where it does

not—*large noise* assumption (see section 2 for more details on noise settings). Small noise has been tackled in [20] for a quite restricted noise model. [2] has then shown that in case of large noise, all usual step-size adaptation rules diverge or stop converging: the usual behavior of evolutionary algorithms for models with large noise is that they stop converging as they get too close to the optimum, and then keep a residual error, with a step-size which does not decrease to zero[1].

[23] and [24] tackle cases of large noise, but only for fitness functions of the form $x \mapsto \lambda \|x - x^*\| + c$—excluding the common case of quadratic (or higher-order polynomial) fitness functions. Furthermore, classical algorithms for noise handling such as Uncertainty Handling for Covariance Matrix Adaptation (UH-CMA), empirically quite efficient for small noise, are unfortunately not yet stable enough to deal with large noise cases: for the Scaled-Translated sphere (STS) model presented below, UH-CMA does not converge. Consistently with these results, [26] has shown that fast convergence involves a number of evaluations running to infinity with the number of iterations. This was further developed in [23, 24] with both lower bounds and algorithms reaching the bound in many cases. However, the natural case of fitnesses that are quadratic in the distance to the optimum was not covered. In the following, we show that:

- the Estimation of Distribution Algorithm defined in [24] and recalled in Algorithm 3, based on a Race (termed R-EDA) has good theoretical guarantees (e.g. outperforming UH-CMA), for both small noise and large noise scenarios[2];

- for $p = 2$, R-EDA is empirically outperformed in case of large noise by surrogate models such as Quadratic Logistic Regression (QLR), that fits a quadratic model using a Bayesian prior;

- R-EDA also converges at a controlled rate for polynomial functions of the distance to the optimum.

Note that R-EDA has first been used in [24], and has not been modified for this work: all positive properties of R-EDA are preserved, in particular the convergence in many difficult cases, including optimality for fitnesses f such that $f(x) = c + \Theta(\|x - x^*\|)$, i.e. functions that behave similarly to a translated sphere function in the neighborhood of the optimum x^*.

2. FRAMEWORK

In this section, our framework for expensive noisy optimization is introduced.

The optimization framework is described in Algorithm 1. This is a black-box optimization framework: the algorithm

---

[1]It remains large even w.r.t. computers' numerical accuracy of zero.

[2]Our R-EDA algorithm, on the other hand, is not supposed to be a practical algorithm; it is here for showing complexity upper bounds, and to show that these complexity upper bounds can be reached by evolutionary algorithms with races.

Algorithm 1 Noisy optimization framework. Opt is an optimization heuristic: it takes as input a sequence of visited points and their binary, noisy fitness values, and outputs a candidate optimum, that is a points of the domain such that $f(x, t)$ is as small as possible. This point is the point whose fitness is asked next. Opt is successful on target f parameterized by t and random noise θ if $Loss(t, \theta, Opt)$ is small.

Parameters: N, number of fitness evaluations; t, unknown element of X.
θ: random state of the nature $\in [0, 1]^{\mathbb{N}}$; each coordinate θ_i for $i \in \{1, 2, \dots\}$ is uniformly distributed in $[0, 1]$.
for $n \in [[0, N-1]]$ **do**
  $x_{n+1}^{t,\theta} = Opt(x_1^{t,\theta}, \dots, x_n^{t,\theta}, y_1^{t,\theta}, \dots, y_n^{t,\theta})$
  $y_{n+1}^{t,\theta} = (f(x_{n+1}^{t,\theta}, t) < \theta_{n+1})?1 : 0$ // Return noisy fitness $\sim \mathcal{B}(f(x_{n+1}^{t,\theta}, t))$
end for
$Loss(t, \theta, Opt) = d(t, x_N^{t,\theta})$

---

can request the fitness values at any chosen point, and no other information on the fitness function is available. We consider a fitness function f parameterized by the (unknown) location of its optimum, t. The noise is accounted for by a random variable $\theta \in [0, 1]^{\mathbb{N}}$; each coordinate θ_i for $i \in \{1, 2, \dots\}$ is uniformly distributed in $[0, 1]$. The goal is to find the optimum t of $f(., t)$, by observing noisy measurement of f at x_i. Measurements are random variables $F_t(x_i)$ with law $\mathcal{L}$ in $[0, 1]$. They satisfy $\mathbb{E}[F_t(x_i)] = f(x_i, t)$. For the proof of the lower bound, the law of random variable $F_t(x_i)$ is Bernouilli, with parameter $f(x_i, t)$ as shown in Algorithm 1. This fits applications based on highly noisy optimization, such as games: let x be a parameter of a game strategy, that we wish to set at its best value; a noisy observation is a game against a baseline, resulting either in a win or in a loss; the aim is to find the value of x maximizing the probability of winning. Usual viability problems or binary control problems tackled by direct policy search also involve this kind of optimization.

We are interested in the number of requests needed for an optimization algorithm to find optimum t with precision ϵ and confidence $1 - \delta$; $\epsilon = \|x_n - t\|$ is the Euclidian distance between t and the output x_n of the algorithm after n fitness calls. The paper focuses on fitnesses of the form $(x, t) \mapsto c + \lambda \|x - t\|^p$, referred to as the Scaled-Translated sphere (STS) model. It is more general than the STS model of [24] which addresses only $p = 1$. In the following, t is not handled stochastically, i.e. the lower bounds are not computed in expectation w.r.t. all the possible fitness functions yielded by different values of t. Rather, we will consider the worst case on t. Therefore the only random variable in this framework is θ, accounting for noise in fitness measurements, and all probability / expectation operators are w.r.t. θ. For simplicity, we considered only deterministic optimization algorithms; the extension to stochastic algorithms is straightforward by including a random seed of the algorithm in θ.

