Lower Bounds for Comparison Based Evolution Strategies using VC-dimension and Sign Patterns*

Hervé Fournier† Olivier Teytaud‡

January 4, 2010

Abstract

We derive lower bounds on the convergence rate of comparison based or selection based algorithms, improving existing results in the continuous setting, and extending them to non-trivial results in the discrete case. This is achieved by considering the VC-dimension of the level sets of the fitness functions; results are then obtained through the use of the shatter function lemma. In the special case of optimization of the sphere function, improved lower bounds are obtained by an argument based on the number of sign patterns.

Keywords: Evolution Algorithms; Convergence Speed; VC-dimension; Sign Patterns; Sphere Function.

1 Introduction

Evolution strategies (ES), defined by Rechenberg [21], are a family of optimization algorithms with nice robustness properties [11]. They are often termed "order 0" methods as, in the continuous domain, they do not use gradients or Hessians. Interestingly, most ES are in fact a special case of order 0 methods: in addition to not using gradients, they only use comparisons between fitness values and not the fitness values themselves. Using comparisons only makes an important difference since it is known that, even without gradient, a super-linear convergence can be obtained when using fitness values – see *e.g.* [3], using surrogate models for a super-linear convergence rate.

Comparisons allow the selection of a subset of the points, but they can also be used for ranking either these selected individuals or the whole population. Examples of algorithms using more information than just the selection of a subset are some roulette-wheel algorithms (stochastic sampling [5] and rank-based fitness assignment [4, 29]), evolution strategies using weighted recombination [13, 1] or BREDA [11].

Hence, the wide family of order 0 methods can be divided into (i) algorithms using fitness values; (ii) algorithms using the full ranking of all the population; (iii) algorithms using the full ranking of selected points; (iv) algorithms using only the set of selected points. In cases

*A preliminary version of this paper appeared in PPSN 2008 [27].

†Laboratoire PRiSM, CNRS UMR 8144 and Univ. Versailles Saint-Quentin-en-Yvelines, 45 av. des États-Unis, 78035 Versailles, France. Email: herve.fournier@prism.uvsq.fr

‡TAO (Inria), LRI, UMR 8623 (CNRS - Univ. Paris-Sud), Bât 490, Univ. Paris-Sud 91405 Orsay, France. Email: olivier.teytaud@inria.fr

(ii) to (iv), the branching factor of the algorithm, i.e., the number of possible outcomes for the information extracted from the population in one iteration, is finite and bounded.

This fact has been used in [26] in order to provide lower bounds that match some upper bounds known for evolutionary algorithms [9, 2, 23]. The optimality of this comparison-based principle for some robustness criterion was shown in [11] – see also [4, 29, 5]. The tools provided in [26] for proving lower bounds for evolutionary algorithms are interesting, but, as pointed out by the authors, the bounds for the (μ, λ)-ES are far too small for $\mu > 1$ and λ larger than the dimension, while the discrete case provides essentially trivial results. In this work, we present improved lower bounds on the convergence rate of evolution strategies of type $(\mu \dot{+} \lambda)$-ES in terms of the VC-dimension of level sets of the fitness functions. In the special case of optimization of the sphere function, improved lower bounds on the convergence rate of evolution strategies are presented; they are obtained by bounding the number of sign patterns realized by a system of equations.

We now give some elements of comparison with existing results. As explained above, the present work builds on [26], where the branching factor was introduced for the analysis of evolutionary algorithms; results obtained here are improved in the case of families of fitness functions with bounded VC-dimension. The first lower bounds for some evolutionary algorithms in continuous domains were provided in [17, 15, 16]. The present paper extends these results by considering a wider family of evolutionary algorithms – our analysis encompasses all (μ, λ)-ES and $(\mu + \lambda)$-ES. We also generalize existing results in the discrete case by considering arbitrary values of parameters λ and μ, and improve previous results from [10] in the special cases of $(1 + \lambda)$-ES or $(\mu + 1)$-ES – a detailed comparison with state of the art is provided within the paper regarding these cases. We also remark that [25, Theorem 2], which is a complexity lower bound for a variant of particle swarm optimization, is included in the main theorem of [26].

The paper is organized as follows. Basic definitions and terminology of evolution strategies we consider are described in Section 2. Lower bounds on $(\mu \dot{+} \lambda)$-ES based on the branching factor, obtained in [26], are recalled in Section 3. Improved lower bounds on $(\mu \dot{+} \lambda)$-ES in terms of the VC-dimension are presented in Section 4. The special case of the sphere function is studied in Section 5. In the case of full ranking algorithms, we show that an argument based on the number of sign patterns can yield better bounds than the one obtained by VC-dimension arguments. We also present a simple algorithm based on full ranking, which permits an almost linear speed-up when the size of the offspring is linear in the dimension. Final remarks are presented in Section 6.

Notations. In all the paper, $\log(x)$ denotes the logarithm with basis 2, i.e. $\log(2) = 1$. The set of integers $\{1, 2, \ldots, n\}$ is denoted by $[[1, n]]$. The notation $|\cdot|$ is used to denote both the cardinal of a set and the length of a vector (i.e., $|(x_1, \ldots, x_n)| = n$). At last, $\|x\|_2$ denotes the Euclidean norm of the vector x: that is, $\|(x_1, \ldots, x_n)\|_2 = (x_1^2 + \ldots + x_n^2)^{1/2}$.

2 Evolution Strategies of type $(\mu \dot{+} \lambda)$

This section is devoted to a formal definition of algorithms of type $(\mu \dot{+} \lambda)$ – evolution strategies of type $(\mu \dot{+} \lambda)$, or $(\mu \dot{+} \lambda)$-ES – considered in this paper. We refer the reader to Beyer and Schwefel [7] for a comprehensive introduction to evolution strategies.

The aim of $(\mu \dot{+} \lambda)$-ES is to find the minimum of a real function f, called the fitness function, defined over a domain D. These algorithms work with comparisons only: given

two points $x, y \in D$, they use only the sign of $f(x) - f(y)$. More precisely, given some points $z_1, \ldots, z_p \in D$, these algorithms are given (possibly partial) information on the signs of $f(z_i) - f(z_j)$, for all $1 \leqslant i < j \leqslant p$. The exact information on this sign vector these algorithms have access to depends on the specific type of algorithms considered. Of course these algorithms are not required to work for a single fitness function, but for a whole family of fitness functions. In the following, we denote by $\mathcal{F}$ this set of fitness functions.

In the rest of the paper, unless otherwise explicitly stated, we assume equality of fitness values $f(x) = f(y)$, for two points x and y generated at any epoch, never occurs. This is a reasonable hypothesis in the continuous setting (*e.g.*, when $D = [0,1]^d$) where this assumption almost surely holds for many combinations of algorithms and fitness functions. We present a way of handling the general case in Section 4.6.

Let λ and μ be two integers. A Selection Based (SB-$(\mu \,\dot{+}\, \lambda)$-ES) is a randomized algorithm working as follows. Its outcome is a sequence of approximations of the optimum, as proposed by the *proposal* function. There is a set $\mathcal{I}$ of internal states. The algorithm starts in the initial state $I_0 \in \mathcal{I}$ returned by the function *initial_state*. At each iteration, the algorithm consists of the following three successive steps. First generate a set of λ points, called the *offspring*. Then select only the μ best ones, i.e. the μ points with lowest fitness values; in the case of an SB-(μ, λ)-ES, the points generated at previous stages are forgotten and this selection is performed only among the offspring, while an algorithm of type SB-$(\mu + \lambda)$-ES selects the μ best points among the offspring *and* the points selected at the previous step (hence these μ selected points are always the μ points with lowest fitness values found so far). Finally, the internal state I_n is updated. (Notice that $\mu \leqslant \lambda$ must hold in the case of an SB-(μ, λ)-ES.)