In the following, $\tilde{O}$ means that logarithmic factors in ϵ are neglected. In all the paper, $[[a, b]] = \{a, a+1, a+2, \dots, b\}$.

Races

The algorithm used to prove upper bounds on convergence rates is based on Bernstein confidence bounds. It is a variation of the well-known Hoeffding bounds [19] (aimed at quantifying the discrepancy between an empirical mean and an expectation for bounded random variables), which takes variances into account [9, 5, 6]. It is therefore tighter in some settings. A detailed survey of Hoeffding, Chernoff and Bernstein bounds is beyond the scope of this paper; we will only present the Bernstein bound, within its application to *races*. A *race* between two or more random variables aims at distinguishing with high confidence random variables with better expectation from those with worse expectation. Algorithm 2 is a *Bernstein race* applied to distinct points x_i of a domain X—the 3 random variables are $F_t(x_i)$, the goal is to find a good point and a bad point such that we are confident that the good one is closer to the optimum than the bad one.

It is crucial in this situation to ensure that there exist i, j such that $f(x_i, t) \neq f(x_j, t)$, otherwise the race will last very long, and the output will be meaningless. At the end of the race, $3T$ evaluations have been performed, therefore T is called the halting time. Intuitively, the closer the points x_i are in terms of fitness value, the larger T will be. This is formalized below.

The reason why δ' is used in Algorithm 2 as the confidence parameter instead of δ will appear later on (the notation δ is needed elsewhere).

Algorithm 2 Bernstein race between 3 points. Eq. 1 is Bernstein's inequality to compute the precision for empirical estimates (see e.g. [11, p124]); $\hat{\sigma}_i$ is the empirical estimate of the standard deviation of point x_i's associated random variable $F_t(x_i)$ (it is 0 in the first iteration, which does not alter the algorithm's correctness); $\hat{f}(x)$ is the average of the fitness measurements at x. $\mathcal{B}(a)$ denotes a Bernoulli random law with parameter a.

$Bernstein(x_1, x_2, x_3, \delta')$
$T = 0$
repeat
  $T \leftarrow T + 1$
  Evaluate the fitness of points x_1, x_2, x_3 once, *i.e.* evaluate the noisy fitness at each of these points.
  Evaluate the precision:

$$\epsilon_{(T)} = 3 \log\left(\frac{3\pi^2 T^2}{6\delta'}\right)/T + \max_i \hat{\sigma}_i \sqrt{2 \log\left(\frac{3\pi^2 T^2}{6\delta'}\right)/T}. \tag{1}$$

until Two points (*good*, *bad*) satisfy $\hat{f}(bad) - \hat{f}(good) \geq 2\epsilon$
— **return** (*good*, *bad*)

Let us define $\Delta = \sup\{\mathbb{E}F_t(x_1), \mathbb{E}F_t(x_2), \mathbb{E}F_t(x_3)\} - \inf\{\mathbb{E}F_t(x_1), \mathbb{E}F_t(x_2), \mathbb{E}F_t(x_3)\}$. It is known [22] that if $\Delta > 0$ and if we consider a fixed number of arms[3],

- with probability $1 - \delta'$, the Bernstein race is consistent: $\mathbb{E}F_t(good) < \mathbb{E}F_t(bad)$;

[3]We here consider 3 arms only, but more general cases can be handled with a logarithmic dependency (see e.g. [22]).

- the Bernstein race halts almost surely, and with probability at least $1 - \delta'$, the halting time T verifies

$$T \leq K \log\left(\frac{1}{\delta'\Delta}\right)/\Delta^2, \tag{2}$$

where K is a universal constant;

- if, in addition,

$$\Delta \geq C \sup\{\mathbb{E}F_t(x_1), \mathbb{E}F_t(x_2), \mathbb{E}F_t(x_3)\}, \tag{3}$$

then the Bernstein race halts almost surely, and with probability at least $1 - \delta'$, the halting time T verifies

$$T \leq K' \log\left(\frac{1}{\delta'\Delta}\right)/\Delta, \tag{4}$$

where K' depends on C only.

The interested reader is referred to [22] and other references for more information.

3. LOWER BOUND

This section describes a general lower bound derived in [23], and concludes with the application of this bound to the STS model.

Let us consider a domain X, a function $f : X \times X \to \mathbb{R}$, and define

$$d(t_1, t_2) = \sup_{x \in X} |f(x, t_1) - f(x, t_2)|$$

for t_1 and t_2 in X. In all the paper, $B(n, p)$ is a binomial random variable (sum of n independent Bernoulli variables of parameter p).

THEOREM 1. *For any optimization algorithm Opt, let $N \in \mathbb{N}^*$ (a number of points visited), $\epsilon_0 > 0$, $0 < \epsilon < \epsilon_0$, $D \in \mathbb{N}^*$, $\delta \in]0, 1[$. We assume:*

- $H(\epsilon_0, D)$: $\forall \epsilon_1 < \epsilon_0 \; \exists (t_1, \ldots, t_D) \in X^D, \forall (i, j) \in [[1, D]]^2, i \neq j \Rightarrow d(t_i, t_j) = \epsilon_1$ (*generalized dimension*)

- $H_{PAC}(\epsilon, N, \delta)$: $\forall t, P(d(x_N^{t,\theta}, t) < \epsilon/2) \geq 1 - \delta$.

Then, if $\delta < 1/2D$,

$$P(B(N, \epsilon) \geq \lceil \log_2(D) \rceil) \geq 1 - D\delta. \tag{5}$$

The lower bound is related to a topological property of space X: a number D is taken such that for any distance $\epsilon < \epsilon_0$, D equidistant points of X can be found (assumption $H(\epsilon_0, D)$). This is closely related to the dimension of X: for instance, in $\mathbb{R}^d$, the maximum number of such equidistant points is $d + 1$.