Algorithm 1 Selection Based (μ, λ)-ES (resp. Selection Based $(\mu + \lambda)$-ES). Framework for evolution strategies based on selection, working on a fitness function f. The random variable ω is a random seed. An algorithm matching this framework is obtained by specifying the distribution of ω, the space of states, and the functions *initial_state*, *generate*, *update* and *proposal*.

Initialization: $I_0 \leftarrow \text{initial\_state}(\omega)$; $S_0 \leftarrow \emptyset$; $n \leftarrow 0$
while true **do**
  $n \leftarrow n + 1$
  Generation step : $O_n \leftarrow \text{generate}(I_{n-1})$ (*i.e.* generate an offspring O_n of λ distinct points)
  $E_n \leftarrow O_n$ (resp. $E_n \leftarrow O_n \oplus S_{n-1}$)
  $\ell \leftarrow \min(\mu, |E_n|)$
  Selection step: $v_n \leftarrow \text{select}(E_n, f)$
    The vector $v_n = (i_1, \ldots, i_\ell)$ is defined by:

$$\begin{cases} 1 \leqslant i_1 < i_2 < \cdots < i_\ell \leqslant |E_n| \\ \text{for all } j \text{ and } k, \text{ if } j \in v_n \text{ and } k \notin v_n, \text{ then } f(E_{n,j}) < f(E_{n,k}) \end{cases}$$

  Update the internal state: $I_n \leftarrow \text{update}(I_{n-1}, E_n, v_n)$
  $S_n \leftarrow (E_{n,i_1}, \ldots, E_{n,i_\ell})$
  $x^{(f)}_{\omega,n} \leftarrow \text{proposal}(I_n)$
end while

General outline of SB-(μ, λ)-algorithms (resp. SB-$(\mu + \lambda)$-algorithms) is summarized in Algorithm 1. In this algorithm (and Algorithm 2), the concatenation of the two vectors $x = (x_1, \ldots, x_k)$ and $x' = (x'_1, \ldots, x'_\ell)$ is denoted by $x \oplus x' = (x_1, \ldots, x_k, x'_1, \ldots, x'_\ell)$; we also use the shortcut $v \in (x_1, \ldots, x_k)$ to express that there exists $i \in [[1, k]]$ such that $x_i = v$.

Let us detail how the selection step is performed. If $E_n = (z_1, \ldots, z_p)$ is the vector of points considered at step n (either $E_n = O_n$ in the case of SB-(μ, λ)-ES or $E_n = O_n \oplus S_{n-1}$ in the case of SB-$(\mu + \lambda)$-ES), the function *select* returns a vector of μ distinct integers $v_n = (i_1, \ldots, i_\mu)$ such that:

$$\begin{cases} i_1 < \ldots < i_\mu \\ \text{for all } j \in \{i_1, \ldots, i_\mu\} \text{ and } k \in [[1, p]] \setminus \{i_1, \ldots, i_\mu\},\ f(z_j) < f(z_k) \end{cases}$$

Notice that the length of the vector v_n is equal to μ except maybe during the first few iterations of the algorithm in the case $\lambda < \mu$: This explains the use of the auxiliary variable ℓ in Algorithm 1.

Algorithms with the "+" are usually termed *elitist*; this means that we always keep the best individuals. Algorithms with the "," are termed *non-elitist*.

Finally, we would like to explain a generalization of SB-$(\mu \overset{+}{,} \lambda)$-ES, called Full Ranking $(\mu \overset{+}{,} \lambda)$-ES. Instead of just giving the best μ points (i.e., the μ points with the lowest fitness values), we can consider a selection procedure which returns the best μ points *ordered with respect to their fitness*: the algorithm knows which selected point is the best, which point is the second best, and so on.

Algorithm 2 Full Ranking (μ, λ)-ES (resp. Full Ranking $(\mu + \lambda)$-ES). Framework for evolution strategies based on full ranking, working on a fitness function f. The random variable ω is a random seed. Compared to Algorithm 1, the vector of integers v_n obtained at the selection step is now ordered with respect to the fitness values of points from E_n; this framework is thus more general.

Initialization: $I_0 \leftarrow \text{initial\_state}(\omega)$; $S_0 \leftarrow \emptyset$; $n \leftarrow 0$
while true **do**
  $n \leftarrow n + 1$
  Generation step : $O_n \leftarrow \text{generate}(I_{n-1})$ (*i.e.* generate an offspring O_n of λ distinct points)
  $E_n \leftarrow O_n$ (resp. $E_n \leftarrow O_n \oplus S_{n-1}$)
  $\ell \leftarrow \min(\mu, |E_n|)$
  Selection step: $v_n \leftarrow \text{select}(E_n, f)$
    The vector $v_n = (i_1, \ldots, i_\ell)$ is defined by:

$$\begin{cases} i_1, \ldots, i_\ell \in [[1, |E_n|]] \\ f(E_{n,i_1}) < f(E_{n,i_2}) < \cdots < f(E_{n,i_\ell}) \\ \text{for all } j \notin v_n,\ f(E_{n,i_\ell}) < f(E_{n,j}) \end{cases}$$

  Update the internal state: $I_n \leftarrow \text{update}(I_{n-1}, E_n, v_n)$
  $S_n \leftarrow (E_{n,i_1}, \ldots, E_{n,i_\ell})$
  $x_{\omega,n}^{(f)} \leftarrow \text{proposal}(I_n)$
end while

The outline of these algorithms is summarized in Algorithm 2. More precisely, the selection step described in this algorithm works as follows. Given the vector of points $E_n = (z_1, \ldots, z_p)$ considered at step n, the function *select* returns a vector of μ distinct integers $v_n = (i_1, \ldots, i_\mu)$ such that:

$$\begin{cases} f(x_{i_1}) < \ldots < f(x_{i_\mu}) \\ \text{for all } j \in [[1, p]] \setminus \{i_1, \ldots, i_\mu\},\ f(z_{i_\mu}) < f(z_j) \end{cases}$$

(Once again, the length of the vector v_n may not be equal to μ at the beginning of the algorithm.)

Note that both Algorithms 1 and 2 define a class of algorithms: in order to obtain an algorithm, one has to specify the distribution of ω, how the offspring is generated (function *generate*), the space of states $\mathcal{I}$ as well as the functions *initial_state* and *update*, and finally the function *proposal*. Throughout the paper, we assume that all functions involved in these algorithms are measurable. A typical case is retrieved when the offspring is randomly and independently drawn according to a Gaussian distribution, with parameters (mean, variance and covariances) depending on the internal state of the algorithm.

Finally, let us remark that the whole source of randomization in the class of algorithms defined in this section is given by ω. Functions involved in these algorithms, such as the one generating the offspring, do not explicitly depend on the random seed ω in our presentation; this is because the whole source of randomization can be held in the random state I. Let us also notice that ω is not necessarily a real random variable: for example, ω can be chosen to be a countable sequence of independent random variables uniform in $[0, 1]$. Hence, randomized algorithms which draw a finite number of real numbers at each step of computation, as it happens in the Gaussian case discussed above, are easily seen to fall into the setting presented here.

3 Branching factor and convergence rate

We consider a (possibly discrete) domain $D \subset \mathbb{R}^d$ and a norm $\|\cdot\|$ on $\mathbb{R}^d$. For $\varepsilon > 0$, we define $N(\varepsilon)$ to be the maximum integer n such that there exist n distinct points $x_1, \ldots, x_n \in D$ with $\|x_i - x_j\| \geqslant 2\varepsilon$ for all $i \neq j$. In particular, $N(\varepsilon) = |D|$ when ε is small enough in the case of a finite domain D, and $\log N(\varepsilon) \sim d \log(1/\varepsilon)$ when $\varepsilon \to 0$ if the domain D is bounded with non-empty interior.

If each function $f \in \mathcal{F}$ has one and only one optimum $f^* = \arg\min_{x \in D} f(x)$, for any given optimization algorithm as in Algorithm 2, and for $\varepsilon > 0$ and $\delta > 0$, we define $n_{\varepsilon,\delta}$ be the minimum number n of iterations such that with probability at least $1 - \delta$, an optimum is found at the n-th iteration within distance ε; *i.e.*, $n_{\varepsilon,\delta}$ is minimal such that for all $n \geqslant n_{\varepsilon,\delta}$ and for all $f \in \mathcal{F}$,

$$\mathbb{P}_\omega[\|x^{(f)}_{\omega,n} - f^*\| < \varepsilon] \geqslant 1 - \delta$$

where $\mathbb{P}_\omega$ is the probability operator on ω and $x^{(f)}_{\omega,n}$ is the n-th point given by the proposal function in Algorithms 1 and 2.

In order to state lower bounds on the convergence rate of evolution strategies obtained in [26], we first need to introduce a couple of definitions. Consider an algorithm of type $(\mu \dot{+} \lambda)$-ES working over a set of fitness functions $\mathcal{F}$. Let us define $L_n(\omega)$, the number of different paths followed by the algorithm on the random seed ω after n steps of computation when the function f runs over $\mathcal{F}$. More precisely,

$$L_n(\omega) = |\{(I_0, I_1, \ldots, I_n) \ : \ f \in \mathcal{F}\}|\,,$$

where the states I_i in the sequence above implicitly depend on both the function f and the random seed ω.

The *branching factor* K of this algorithm is defined as

$$K = \sup_E |\{\text{select}(E, f) \ : \ f \in \mathcal{F}\}|,$$

where the supremum holds for:

- E any set of λ points from D in the case of SB-(μ, λ)-ES or Full Ranking (μ, λ)-ES;
- E any set of $\mu+\lambda$ points from D in the case of SB-$(\mu+\lambda)$-ES or Full Ranking $(\mu+\lambda)$-ES.

Of course the branching factor implicitly depends on the algorithm and on the set of fitness functions $\mathcal{F}$.

Notice that the number of different paths followed by some algorithm can be bounded in terms of the branching factor as follows: $L_n(\omega) \leqslant K \cdot L_{n-1}(\omega)$ for all $n > 0$. Since the number of different points proposed after n steps of computation, when f runs over $\mathcal{F}$, satisfies

$$|\{x_{w,n}^{(f)} \ : \ f \in \mathcal{F}\}| \leqslant L_n(\omega),$$

the algorithm converges slowly if $L_n(\omega)$ is small. This is formally quantified by the following result from Teytaud and Gelly [26], restricted here to our purpose, relating the convergence rate to the branching factor of a $(\mu \dot{+} \lambda)$-algorithm.

Theorem 1 (Lower bound on the convergence rate of $(\mu \dot{+} \lambda)$-ES.) *Consider a Full Ranking (μ, λ)-ES or $(\mu + \lambda)$-ES, as defined in Algorithm 2, and a set $\mathcal{F}$ of fitness functions on domain D, i.e. $\mathcal{F} \subset \mathbb{R}^D$, such that any fitness function $f \in \mathcal{F}$ has only one min-argument f^*, and such that $\{f^* \ : \ f \in \mathcal{F}\} = D$. Let $\varepsilon > 0$ and $0 \leqslant \delta < 1$. Let $L_n(\omega)$ be the number of different paths (when the function f runs over $\mathcal{F}$) followed by the algorithm on the random seed ω. Then*

$$\mathbb{E}_\omega[L_{n_{\varepsilon,\delta}}(\omega)] \geqslant (1-\delta)N(\varepsilon).$$

In particular, if K denotes the branching factor of the algorithm, then

$$n_{\varepsilon,\delta} \geqslant \frac{\log(1-\delta)}{\log K} + \frac{\log N(\varepsilon)}{\log K}.$$

The assumption $\{f^* \ : \ f \in \mathcal{F}\} = D$ simply means that the algorithm can't assume that the optimum of the fitness function is in a restricted subdomain of D. This is a natural assumption since it is satisfied as soon as the set of fitness functions is closed under translations. However, cases where $\{f^* \ : \ f \in \mathcal{F}\} \subsetneq D$ can also be considered by defining $N(\varepsilon)$ to be the maximum number of optima of fitness functions with mutual distances at least 2ε.

The convergence rate is defined as

$$\mathrm{CR}_\varepsilon^{(\mu,\lambda)} = \exp\left(-\frac{\log N(\varepsilon)}{d n_{\varepsilon,\frac{1}{2}}}\right).$$

The constant $\frac{1}{2}$ used here is arbitrary; results are essentially preserved with other constants (see the dependency in δ in Theorem 1). The faster the algorithm, the lower the convergence rate. Therefore, Theorem 1 provides a lower bound on the computational cost, and a lower bound on the convergence rate.

Theorem 1 can be reformulated as follows. Consider a $(\mu \dot{+} \lambda)$-ES satisfying the hypothesis of Theorem 1. Let $\alpha(\varepsilon) = (1 - 1/\log N(\varepsilon))^{-1}$. (We shall use this notation throughout the paper.) Then

$$n_{\varepsilon,\frac{1}{2}} \geqslant \frac{\log N(\varepsilon)}{\log K \cdot \alpha(\varepsilon)}. \tag{1}$$

In terms of the convergence rate, this gives

$$\mathrm{CR}_{\varepsilon}^{(\mu,\lambda)} \geqslant \exp\left(\frac{-\log K \cdot \alpha(\varepsilon)}{d}\right). \tag{2}$$

At last, please note that the expected runtime for reaching approximation error ε is at least $\frac{1}{2}n_{\varepsilon,\frac{1}{2}}$; therefore, results on $n_{\varepsilon,\frac{1}{2}}$ also provide lower bounds on the expected runtime.

4 Bounds on the convergence rate using VC-dimension

Lower bounds on the convergence rate of evolution strategies can be obtained from Equation 2 (Section 3) as soon as an upper bound on the branching factor K is known. Therefore, bounding the branching factor K is crucial.

The first solution for bounding the branching factor has been published in Teytaud and Gelly [26], using the fact that the number of subsets of size μ of a set of λ points (and thus the branching factor K) is at most $\binom{\lambda}{\mu} \leqslant \binom{\lambda}{\lfloor \lambda/2 \rfloor}$.

This surely holds, but it is a worst case on all possible selections. If we have the additional hypothesis that the fitness functions are *reasonable*, many subsets of size μ of a subset of size λ cannot be realized by a selection step. As an example, consider the case of a selection based algorithm with $\lambda = 6$ and $\mu = 3$, optimizing the set of sphere functions $\mathcal{F} = \{x \longmapsto \|x - x_0\|_2 \ : \ x_0 \in \mathbb{R}^2\}$ in the plane. Consider the offspring $O = \{a, b, c, d, e, f\} \subset \mathbb{R}^2$ as shown in Figure 1. The subset $\{a, b, c\}$ cannot be selected: indeed, it should correspond to the three points among O with the smallest fitness values with respect to some fitness function $x \longmapsto \|x - x_0\|_2$. This means there should exist a disk D centered on x_0 in the plane such that $D \cap O = \{a, b, c\}$: this is not possible since d lies in the convex hull of $\{a, b, c\}$. Other triplets such as $\{a, c, f\}$ cannot be selected either for the same reason. Of course this is not specific to the configuration of points presented in Figure 1: taking this effect into account gives a smaller bound on the branching factor.

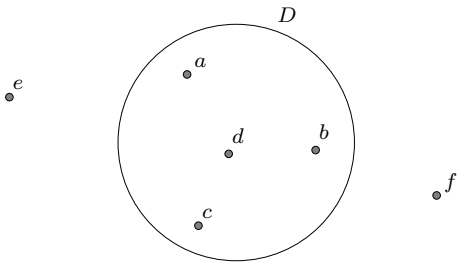

Figure 1: The set $\{a, b, c\}$ cannot be selected when optimizing sphere functions.

The phenomenon depicted on the example above is precisely quantified in the general case by the theory of VC-dimension and the shatter function lemma (also known as Sauer's lemma). In this section, we show how a VC-dimension hypothesis on the fitness functions permits improved lower bounds on the convergence rate of $(\mu \dot{+} \lambda)$-ES.