The theorem states that if an optimization algorithm is able to find the optimum at precision ϵ with probability $1 - \delta$ in N fitness calls (i.e. the algorithm satisfies assumption $H_{PAC}(\epsilon, N, \delta)$), then N is necessarily large; the theorem explicitly gives a lower bound on N. Indeed, Eq. 5 implies

a clearer expression of the lower bound (using Chebyshev inequality):

$$N = \Omega(\log_2(D)/\epsilon) \qquad (6)$$

for fixed D, where N is the number of iterations required to reach precision ϵ with confidence $1 - \delta$ for $\delta < 1/2D$. The theorem holds for any monotonic transformation of the sphere function. However, the distance d is not the same for different classes of fitnesses. As mentioned earlier, we are interested in the Scaled-Translated sphere model $((x, t) \mapsto c + \lambda \|x - t\|^p$ with optimum t).

COROLLARY 2. *Under the conditions of Theorem 1, for any optimization algorithm learning a fitness of the STS model, if ϵ_N is the quantile $1 - \delta$ of the Euclidean distance to the optimum after N fitness calls and if $p \geq 1$, then $\epsilon_N = \Omega(log(D)/N)$.*

As stated in [23], the lower bound for $p = 1$ is straightforward, since in this case it is clear that $d(t_1, t_2) = \|t_1 - t_2\|$. Moreover, in the general STS model, we can show that for any $p \geq 2$, $d(t_1, t_2) = \Theta(\|t_2 - t_1\|)$, which validates the above corollary. The lower bound of the corollary is tight for $p = 1$ (see [23]). We will see that it is also tight if $p = 2$ for $c = 0$—in this case, both QLR and R-EDA reach this dependency.

4. UPPER BOUNDS

Upper bounds on the convergence rate for the STS model will now be presented, using R-EDA (Algorithm 3 along with a Bernstein race). In the model restricted to $p = 1$, upper bounds for small noise (i.e. $c = 0$) have been derived in [23], and upper bounds for large noise (i.e. $c > 0$) have been derived in [24]. In both cases, the bounds match the lower bound. This is why we focus on $p \geq 2$, which includes the case $p = 2$ that often appears in practice. In this section, the optimum will be referred to as x^*, and $f(x, x^*)$ will be noted $f(x)$ for short.

R-EDA is a (3,3) evolution strategy: the parent population consists of 3 points, and 3 points are generated from this population and act as the new population. The difference with respect to "standard" EDAs is as follows:

- the algorithm is derandomized: population t is generated deterministically from population $t - 1$;

- since $\mu = \lambda$, there is no need for actually ranking all the points (the algorithm still orders two points among the three as will be seen below).

The algorithm is comparison-based (since fitness values only matter by how they order the population), and fits the general description of an EDA.

Sketch of R-EDA (Algorithm 3). We will use R-EDA (Algorithm 3) for showing the upper bounds. It proceeds by iteratively splitting the domain in two (not necessarily equal) halves, and retaining the one that most probably contains the optimum. At iteration n, from the n_{th} domain $[x_n^-, x_n^+]$, the $(n + 1)_{th}$ domain $[x_{n+1}^-, x_{n+1}^+]$ is obtained by:

Algorithm 3 R-EDA: algorithm for optimizing noisy fitness functions. *Bernstein* denotes a Bernstein race, as defined in Algorithm 2. The initial domain is $[x_0^-, x_0^+] \in \mathbb{R}^D$, δ is the confidence parameter.

$n \leftarrow 0$
while True **do**
  $c_n = \arg\max_i (x_n^+)_i - (x_n^-)_i$ // Pick the coordinate with highest uncertainty
  $\delta_n^{\max} = (x_n^+)_{c_n} - (x_n^-)_{c_n}$
  for $i \in [[1, 3]]$ **do**
    $x'^i_n \leftarrow \frac{1}{2}(x_n^- + x_n^+)$ // Consider the middle point
    $(x'^i_n)_{c_n} \leftarrow (x_n^-)_{c_n} + \frac{i-1}{2}(x_n^+ - x_n^-)_{c_n}$ //The c_n^{th} coordinate may take $3 \neq$ values
  end for
  $(good_n, bad_n) = Bernstein(x'^1_n, x'^2_n, x'^3_n, \frac{6\delta}{\pi^2(n+1)^2})$.
  // A good and a bad point
  Let H_n be the halfspace
  $\{x \in \mathbb{R}^D; \|x - good_n\| \leq \|x - bad_n\|\}$
  Split the domain: $[x_{n+1}^-, x_{n+1}^+] = H_n \cap [x_n^-, x_n^+]$
  $n \leftarrow n + 1$
end while

- Finding the coordinate c_{n+1} such that $\delta_n^{max} = (x_n^+)_{c_n} - (x_n^-)_{c_n}$ is maximal;

- Selecting three regularly spaced points along this coordinate (see Figure 1);

- Repeatingly assessing those 3 points until we have confidence that the optimum is closer to one point x'^i_n than to another x'^j_n (by Bernstein race);

- Splitting the domain by the hyperplane in the middle of these points and normal to the line they define, and keeping only the side of the domain containing x'^i_n.

It is important to notice that three points selected at each iteration are necessarily distinct. A key element in proving upper bounds with this algorithm is that the fitness monotonic in the distance to the optimum ($\|a - x^*\| > \|b - x^*\| \Rightarrow f(a) > f(b)$), and it also has spherical symmetry ($\|a - x^*\| = \|b - x^*\| \Rightarrow f(a) = f(b)$). Consequently, it is guaranteed that when choosing three points as in Algorithm 3, at least one of them will have an expected fitness that is different from two others. That is why the race will output a consistent result with high probability.

For simplicity, it is assumed that the initial domain is a hyperrectangle. Consequently, at any iteration n, the halfspace H_n is a hyper-rectangle, whose largest axis' length δ_n^{max} (defined in Algorithm 3) satisfies $\delta_n^{max} \leq \frac{3}{4}^{\lfloor n/D \rfloor}$. The straightforward proof of this fact is given in [23], where R-EDA first appears.

The following lemma will be used for the upper bound. A similar lemma was published in [23], but it only applied to $p = 1$. Notations are those introduced in Algorithm 3.

LEMMA 3. (The conditions of the Bernstein race are met)

Assume that $x^* \in [x_n^-, x_n^+]$ and $p \geq 2$. Then

$$\max_{(i,j)\in[[1,3]]^2} f(x_n^{'j}) - f(x_n^{'i}) \geq 2\left(\frac{\delta_{max}^n}{2}\right)^p. \qquad (7)$$

Proof of Lemma 3. Let $\bar{x_n^*}$ be the projection of x^* on the line on which $x_n^{'1}, x_n^{'2}, x_n^{'3}$ lie. The result will now be proved for $(\bar{x_n^*})_{c_n} \in [(x_n^{'1})_{c_n}, (x_n^{'2})_{c_n}]$. The proof for the case $(\bar{x_n^*})_{c_n} \in [(x_n^{'2})_{c_n}, (x_n^{'3})_{c_n}]$ is symmetric (see Figure 1).
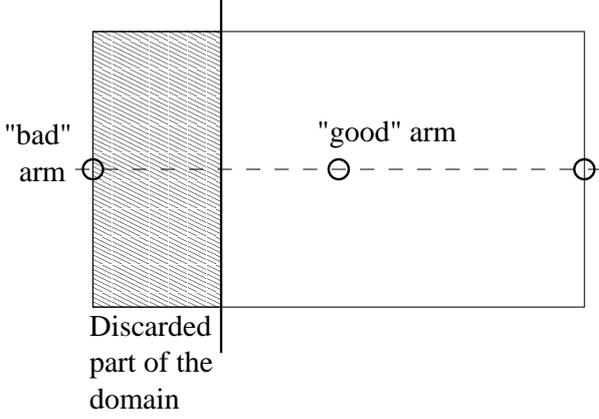

Figure 1: The large rectangle is the domain $[x_n^-, x_n^+]$. The three circles are arms $x_n'^1, x_n'^2, x_n'^3$; the left arm is the "bad" arm, whereas the arm in the center is the "good" arm, *i.e.* the one which proved to be closer to the optimum than the left arm, with confidence $1 - 6\delta/(\pi^2 n^2)$.

First of all, we have

$$\Delta_n \doteq \max_{i,j\in[[1,3]]^2} f(x_n^{'i}) - f(x_n^{'j}) \geq f(x_n^{'3}) - f(x_n^{'2}).$$

By Pythagora's theorem, $\forall i \in [[1,3]], ||x'^i - x^*||^2 = ||x_n'^i - \bar{x_n^*}||^2 + ||\bar{x_n^*} - x^*||^2$. Thus,

$$\Delta_n \geq \left(\sqrt{||x_n'^3 - \bar{x_n^*}||^2 + ||\bar{x_n^*} - x^*||^2}\right)^p$$
$$- \left(\sqrt{||x_n'^2 - \bar{x_n^*}||^2 + ||\bar{x_n^*} - x^*||^2}\right)^p.$$

Note that $||x'^3 - \bar{x_n^*}|| = ||x_n'^2 - \bar{x_n^*}|| + \delta_n^{\max}/2$. Define $d = ||\bar{x_n^*} - x^*||^2$ and $a = ||x'^3 - \bar{x_n^*}||$. Then, observing that $\delta_n^{\max} \geq a \geq \delta_n^{\max}/2$, we have

$$\Delta_n \geq \left(\sqrt{a^2 + d}\right)^p - \left(\sqrt{(a - \delta_n^{\max}/2)^2 + d}\right)^p$$
$$\geq a^p \left(\left(\sqrt{1 + d/a^2}\right)^p - \left(\sqrt{\left(1 - \frac{\delta_n^{\max}}{2a}\right)^2 + \frac{d}{a^2}}\right)^p\right)$$
$$\geq \left(\frac{\delta_n^{\max}}{2}\right)^p \left((\sqrt{1 + d/a^2})^p - (\sqrt{\frac{1}{4} + \frac{d}{a^2}})^p\right). \qquad (8)$$

By setting $u = d/a^2$, it is clear that Δ_n is greater than the minimum of $u \mapsto (\sqrt{1+u})^p - (\sqrt{1/4+u})^p$ on the interval

$[0, D]$ (since $\sqrt{d} = ||\bar{x_n^*} - x^*|| \leq \sqrt{D}\delta_n^{\max}/2$). This function is non-decreasing for $p \geq 2$, and therefore its minimum is its value in 0, which is, for all $p \geq 2$, at least $\frac{1}{2}$; injecting in Equation 8 yields $\Delta_n \geq 2\left(\frac{\delta_n^{\max}}{2}\right)^p$, as stated by Eq. 7. $\square$

THEOREM 4. (Upper bounds for the STS model) *Consider the STS model, and a fixed dimension D. The number of evaluations requested by R-EDA (Algorithm 3) to reach precision ϵ with probability at least $1 - \delta$ is $\tilde{O}(\frac{\log(1/\delta)}{\epsilon^{2p}})$.*

Proof of Theorem 4. First, note that at iteration n, ϵ is upper bounded by $||x_n^- - x_n^+||$. Eq. 7 (shown in Lemma 3) ensures that $\Delta_n = \Omega(||x_n^+ - x_n^-||^p)$ (Δ_n is defined by Eq. 3). Therefore, applying the concentration inequality, presented as Eq. 2, the number of evaluations in the nth iteration is at most

$$\tilde{O}\left(\log\left(\frac{6\delta}{\pi^2(n+1)^2}\right)/||x_n^- - x_n^+||^{2p}\right). \qquad (9)$$