This section is organized as follows. We first recall basic facts on VC-dimension in Section 4.1. The VC-dimension hypothesis we need on fitness functions is stated in Section 4.2, where a summary of results obtained in this section is given. Next three subsections are devoted to the proofs of the main results. In Section 4.3, we use the VC-dimension arguments in order to obtain lower bounds on the convergence rate of Selection Based (μ, λ)-ES.

Section 4.4 is devoted to the case of non-elitist full ranking strategies. We explain how these bounds can be adapted to the case of elitist algorithms in Section 4.5. Finally, a method to handle the case of equality (i.e., when it is possible for two generated points x and y to satisfy $f(x) = f(y)$) is presented in Section 4.6.

4.1 VC-dimension and the shatter function lemma

We now briefly recall the definition of VC-dimension and the shatter function lemma [28, 24] – our presentation is based on [18]. A set system on a set A is a family $\mathcal{S}$ of subsets of A. For $B \subseteq A$, we define the restriction of $\mathcal{S}$ to B as $\mathcal{S}|_B = \{S \cap B \ : \ S \in \mathcal{S}\}$. The VC-dimension of the set system $\mathcal{S}$ defined over A is defined as $\sup\{|B| \ : \ \mathcal{S}|_B = 2^B\}$ where 2^B denotes the power set of B; in other words, it is the size of the largest subset B of A such that any subset of B can be obtained by intersecting B with an element of $\mathcal{S}$. Given a set system $\mathcal{S}$ over A, the *shatter function* $\pi_{\mathcal{S}}$ is defined by

$$\pi_{\mathcal{S}}(m) = \max\left\{\left|\mathcal{S}|_B\right| \ : \ B \subseteq A, \ |B| = m\right\}.$$

Thus $\pi_{\mathcal{S}}(m)$ is the maximum number of different subsets of A which can be obtained by intersecting a single subset of size m of A with all elements of $\mathcal{S}$. We next recall the following lemma which gives an upper bound on $\pi_{\mathcal{S}}$ in terms of the VC-dimension of $\mathcal{S}$.

Lemma 2 (Shatter function lemma) *For any set system $\mathcal{S}$ of VC-dimension d, then for all integer m, it holds that $\pi_{\mathcal{S}}(m) \leqslant \sum_{i=0}^{d} \binom{m}{i}$.*

At last, let us recall the following classical bound [8] which is valid whenever $d \geqslant 3$:

$$\sum_{i=0}^{d} \binom{m}{i} \leqslant m^d. \tag{3}$$

Note that the trivial bound $\sum_{i=0}^{d} \binom{m}{i} \leqslant 2^m$ is tight when $m \leqslant d$. The interesting case happens when m is large with respect to the VC-dimension d: the bound stated in Equation 3 becomes polynomial in m in this case.

4.2 The VC-dimension hypothesis on fitness functions

Given a function f defined over a domain D and $r > 0$, let

$$B_{f,r} = \{x \in D \ : \ f(x) < r\}.$$

We define the *level sets* $L_{\mathcal{F}}$ of a set of functions $\mathcal{F}$ defined over D as

$$L_{\mathcal{F}} = \{B_{f,r} \ : \ f \in \mathcal{F}, \ r > 0\}.$$

In this section, we establish lower bounds on the convergence rate of algorithms optimizing $\mathcal{F}$ in terms of the VC-dimension of the level sets $L_{\mathcal{F}}$.

The intuitive idea in the case of selection based algorithms is the following. The set of points selected from the offspring O at some step is necessarily the trace on O of some level set, *i.e.* it belongs to $\{O \cap B \ : \ B \in L_{\mathcal{F}}\}$. If the VC-dimension of $L_{\mathcal{F}}$ is finite, the shatter

function lemma provides a bound on the possible number of subsets selected, and thus on the branching factor, which improves on the naïve combinatorial bound.

Lower bounds on the convergence rate of evolution strategies obtained in this section are summarized in Figure 2. (We recall that $\alpha(\varepsilon) = (1 - 1/\log N(\varepsilon))^{-1}$ – see Section 3.) In the rest of this section, we shall state bounds for set systems of VC-dimension at least 3. However, the case of VC-dimension smaller than 3 can be handled in a similar way: the bound stated in Equation 3 has to be replaced with $\sum_{i=0}^{d} \binom{m}{i} \leqslant m^d + 1$.

	Lower bound on $\mathrm{CR}_{\varepsilon}^{(\mu,\lambda)}$
Selection Based (μ, λ)-ES	$\exp\left(-\frac{V}{d} \log \lambda \cdot \alpha(\varepsilon)\right)$
Selection Based $(\mu + \lambda)$-ES	$\exp\left(-\frac{V}{d} \log(\mu + \lambda) \cdot \alpha(\varepsilon)\right)$
Full Ranking (μ, λ)-ES	$\exp\left(-\frac{V}{d}(4\mu + \log \lambda) \cdot \alpha(\varepsilon)\right)$
Full Ranking $(\mu + \lambda)$-ES	$\exp\left(-\frac{V}{d}(4\mu + \log(\mu + \lambda)) \cdot \alpha(\varepsilon)\right)$

Figure 2: Lower bound on the convergence rate when the level sets of fitness functions have finite VC-dimension V in $\mathbb{R}^d$.

We now give a couple of applications based on classical set systems of bounded VC-dimension in $\mathbb{R}^d$ [8]. Axis-parallel hyper-rectangles have a VC-dimension bounded by $2d$. Sphere functions in $\mathbb{R}^d$ (for the Euclidean norm) have a VC-dimension equal to $d + 1$. The VC-dimension of ellipsoids in $\mathbb{R}^d$ is bounded by $d(d+1)/2 + d$. More generally, subsets of $\mathbb{R}^d$ defined by polynomial inequalities involving a finite number of real parameters, and Boolean combinations of these, have a finite VC-dimension [18, Chapter 10.3]. Hence, any algorithm of type $(\mu \overset{+}{,} \lambda)$-ES optimizing a set of fitness functions with these level sets on a domain $D \subset \mathbb{R}^d$ has a convergence rate which satisfies the bounds given in Figure 2. Let us remark that the sphere functions, i.e., the set of fitness functions $\mathcal{F} = \{f_c \,:\, c \in \mathbb{R}^d\}$ where $f_c(x) = \|x - c\|_2$, are a special case of functions with spheres as level sets, but are not the only ones (the same remark applies to hyper-rectangles and ellipsoids, see Figure 3).

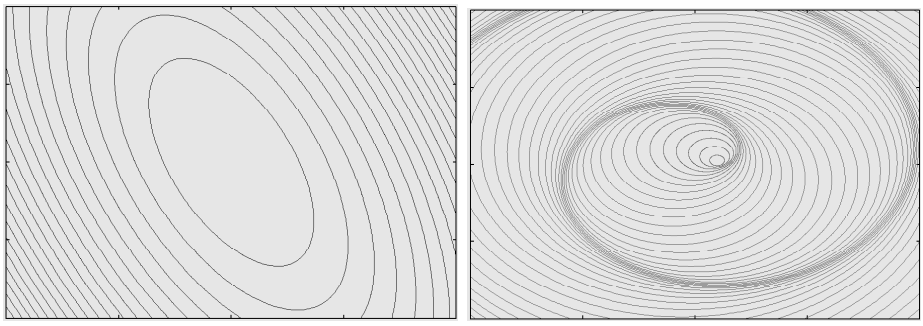

Figure 3: Left: level sets of a quadratic positive definite form: the level sets are ellipsoids. Right: an exemple of function with ellipsoids as level sets, without being quadratic. The VC-dimension bound given in the text holds in both cases.

4.3 Selection based non-elitist strategies

In this section, we give a lower bound on the convergence rate of Selection Based (μ, λ)-ES (strategies matching the framework described in Algorithm 1). The following lemma provides an upper bound on the branching factor in terms of the VC-dimension of the level sets of the fitness functions $\mathcal{F}$.

Lemma 3 *Consider an SB-(μ, λ)-ES as described in Algorithm 1. Let $V \geqslant 3$ be the VC-dimension of the level sets of the family $\mathcal{F}$ of fitness functions under consideration. Then the branching factor of this algorithm satisfies $K \leqslant \lambda^V$.*

Proof: Given a set of λ points $E = \{x_1, \ldots, x_\lambda\}$ in the domain D, and $f \in \mathcal{F}$, let us define $S_f(E)$ to be the subset S of size μ of E corresponding to the μ points of E with lowest fitness values with respect to f. Note that the branching factor satisfies

$$K \leqslant \max_{E \subset D,\, |E|=\lambda} |\{S_f(E) \ : \ f \in \mathcal{F}\}|.$$

Now note that for any E, the set S of the μ points of E with lowest value (with respect to the fitness function f) can be separated from $E \setminus S$ by an element from the level sets: in other words, there exists $B \in L_{\mathcal{F}}$ such that $B \cap E = S$. It follows that

$$|\{S_f(E) \ : \ f \in \mathcal{F}\}| \leqslant \pi_{L_{\mathcal{F}}}(\lambda).$$

If the VC-dimension of $L_{\mathcal{F}}$ is at most V, it follows from the shatter function lemma (Lemma 2) and the bound given in Equation 3 that $\pi_{L_{\mathcal{F}}}(\lambda) \leqslant \lambda^V$. Thus $K \leqslant \lambda^V$. □

Theorem 4 (Selection Based (μ, λ)-ES) *Consider an SB-(μ, λ)-ES (Algorithm 1) in a domain $D \subset \mathbb{R}^d$, such that $D = \{f^* \ : \ f \in \mathcal{F}\}$. Let $V \geqslant 3$ be the VC-dimension of the level sets of $\mathcal{F}$. The convergence rate of this algorithm satisfies*

$$\mathrm{CR}_\varepsilon^{(\mu,\lambda)} \geqslant \exp\left(-\frac{V \log \lambda}{d} \cdot \alpha(\varepsilon)\right).$$

Proof: Lemma 3 shows that $K \leqslant \lambda^V$, *i.e.*

$$\frac{\log K}{d}\alpha(\varepsilon) \leqslant V\frac{\log \lambda}{d}\alpha(\varepsilon).$$

The result follows by substituting this upper bound into Equation 2 from Theorem 1. □

Remark. If $V = 1$ or $V = 2$, then the bound obtained in Theorem 4 becomes $\mathrm{CR}_\varepsilon^{(\mu,\lambda)} \geqslant \exp\left(-\frac{V \log(\lambda+1)}{d} \cdot \alpha(\varepsilon)\right)$. The case $V < 3$ can be handled in a similar way throughout the paper: $V \log \lambda$ is replaced with $V \log(\lambda + 1)$ in this case.

The bound obtained in Theorem 4 is interesting when λ is large, and μ not too close to 1 or λ. Otherwise, the naïve combinatorial bound $K \leqslant \binom{\lambda}{\mu}$ leads to a smaller branching factor than Lemma 3. Combined with Equation 2 as above, it yields better bounds on the convergence rate when μ is close to 1 or λ.

4.4 Full ranking non-elitist strategies

We now consider Full Ranking (μ, λ) evolution strategies. That is, we move from Algorithm 1 to the more general setting of Algorithm 2. We study the extent to which lower bounds obtained for SB-(μ, λ)-ES are modified when we use the full ranking information.

For evolution strategies of type Full Ranking (μ, λ)-ES, an upper bound on the number of possible outcomes of the selection step (including the ranking of children) is obtained by multiplying by $\mu!$ the number of possible outcomes in the case of selection only. This gives $\mathrm{CR}_\varepsilon^{(\mu,\lambda)} \geqslant \exp\left(-(V\log(\lambda) + \mu\log\mu)/d \cdot \alpha(\varepsilon)\right)$. However, a stronger bound can be proved in the case where μ is large with respect to the VC-dimension V of the level sets of the fitness functions.

The following lemma provides an upper bound on the number of possible orders of a fixed set of points with respect to fitness functions $f \in \mathcal{F}$, when the level sets of $\mathcal{F}$ have a finite VC-dimension.

Lemma 5 *Let $\mathcal{F}$ be a set of functions on a domain D. Let $V \geqslant 3$ be the VC-dimension of level sets of $\mathcal{F}$. Let $x_1, \ldots, x_n$ be distinct points in D. The number of permutations π of $[[1, n]]$ such that there exists $f \in \mathcal{F}$ satisfying*

$$f(x_{\pi(1)}) < f(x_{\pi(2)}) < \ldots < f(x_{\pi(n)})$$

is at most 2^{4Vn}.

Proof: Let $\gamma(n)$ denote the maximum number of permutations realized by a fixed set of n points of D with respect to all functions of $\mathcal{F}$. Let p be the integer satisfying $2^{p-1} < n \leqslant 2^p$. Let $n' = 2^p$.

A possible order on n' points is completely determined by the $n'/2$ points with smallest values with respect to f, multiplied by the number of possible orders on two sets of $n'/2$ points. Therefore, by Lemma 3, $\gamma(n') \leqslant {n'}^V \gamma(n'/2)^2$.

By iteratively splitting the original set until we get sets of size 2, we obtain:

$$\gamma(n') \leqslant {n'}^V \left(\frac{n'}{2}\right)^{2V} \cdots \left(\frac{n'}{2^{p-1}}\right)^{2^{p-1}V}.$$

It follows that

$$\log\gamma(n') \leqslant V\left(\sum_{i=0}^{p-1} 2^i \log\left(\frac{n'}{2^i}\right)\right).$$

Of course $n'/2^i = 2^{p-i}$. Moreover, $\sum_{i=0}^{p-1} 2^i(p-i) = 2^{p+1} - p - 2 \leqslant 2n' \leqslant 4n$. This gives $\log\gamma(n) \leqslant 4Vn$. □

Theorem 6 (Full Ranking (μ, λ)-ES) *Consider a (μ, λ)-ES (Algorithm 2) in a domain $D \subset \mathbb{R}^d$, such that $D = \{f^* \,:\, f \in \mathcal{F}\}$. Let $V \geqslant 3$ be the VC-dimension of the level sets of $\mathcal{F}$. The convergence rate of this algorithm satisfies*

$$\mathrm{CR}_\varepsilon^{(\mu,\lambda)} \geqslant \exp\left(-\frac{V\,(4\mu + \log\lambda)}{d} \cdot \alpha(\varepsilon)\right).$$

Proof: The branching factor of this algorithm is bounded by $K \leqslant \lambda^V \gamma(\mu)$ where $\gamma(\mu)$ is the possible number of orders on the μ selected points with respect to fitness values. Lemma 5 shows that $\log\gamma(\mu) \leqslant 4V\mu$. Then, Equation 2 yields the lower bound on the convergence rate stated above. □

4.5 Elitist strategies

For the sake of completeness, we state the analog of previous results in the elitist setting.

Corollary 7 (Selection Based $(\mu+\lambda)$-ES and Full Ranking $(\mu+\lambda)$-ES) *Let $\mathcal{F}$ be a set of fitness functions defined in a domain $D \subset \mathbb{R}^d$, such that $D = \{f^* \ : \ f \in \mathcal{F}\}$. Let V be the VC-dimension of the level sets of $\mathcal{F}$. The convergence rate of an algorithm of type Selection Based $(\mu+\lambda)$-ES for $\mathcal{F}$ satisfies:*

$$\mathrm{CR}_{\varepsilon}^{(\mu,\lambda)} \geqslant \exp\left(-\frac{V\log(\mu+\lambda)}{d}\cdot\alpha(\varepsilon)\right).$$

For an algorithm of type Full Ranking $(\mu+\lambda)$-ES, the following holds:

$$\mathrm{CR}_{\varepsilon}^{(\mu,\lambda)} \geqslant \exp\left(-\frac{V\left(4\mu+\log(\mu+\lambda)\right)}{d}\cdot\alpha(\varepsilon)\right).$$

Proof: Any algorithm of type SB-$(\mu+\lambda)$-ES can be simulated by an algorithm of type SB-(μ',λ')-ES with $\mu'=\mu$ and $\lambda'=\mu+\lambda$. Indeed, this can be achieved by remembering the best μ points found at step n, and generate them at step $n+1$, together with λ new points. Bounds stated above on the convergence rate of elitist (μ,λ)-ES are thus obtained easily from Theorem 4.