Now, let us consider the number $N(\epsilon)$ of iterations before a precision ϵ is reached. Eq. 4 shows that there is a constant $k < 1$ such that $\epsilon \leq ||x_n^+ - x_n^-|| \leq Ck^{N(\epsilon)}$. Injecting this in Eq. 9 shows that the cost (the number of evaluations) in the last call to the Bernstein race is

$$Bound_{last}(\epsilon) = \tilde{O}\left(-\log\left(\frac{6\delta}{\pi^2(N(\epsilon)+1)^2}\right)/\epsilon^{2p}\right). \quad (10)$$

Since $N(\epsilon) = O(\log(1/\epsilon))$, $Bound_{last} = O(\log(\log(1/\epsilon)/\delta))/\epsilon^{2p}$. For a fixed dimension D, the cost of the $(N(\epsilon) - i)^{th}$ iteration is $O(\lceil Bound_{last}/(k')^i\rceil)$ because the algorithm ensures that after D iterations, $||x_n^+ - x_n^-||$ decreases by at least $3/4$ (see Eq. 4). The sum of the costs for $N(\epsilon)$ iterations is the sum of $O(Bound_{last}/(k')^i)$ for $i \in [[0, N(\epsilon) - 1]]$, that is $O(Bound_{last}/(1 - k')) = O(Bound_{last})$ (plus $O(N(\epsilon))$) for the rounding associated to the $\lceil ... \rceil$). The overall cost is therefore $O(Bound_{last} + \log(1/\epsilon))$, yielding the expected result. $\square$

Theorem 4 can be modified to use the small noise assumption, i.e. the case $c = 0$. We then get a Bernstein's type rate, as follows:

THEOREM 5. (Upper bounds for the STS model with small noise) *Consider the STS model, and a fixed dimension D. Assume additionally that $c = 0$, i.e. the scaled sphere model. The number of evaluations requested by R-EDA (Algorithm 3) to reach precision ϵ with probability at least $1 - \delta$ is $\tilde{O}(\frac{\log(1/\delta)}{\epsilon^p})$.*

Proof of Theorem 5. The variance of a Bernoulli random variable is always upper bounded by its expectation. The case $c = 0$ implies that the expectation is upper bounded by the square of the distance to the optimum. Therefore, Eq. 3 holds. Thanks to Eq. 3, we can then use Eq. 4 instead of Eq. 2 in the proof of Theorem 4. This yields the expected result. $\square$

Note that this analysis is not limited to fitnesses that are exactly described by $f(x) = c + ||x - x^*||^p$, but apply to any

monotonic transformation of the sphere function that has a Taylor expansion of degree p around its optimum.

5. EXPERIMENTS
In this section, we illustrate results of our experiments with an algorithm without surrogate models, UH-CMA, introduced in [17], and an algorithm with surrogate models, QLR (based on Quadratic Logistic Regression, i.e. it is assumed that the function is locally quadratic).

UH-CMA is an uncertainty handling approach based on a state-of-the-art CMA-ES.

QLR, in comparison to many alternative methods, has only one mega-parameter to adjust (a Bayesian prior) and keeps information on all observed data.

5.1 Experimental results for UH-CMA— optimization without surrogate models
UH-CMA has been developed with intensive testing on the BBOB challenge [4], which includes mild models of noise. See [16] for the source code used in these experiments. The optimization domain is $\mathbb{R}^2$. Let $\mathcal{B}(q)$ denote a Bernoulli distribution of parameter q, $\mathcal{N}(\mu, \sigma^2)$ denote a Gaussian distribution centered on μ with variance σ^2, and $\mathcal{U}(I)$ denote a uniform distribution on interval I. UH-CMA[4] was tested on four different noisy fitnesses: 1) $\|x\|^2(1 + \mathcal{N}(0, 0.1))$; 2) $\|x\|^2 + \mathcal{U}([0,1])$; 3) $\mathcal{B}(\|x\|^2)$; 4) $\mathcal{B}(\|x\|^2 + 0.5)$.

The experiements with UH-CMA have been carried out using the following setting. The number of repeats equals 100, the population size $\lambda = 4 + \lfloor 3 \log N \rfloor$, where N is the problem dimension, and the parent number $\mu = \lfloor \lambda/2 \rfloor$.

The initial values required by UH-CMA to start the search were sampled from $\mathcal{U}([0,1]^2)$. The convergence (and divergence) of UH-CMA—illustrated on Figure 2—is known to be log-linear.

For $\|x\|^2(1 + \mathcal{N}(0, 0.1))$, the algorithm converges efficiently: the precision decreases exponentially as the number of iterations increases. For $\|x\|^2 + \mathcal{U}([0,1])$, the precision stops improving after a few hundred iterations. For $\mathcal{B}(\|x\|^2)$ and $\mathcal{B}(\|x\|^2 + 0.5)$ we observed divergence.

Let us point out that by adding some specific rules for averaging multiple fitness evaluations depending on the step-size, specifically for each fitness function, it is possible to obtain much better rates [15]. However, the rates remain worse than those reached by QLR, as shown in the following section.

5.2 Experiments with QLR—optimization with surrogate models
QLR is based on a Bayesian quadratic logistic regression. It samples regions of the search space with maximum variance of the posterior probability, i.e. regions with high variance

---

[4]The version of UH-CMA used in our experiments is the one available in http://www.lri.fr/~hansen/cmaesintro.html. The noise handling was activited and it was not modified in any manner.

conditionally to past observations. This is a key difference w.r.t. algorithms without surrogate models, which tend to sample points close to the optimum. QLR is fully described by [12, 8, 21] (design of experiments for quadratic logistic model), [25] (active learning for logistic regression). See [10] for the code we used here, specifically tailored to binary noisy fitnesses.