In the same way, a Full Ranking $(\mu+\lambda)$-ES can be simulated by a Full Ranking (μ',λ')-ES. The bound on the convergence rate of elitist $(\mu+\lambda)$-ES follows by Theorem 6. □

4.6 Handling points with equal fitness values

This section explains how to deal with the general case where equality between fitness values of generated points, at any epoch, is allowed to occur. One must define how Algorithms 1 and 2 behave in this case, since several selection vectors (denoted by v_n in the algorithms) may satisfy the required properties when two points of the considered set E have the same fitness. We first consider algorithms where the selection procedure returns the smallest vector v_n of length μ with respect to lexicographic order, as formalized below.

We begin with the case of Selection Based (μ,λ)-ES. Consider such an algorithm for a set of fitness functions $\mathcal{F}$. We assume that the level sets of $\mathcal{F}$ have a VC-dimension $V \geqslant 3$. Our aim is to bound the branching factor K of such an algorithm (i.e., we shall state an analog of Lemma 3 when equality is allowed).

Consider a fixed vector of points $E=(x_1,\ldots,x_\lambda)\in D^\lambda$. For $f\in\mathcal{F}$, let σ be the smallest permutation of $\{1,\ldots,\lambda\}$ with respect to lexicographic order such that:

$$f(x_{\sigma(1)}) \leqslant f(x_{\sigma(2)}) \leqslant \ldots \leqslant f(x_{\sigma(\lambda)}). \tag{4}$$

This means that σ satisfies the inequalities above and obeys the following constraint: for all $j,k\in\{1,\ldots,\lambda\}$, if $j\leqslant k$ and $f(x_j)=f(x_k)$, then $\sigma(j)\leqslant\sigma(k)$. The selected set is $\{x_{\sigma(1)},x_{\sigma(2)},\ldots,x_{\sigma(\mu)}\}$.

We shall now give a bound on the branching factor K, *i.e.* a bound on

$$\sup_E \left|\left\{\{x_{\sigma(1)},x_{\sigma(2)},\ldots,x_{\sigma(\mu)}\} \ : \ f\in\mathcal{F}\right\}\right|.$$

Let $p \in [[1, \lambda]]$ be the smallest integer such that $f(x_{\sigma(p)}) = f(x_{\sigma(\mu)})$. In the same way, let $q \in [[1, \lambda]]$ be the greatest integer such that $f(x_{\sigma(q)}) = f(x_{\sigma(\mu)})$. Notice that p and q are uniquely defined from E and f.

Obviously, there exist two level sets A and B of f such that $\{x_{\sigma(1)}, \ldots, x_{\sigma(p-1)}\} = A \cap E$ and $\{x_{\sigma(1)}, \ldots, x_{\sigma(q)}\} = B \cap E$.

The set of selected points $\{x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(\mu)}\}$ is the union of $A \cap E$ and the $\mu - |A \cap E|$ points with smallest ranges (in the vector E) from $(B \cap E) \setminus (A \cap E)$.

Hence, the number of possible selections is bounded from above by the number of possible pairs $(A \cap E, B \cap E)$, where A and B are arbitrary level sets:

$$K \leqslant \sup_E |\{(A \cap E, B \cap E) \ : \ f \in \mathcal{F}\}| .$$

On the other hand, $|\{A \cap E \ : \ f \in \mathcal{F}\}| \leqslant \pi_{L_{\mathcal{F}}}(\lambda)$ and $|\{B \cap E \ : \ f \in \mathcal{F}\}| \leqslant \pi_{L_{\mathcal{F}}}(\lambda)$. (We recall that $\pi_{L_{\mathcal{F}}}$ is the shatter function.) We deduce that the number of possible selections for this fixed E, when f runs over $\mathcal{F}$, is bounded by $\pi_{L_{\mathcal{F}}}(\lambda)^2$. Hence, the branching factor of this algorithm satisfies:

$$K \leqslant \pi_{L_{\mathcal{F}}}(\lambda)^2. \tag{5}$$

Under the hypothesis of Theorem 4 and using the same notations, we have shown that the convergence rate of an algorithm of type Selection Based (μ, λ)-ES optimizing functions from $\mathcal{F}$ satisfies

$$\mathrm{CR}_{\varepsilon}^{(\mu,\lambda)} \geqslant \exp\left(-2\frac{V \log \lambda}{d} \cdot \alpha(\varepsilon)\right) \tag{6}$$

in the general case – that is, when equality is allowed to occur.

If, instead of constraining the selection by the lexicographic order, we decide that the algorithm is allowed to know both $A \cap E$ and $B \cap E$, then the branching factor still satisfies the inequality $K \leqslant (\pi_{L_{\mathcal{F}}}(\lambda))^2$. Hence, the bound given in Equation 6 also holds in algorithms where the selection step is based on this information. This includes the natural case where the selection at a step is chosen uniformly at random among all valid selections. (We recall here that, although the generation of points is deterministic in our model, this can be simulated through the use of the random state I.)

The same technique applies to full ranking algorithms. In this case, we assume the selection step is performed through the following protocol. First the algorithm is given the (ranges of) elements of $A \cap E$ and $B \cap E$. Then it must choose some subset S' of $(B \cap E) \setminus (A \cap E)$ of size $\mu - |A \cap E|$. Finally, the algorithm is returned the full ranking of the selected points $(A \cap E) \cup S'$. We emphasis here that the algorithm is not given the full ranking of all points of $B \cap E$. The upper bound on the branching factor in proof of Theorem 6 becomes $K \leqslant (\pi_{L_{\mathcal{F}}}(\lambda))^2 \gamma(\mu)$; that is, $K \leqslant (\lambda^V)^2 \cdot \gamma(\mu)$. Hence, the convergence rate is bounded by

$$\mathrm{CR}_{\varepsilon}^{(\mu,\lambda)} \geqslant \exp\left(-\frac{V\,(4\mu + 2\log\lambda)}{d} \cdot \alpha(\varepsilon)\right)$$

for Full ranking strategies in the general case.

Finally, the elitist case is deduced from the non-elitist case in the same way as in the proof of Corollary 7, by a simulation argument. Hence, the four bounds given in Figure 2 are valid modulo a multiplicative factor 2 in front of $\log \lambda$ (or $\log(\mu + \lambda)$ in the elitist cases) in the general case where equality of fitness values is allowed to occur.

Applications to discrete domains

We shall consider (μ, λ)-ES for fitness functions defined over a discrete domain. Since many classical benchmark functions in the discrete case have points with equal fitness values, we shall use the bounds obtained above when using the VC-dimension argument. As we consider arbitrary generation and update rules, elitist strategies are indirectly considered as well: indeed, $(\mu + \lambda)$-ES can be simulated by (μ, λ')-ES, with $\lambda' = \mu + \lambda$.

We shall give lower bounds on the number of steps $n_{0,\frac{1}{2}}$ needed to solve problems with approximation error 0 and confidence $\frac{1}{2}$. As usual, the constant $\frac{1}{2}$ is somewhat arbitrary; see Theorem 1 and remarks thereafter for the dependency in the confidence $1 - \delta$. As already mentioned, lower bounds on $n_{0,\frac{1}{2}}$ can be translated into lower bounds on the runtime expectation.

We consider below (i) the OneMax function, analyzed with simple combinatorial arguments; (ii) the sphere function on $\{0, 1, 2, \ldots, p-1\}^d$; (iii) the sphere function on permutations of $\{1, 2, \ldots, p\}$.