QLR was tested on fitnesses of the form $B(\|x\|^p + c)$, for p in $\{1, 2\}$ and c in $\{0, 1/2\}$. The search space is $\mathbb{R}^2$. Figure 3 shows the experimental results:

Top left (p=1, c=0): QLR converges on $x \mapsto B(\|x - x^*\|)$, but with a suboptimal exponent $\frac{1}{2}$ (the slope of the curve is $-\frac{1}{2}$ in log-scale), i.e. $\mathbb{E}f(x_n) - \mathbb{E}f(x^*) \simeq \Theta(1/\sqrt{n})$. R-EDA reaches a better $1/n$ in this case;

Top right (p=1, c=1/2): QLR converges with optimal exponent $1/\sqrt{n}$ also reached by R-EDA;

Bottom left (p=2, c=0): QLR reaches $\mathbb{E}f(x_n) - \mathbb{E}f(x^*) \simeq \Theta(1/n)$ as well as R-EDA;

Bottom right (p=2, c=1/2): QLR still reaches $\mathbb{E}f(x_n) - \mathbb{E}f(x^*) \simeq \Theta(1/n)$ whereas R-EDA only reaches $1/\sqrt{n}$.

6. CONCLUSION
The convergence rates for R-EDA (see [24]) and QLR are as follows:

f	$\|x_n - x^*\|$ for R-EDA	Known lower-bound	$\|x_n - x^*\|$ for QLR ($p = 2$)
$\lambda\|x - x^*\|$	$O(1/n)$	$\Omega(1/n)$	$\simeq 1/\sqrt{n}$
$\lambda\|x - x^*\| + c$	$\tilde{O}(1/\sqrt{n})$	$\Omega(1/n)$	$\simeq 1/\sqrt{n}$
$g(\|x - x^*\|)$	$o(1)$	$-$	$-$
$\lambda\|x - x^*\|^p + c$	$\tilde{O}(1/n^{1/2p})$	$\Omega(1/n)$	$\simeq 1/\sqrt{n}$
$\lambda\|x - x^*\|^p$	$\tilde{O}(1/n^{1/p})$	$\Omega(1/n)$	$\simeq 1/\sqrt{n}$
$\lambda\|x - x^*\|^2$	$\tilde{O}(1/\sqrt{n})$	$\Omega(1/n)$	$\simeq 1/\sqrt{n}$

Convergence rates are given for minimization; the fitness at point x is the Bernoulli random variable $\mathcal{B}(f(x))$ with parameter $\min(1, \max(0, f(x)))$, x_n is the approximation of the optimum after n fitness evaluations, x^* is the optimum, $c > 0$, and g is some increasing mapping.

For the rightmost column, it is important to point out that we tested QLR *without* knowledge of the parameter p, so that the comparison with other algorithms is fair. In particular, there is a single algorithm, R-EDA, which provably realizes the upper bounds above; a better algorithm should be better for all cases simultaneously without problem-specific parametrization.

The original results of our paper are presented by three last rows and the rightmost column; in particular we have shown:

- The upper and lower bounds for an exponent $p > 1$;

- For $p = 1$ and $c = 0$, QLR is not optimal; R-EDA reaches (provably) $\tilde{O}(1/n)$ whereas QLR has convergence $1/\sqrt{n}$. By construction, it is probably difficult for QLR to do better than $1/\sqrt{n}$;
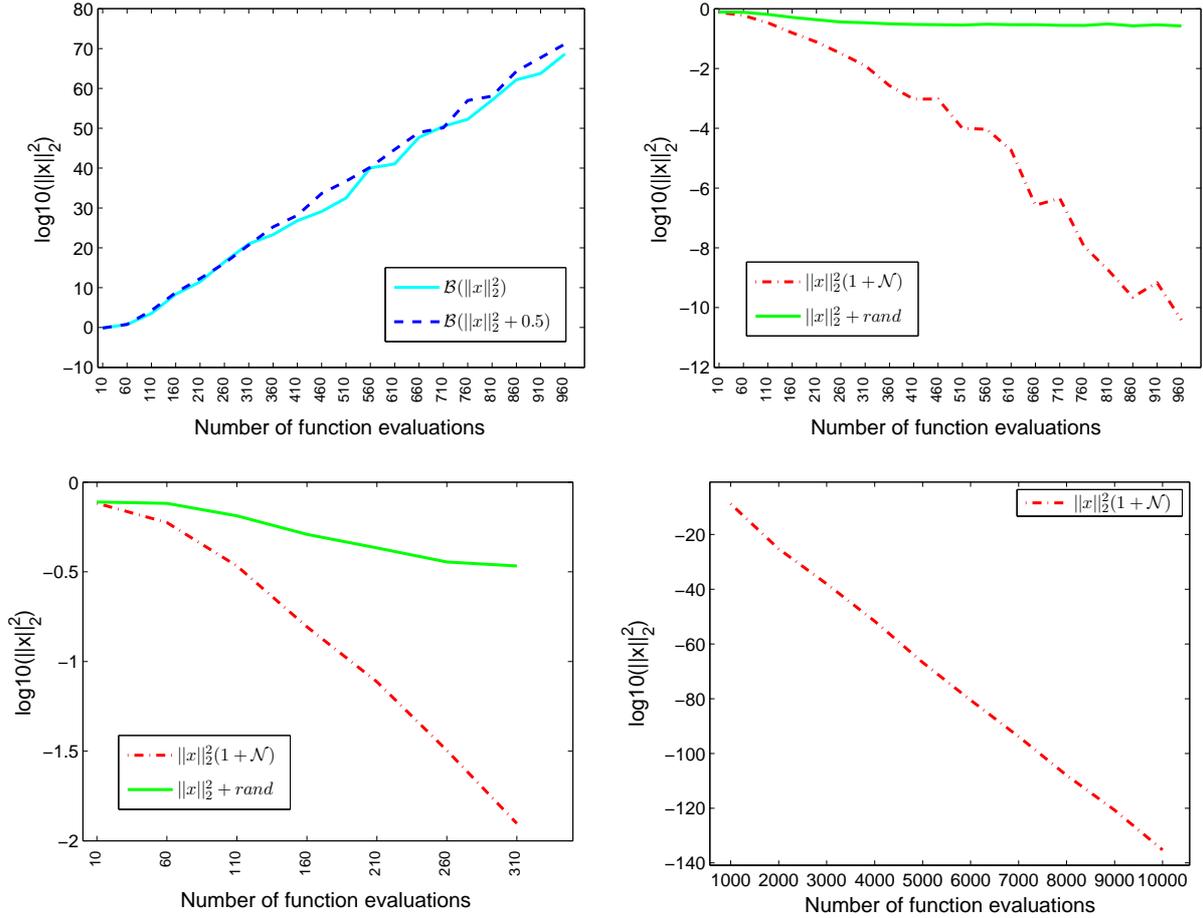
Figure 2: **Optimization with UH-CMA,** $\log 10$ **is the logarithmic function to the base 10. We see that (i) cases with variance not decreasing to zero are not handled properly; (ii) Bernoulli noise (even with variance going to zero) is not handled properly.**

- For $p = 1$ and $c > 0$, QLR and R-EDA perform equivalently $(1/\sqrt{n})$; the lower bound does not match the upper bound. For R-EDA we have a mathematical proof and for QLR empirical evidence.

- For $p = 2$ and $c = 0$, QLR and R-EDA perform equivalently $(1/\sqrt{n})$; the lower bound does not match the upper bound. For R-EDA we have a mathematical proof and for QLR empirical evidence.

- For $p = 2$ and $c > 0$, QLR (empirically) performs better than the proved upper bound and worse than the proved lower bound.

There is therefore still room for improvements.

Results for QLR and for UH-CMA are empirical, based on current versions of the algorithms. The available implementations of UH-CMA cope quite well with small noise situations, but as soon as the variance does not go to zero sufficiently fast they do not succeed.

R-EDA is efficient in many cases, yet its theoretical convergence rates are suboptimal in the case $B(c+||x-x^*||^2)$, more

relevant from a practical point of view. However, R-EDA is not limited to Bernoulli-like fitness functions, whereas QLR is. This is why QLR is more efficient in the case $B(c + ||x - x^*||^2)$ for $c > 0$. UH-CMA does not converge in such cases, what demonstrates that algorithms tailored for small noise models do not easily extend to models with large noise. However, UH-CMA is the only algorithm with log-linear precision as a function of the number of iterations in the easy case $||x - x^*||^2(1 + \mathcal{N})$.

R-EDA can be applied to any fitness of the form $x \mapsto g(||x - x^*||)$ with x^* the optimum and g an increasing mapping, and will converge to the optimum. If, in addition, if g is differentiable in 0 with non-null derivative, then the convergence rate is guaranteed to meet the rates $p = 1$ presented above. More generally, if g is p times differentiable in 0, with the $p - 1$ first derivatives null, then the convergence rate is the general rate presented above for a given p. A relevant further work would be to extend the algorithm to non-spherical models (i.e. no spherical symetry around the optimum), in order to have more general convergence bounds.
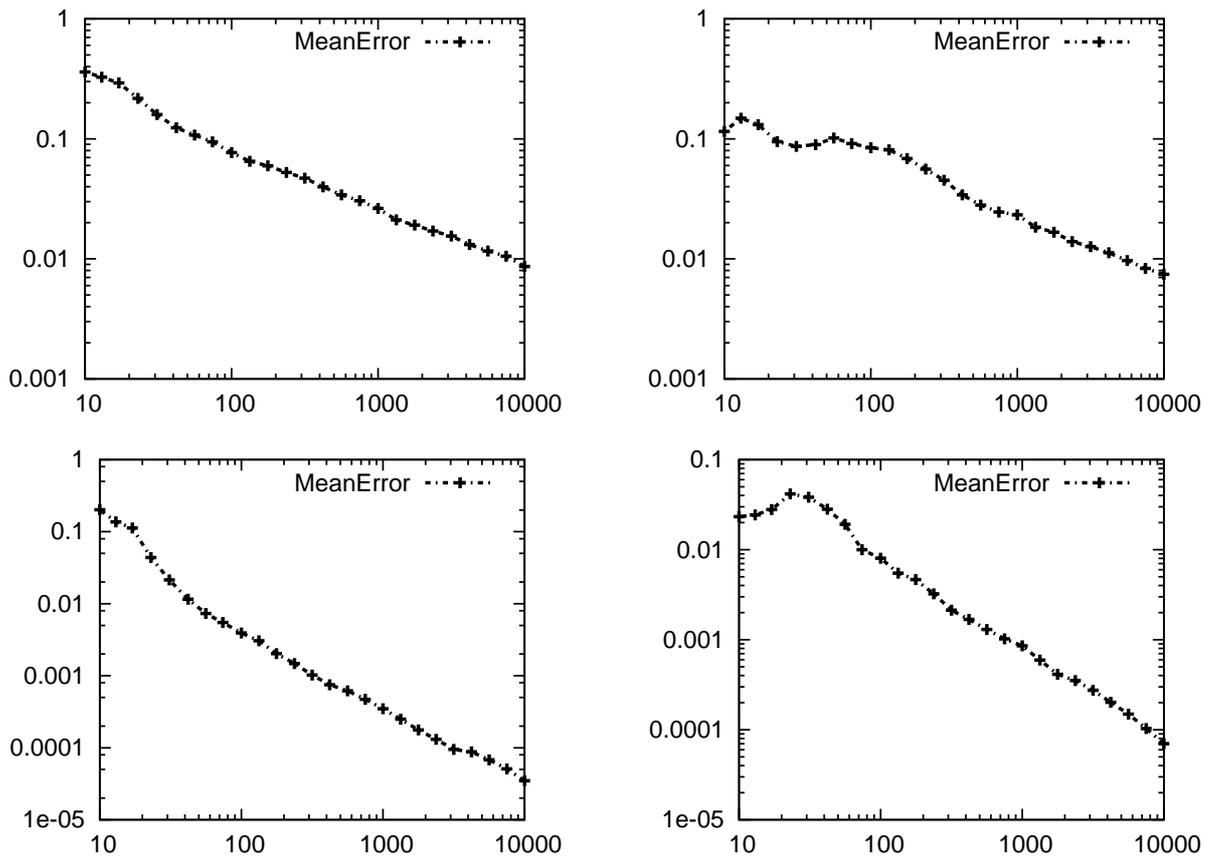
Figure 3: Convergence rate of QLR in various cases. On the X-axis: the number of evaluations; on the Y-axis: $\mathbb{E}f(x_n) - \mathbb{E}f(x^*)$. Both are in log-scale to emphasize the exponent. The noisy fitnesses tested are of the form $B(||x||^p + c)$ (top: $p = 1$, bottom: $p = 2$, left: $c = 0$, right: $c = 1/2$).