(i) Application to OneMax**.** Consider the domain $D = \{0, 1\}^d$. For ε sufficiently small, the balls are singletons; it follows that $N(\varepsilon) = N(0) = 2^d$ and $\alpha(\varepsilon) = \alpha(0) = 1/(1-1/d)$ when ε is small enough and $d \geqslant 2$. Let us consider the OneMax function defined by $x \longmapsto \sum_{i=1}^{d} x_i$, and all its symmetries; the set of fitness functions is

$$\mathcal{F} = \{f_\eta : x \longmapsto \sum_{i \in [[1,d]]} |x_i - \eta_i|,\ \eta \in \{0, 1\}^d\}.$$

The aim is to find the point where this function is maximum (hence, the selection step of an $(\mu \overset{+}{,} \lambda)$-ES keeps μ points with *highest* fitness values). Notice that no $(\mu \overset{+}{,} \lambda)$-ES can avoid generating points with equal fitness values in a single step as soon as $\lambda \geqslant 3$. Indeed, given three different points of $\{0, 1\}^d$, there must be two of them x and y such that the Hamming weight of the symmetric difference of x and y is even; then there exists η such that $f_\eta(x) = f_\eta(y)$.

Let us discuss the case of Selection Based (μ, λ)-ES. The bound on the convergence rate obtained above yields the following runtime for solving OneMax in dimension d with probability 1/2:

$$n_{0,\frac{1}{2}} = \Omega\left(d/\lambda\right) \text{ for } (\mu, \lambda)\text{-ES}; \tag{7}$$

$$n_{0,\frac{1}{2}} = \Omega\left(d/\log\lambda\right) \text{ for } (1, \lambda)\text{-ES}; \tag{8}$$

$$n_{0,\frac{1}{2}} = \Omega\left(d/\log(\lambda+1)\right) \text{ for } (1+\lambda)\text{-ES}. \tag{9}$$

This is obtained as in [26], using Equation 1 and the bound:

- $K \leqslant \binom{\lambda}{\lfloor \lambda/2 \rfloor}$ for (μ, λ)-ES;
- $K \leqslant \lambda$ for $(1, \lambda)$-ES;
- $K \leqslant 1 + \lambda$ for $(1 + \lambda)$-ES.

Equation 7 is more general than results established without the branching factor as it is not limited to $(1, \lambda)$-ES or $(\mu+1)$-ES; we will discuss tightness below. Equation 9 corresponds to a lower bound $\Omega(d\lambda/\log\lambda)$ on the number of evaluations; this is better than the $\Omega(d \log d)$ bound from [10, 30] when $\lambda/\log\lambda$ is larger than $\log d$.

Equations 7-9 provide both:

- improved lower bounds for λ large (*e.g.* parallel case) for $(1+\lambda)$-ES;
- improved lower bounds on what can be done with (μ, λ)-ES, as our lower bounds are tighter, for the comparison-based case (which is known optimal for compositions of the fitness functions with increasing mappings, see [11] for more on this), than the bounds with no assumption on the algorithm as in e.g. [10, 30].

We can give some elements of tightness for Equation 7 in the model where the algorithm has access to the whole set of points in $B \cap E$ (using the notations introduced at the beginning of this section). Notice that this is a quite strong model; in particular, $|B \cap E|$ might be greater than μ. More precisely, Equation 7 is tight in this model in the special cases $\lambda = d+1$ and $\lambda = O(1)$ (independent of d), as follows.

For $\lambda = d+1$, one iteration is enough to compute η (and thus the maximum of the fitness function f_η under consideration). Consider $\mu = 1$; however, as the algorithm has access to $B \cap E$, the selection step may return more than one point. Let $e_i \in \{0,1\}^d$ be the vector with a unique 1 in position i. Generate the set of points $E = \{(0,\dots,0), e_1, e_2, \dots e_d\}$. Let A and B be two level sets satisfying the conditions stated at the beginning of this section (transposed into the setting of a maximization problem). If $B \cap E = \{(0,\dots,0)\}$, then $\eta = 0$. Otherwise, let $\{e_{i_1}, \dots, e_{i_p}\} = B \cap E$; it is easily seen that $\eta = e_{i_1} + \dots + e_{i_p}$ in this case. We have therefore shown that one iteration is enough for finding the optimum.

Consider now the case $\lambda = O(1)$. We can perform the algorithm above restricted to a moving window of λ coordinates, as follows: generate $E_p = \{e_{(p-1)\lambda+1}, \dots, e_{p\lambda}\}$ at the p-th iteration, for $1 \leqslant p \leqslant \lceil d/\lambda \rceil$. (The last step can generate less than λ points.) This algorithm can compute η with runtime d/λ.

For the sake of parallelization of OneMax solving with ES, this suggests the use of λ linear in the dimension d. Indeed, for a parallel evaluation of the λ generated points, we get an almost linear speed-up for $\lambda \leqslant d$ – the convergence rate is $\exp(-O(1))$ for $\lambda = d+1$ whereas it is $\exp(-O(1/d))$ for $\lambda = O(1)$ – while the speed-up is logarithmic in λ for large values of λ.

(ii) Application to the sphere function on grids. We consider the set of sphere functions on the domain $D = \{0, 1, 2, \dots, p-1\}^d \subseteq \mathbb{R}^d$, *i.e.*

$$\mathcal{F} = \{x \longmapsto \|x - x_0\|_2 \ : \ x_0 \in D\}.$$

We consider the case of (μ, λ)-ES. Since equality might occur, we shall apply Equation 5 to bound the branching factor. The VC-dimension of the level sets of $\mathcal{F}$ is equal to $d+1$ which gives $\log K \leqslant 2(d+1)\log\lambda$. Therefore Equation 1 leads to $n_{0,\frac{1}{2}} \geqslant (d \log p)/(\alpha(0) 2(d+1) \log \lambda)$, where $\alpha(0) = 1/(1 - 1/d \log p)$ is close to 1. Hence we have obtained

$$n_{0,\frac{1}{2}} = \Omega\left(\log p / \log \lambda\right).$$

As usual, when λ is small, a better bound can be obtained using $K \leqslant \binom{\lambda}{\lfloor \lambda/2 \rfloor}$. This gives $n_{0,\frac{1}{2}} \geqslant (d \log p)/(\alpha(0) \log \binom{\lambda}{\lambda/2})$. This leads to the alternative lower bound

$$n_{0,\frac{1}{2}} = \Omega(d \log(p)/\lambda),$$

which is better than the previous bound when $\lambda / \log \lambda < d$. This generalizes *e.g.* [25, Theorem 2] by considering a wider family of algorithms and possibly $p > 2$.

(iii) Application to the sphere function on permutations. We now consider the set of sphere functions on permutations. Let D be the set of permutations of $\{1, 2, \ldots, p\}$ and

$$\mathcal{F} = \{x \longmapsto \|x - x_0\|_2 \ : \ x_0 \in D\}.$$

The domain D is often considered in applications, see *e.g.* [20, 19], references therein and variants of the traveling salesman problem (TSP). In many real world cases, the fitness function is neither the length of a path as in the TSP, nor some other white box function, but a black box function (otherwise practitioners would certainly not use an evolutionary algorithm, which is suboptimal for white box problems). We will here consider the simple case of a sphere function on D. We consider again the case of (μ, λ)-ES. This is equivalent to the sphere function on grids, except that $d = p$, $|D| = p!$ and $\alpha(0) = 1/(1 - 1/\log(p!))$. We obtain the bound $n_{0,\frac{1}{2}} = \Omega(\log(p!)/(p \log \lambda))$, *i.e.*, by Stirling's approximation, the runtime for a (μ, λ)-ES for the sphere function on the set of permutations of $\{1, 2, \ldots, p\}$ is

$$n_{0,\frac{1}{2}} = \Omega(\log p / \log \lambda).$$

5 Sign patterns and the case of the sphere function

We present a technique based on the number of sign patterns to derive lower bounds on the convergence rate of full ranking algorithms. Applied to the special case of the sphere function, it shows that the speed-up is asymptotically at most logarithmic in λ. (See Proposition 8.) This improves on the bounds obtained by VC-dimension arguments in Theorem 6 which did not forbid a linear speed-up.