Given the convergence rate table above, one can see that lower bounds for $p > 1$ or $c > 0$ are not tight. A relevant further work would be either to find out how to reach these bounds, or to prove lower bounds achieving tightness—which seems more likely, given that the current lower bounds are quite optimistic.

7. ACKNOWLEDGEMENTS

8. REFERENCES

[1] D. V. Arnold and H.-G. Beyer. Efficiency and mutation strength adaptation of the (mu/muI,lambda)-es in a noisy environment. In *Parallel Problem Solving from Nature*, volume 1917 of *LNCS*, pages 39–48. Springer, 2000.

[2] D. V. Arnold and H.-G. Beyer. A general noise model and its effects on evolution strategy performance. *IEEE Transactions on Evolutionary Computation*, 10(4):380–391, 2006.

[3] D. V. Arnold and H.-G. Beyer. Evolution strategies with cumulative step length adaptation on the noisy parabolic ridge. *Natural Computing: an international journal*, 7(4):555–587, 2008.

[4] A. Auger, S. Finck, N. Hansen, and R. Ros. BBOB 2009: Comparison Tables of All Algorithms on All Noisy Functions. Technical Report RT-0384, INRIA, 04 2010.

[5] S. Bernstein. On a modification of chebyshev's inequality and of the error formula of laplace. *Original publication: Ann. Sci. Inst. Sav. Ukraine, Sect. Math. 1*, 3(1):38–49, 1924.

[6] S. Bernstein. *The Theory of Probabilities*. Gostehizdat Publishing House, Moscow, 1946.

[7] H.-G. Beyer. *The Theory of Evolutions Strategies*. Springer, Heidelberg, 2001.

[8] K. Chaloner. Bayesian design for estimating the turning point of a quadratic regression. *Communications in Statistics—Theory and Methods*, 18(4):1385–1400, 1989.

[9] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations.

Annals of Math. Stat., 23:493–509, 1952.

[10] R. Coulom. Source code for qlr, Mar. 2010. http://remi.coulom.free.fr/QLR/.

[11] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic Theory of Pattern Recognition*. Springer, 1997.

[12] E. Fackle Fornius. *Optimal Design of Experiments for the Quadratic Logistic Model*. PhD thesis, Department of Statistics, Stockholm University, 2008.

[13] J. M. Fitzpatrick and J. J. Grefenstette. Genetic algorithms in noisy environments. *Machine Learning*, 3:101–120, 1988.

[14] U. Hammel and T. Bäck. Evolution strategies on noisy functions: How to improve convergence properties. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving From Nature*, volume 866 of *LNCS*, pages 159–168, Jerusalem, 1994. Springer.

[15] N. Hansen. Personal communication.

[16] N. Hansen. Source code for uh-cma, June 2008. Version 3, http://www.lri.fr/ hansen/cmaesintro.html.

[17] N. Hansen, A. Niederberger, L. Guzzella, and P. Koumoutsakos. A Method for Handling Uncertainty in Evolutionary Optimization with an Application to Feedback Control of Combustion. *IEEE Transactions on Evolutionary Computation*, 2009.

[18] V. Heidrich-Meisner and C. Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 401–408, New York, NY, USA, 2009. ACM.

[19] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[20] M. Jebalia and A. Auger. On multiplicative noise models for stochastic search. In *Parallel Problem Solving From Nature*, Dortmund, Germany, 2008.

[21] A. I. Khuri, B. Mukherjee, B. K. Sinha, and M. Ghosh. Design issues for generalized linear models: A review. *Statistical Science*, 21(3):376–399, 2006.

[22] V. Mnih, C. Szepesvári, and J.-Y. Audibert. Empirical Bernstein stopping. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 672–679, New York, NY, USA, 2008. ACM.

[23] P. Rolet and O. Teytaud. Bandit-based estimation of distribution algorithms for noisy optimization: Rigorous runtime analysis. In *Proceedings of Lion4 (accepted); presented in TRSH 2009 in Birmingham*, 2009.

[24] P. Rolet and O. Teytaud. Adaptive Noisy Optimization. In *EvoStar 2010*, Istambul Turquie, 02 2010.

[25] A. I. Schein and L. H. Hungar. Active learning for logistic regression: An evaluation. *Machine Learning*, 68(3):235–265, 2007.

[26] O. Teytaud and A. Auger. On the adaptation of the noise level for stochastic optimization. In *IEEE Congress on Evolutionary Computation*, Singapour, 2007.