Although it is applied to the sphere function, the technique used here applies to any system where the number of sign patterns can be efficiently bounded, such as quadratic functions with positive Hessian. In fact, polynomials of bounded degree are amenable to this technique – we refer the reader to [22] and [18, Chapter 6.2] for further details. However, as opposed to the previous section, the bound obtained by the sign pattern technique presented here does not apply to the general case of functions with spheres (or ellipsoids) as level sets.

For the sphere function, we point out in Section 5.2 that λ linear in the dimension provides an almost linear speed-up. Indeed, the straightforward parallelization performed by distributing the λ fitness evaluations over λ processors has an almost linear speed-up until at least λ equal to twice the dimension (while the speed-up is asymptotically at most logarithmic in λ by Proposition 8.)

5.1 Lower bounds on full ranking strategies via the number of sign patterns

We present an alternative method to obtain improved lower bounds on the convergence rate of evolution strategies which use full ranking. For a real x, we define its sign to be $\text{sign}(x) = 0$ if $x = 0$, $\text{sign}(x) = 1$ if $x > 0$, and $\text{sign}(x) = -1$ if $x < 0$. Given a fitness function f and a finite set of points $x_1, \ldots, x_\lambda \in D$, we define the set of realizable sign conditions as

$$\text{Sign}_{f,(x_1,\ldots,x_\lambda)} = \{\text{sign}(f(x_i) - f(x_j)) : 1 \leqslant i < j \leqslant \lambda\}.$$

This method can be applied as soon as the number of sign conditions, i.e. the number of possible sign vectors, can be efficiently bounded. Indeed, the following inequality on the

branching factor holds:

$$K \leqslant \max\left\{ |\mathrm{Sign}_{f,(x_1,\ldots,x_\lambda)}| : f \in \mathcal{F},\ x_1,\ldots,x_\lambda \in D \right\}. \tag{10}$$

We now apply the above remark to the sphere functions, in order to obtain an improved lower bound on the convergence rate of full ranking strategies for these functions. We recall that the set of sphere functions is the the set of fitness functions $\mathcal{F} = \{f_c \ :\ c \in \mathbb{R}^d\}$ with $f_c(x) = \|x - c\|_2$ (where $\|\cdot\|_2$ denotes the Euclidean norm).

Proposition 8 *Let $d \geqslant 3$. Consider a Full Ranking (μ, λ)-ES, as in Algorithm 2, optimizing the sphere function in a domain $D \subset \mathbb{R}^d$. Then*

$$\mathrm{CR}_\varepsilon^{(\mu,\lambda)} \geqslant \exp\left(-2\log(\lambda)\cdot\alpha(\varepsilon)\right).$$

Proof: Given two distinct points p and q in $\mathbb{R}^d$, we denote by $H_{p,q}$ be the mediator hyperplane of p and q, i.e. $H_{p,q} = \{x \in \mathbb{R}^d \ :\ \|x-p\|_2 = \|x-q\|_2\}$. At each iteration of the algorithm, an offspring of λ points $\{x_1,\ldots,x_\lambda\}$ is generated and the algorithm receives the sequence of ranges of the μ points with lowest fitness values, ordered with respect to their fitness values. Obviously the branching factor is maximal when $\mu = \lambda$, i.e. when the algorithm is given the full ordering of points with respect to their fitness values. This information corresponds to giving the sign $s_{i,j}$ of $f(x_i) - f(x_j)$ for each $1 \leqslant i < j \leqslant \lambda$. Note that this sign is positive or negative since we assumed equality never occurs. The number of possible sign vectors $s = \{s_{i,j} : 1 \leqslant i < j \leqslant \lambda\}$ in $\mathbb{R}^d$ is exactly the number of cells (full dimensional faces) in the arrangement of hyperplanes $\{H_{x_i,x_j} \ :\ 1 \leqslant i < j \leqslant \lambda\}$. But it is known that n hyperplanes in $\mathbb{R}^d$ define at most n^d cells [18, Chapter 6.1]. Since there are $\binom{\lambda}{2} \leqslant \lambda^2/2$ hyperplanes here, Equation 10 yields $K \leqslant \left(\lambda^2/2\right)^d$. The bound on the convergence rate is now obtained by applying Equation 2. □

Remark. The case of equality (i.e., when it is possible for two generated points x and y to satisfy $f(x) = f(y)$) is easily handled in Proposition 8. Indeed, an upper bound on the number of sign vectors obtained in this case is given by the total number of faces (of any dimension) of the hyperplanes arrangement considered in the proof. This number of faces is known to be $O(n^d)$ for n hyperplanes $\mathbb{R}^d$, where the constant hidden in the big-O notation depends on d. From the formula for the maximum number of faces of a given dimension [12], this constant is bounded by $d^{O(1)}2^d$. The upper bound on the branching factor K follows, and we obtain $\mathrm{CR}_\varepsilon^{(\mu,\lambda)} \geqslant \exp\left(-(2\log(\lambda) + O(1))\cdot\alpha(\varepsilon)\right)$.

5.2 Fast convergence rate with $\lambda = 2d$ for optimizing the sphere function

We point out here that for the specific case of the sphere function, a convergence rate of $\Theta(1)$ can be reached with $\mu = \lambda = 2d$ in the domain $[0,1]^d$ by some algorithm of type Full Ranking (μ,λ)-ES.

This convergence rate is easily obtained with the following algorithm in the spirit of Hooke and Jeeves *Direct Search* methods [14]. Let e_i denote the vector $(0,\ldots,0,1,0,\ldots,0)$ with a unique 1 in position i. First split $[0,1]^d$ into the 2^d cells delimited by the d hyperplanes of equations $x_i = 1/2$; the full ranking of the $2d$ points $\{(\frac{1}{2},\frac{1}{2},\ldots,\frac{1}{2}) + \frac{\eta}{2}e_i \ :\ 1 \leqslant i \leqslant n,\ \eta \in \{-1,1\}\}$ allows the algorithm to decide in which of these cells the optimum lies; then the algorithm proceeds recursively. (See Algorithm 3.)

Algorithm 3 Fast algorithm of type Full Ranking (μ, λ)-ES for the sphere function in the domain $[0,1]^d$ in the special case $\mu = \lambda = 2d$.

$x \leftarrow (1/2, \ldots, 1/2)$; $\sigma \leftarrow 1/2$
while true **do**
  Generate $\lambda = 2d$ distinct points equal to $x \pm \sigma e_i$
    (e_i denotes the vector $(0, \ldots, 0, 1, 0, \ldots, 0)$ with a unique 1 in position i)
  With the ranking information, decide in which octant Δ of $x + [-\sigma, \sigma]^d$ is the optimum
  Move x to the center of the octant Δ
  Set $\sigma \leftarrow \sigma/2$
end while

After n iterations, the point x_n proposed by this algorithm satisfies $\|x_n - f^*\|_2 \leqslant \sqrt{d}/2^n$. Moreover, this distance is realized by some fitness function. It follows that $n_{\varepsilon,\frac{1}{2}} = \log \frac{1}{\varepsilon} + \frac{1}{2} \log d$. On the other hand $\log(N(\varepsilon)) = \Theta(d \log \frac{1}{\varepsilon})$. Thus, we have obtained an algorithm for the sphere function in dimension d which satisfies:

$$\mathrm{CR}_{\varepsilon}^{(2d,2d)} = \exp\left(-\Theta(1)\right). \tag{11}$$

Notice that for $\lambda = O(1)$, independent from d, the branching factor of any algorithm satisfies $K = O(1)$; it follows by Equation 2 that any algorithm optimizing the sphere function in dimension d satisfies $\mathrm{CR}_{\varepsilon}^{(\lambda,\lambda)} \geqslant \exp\left(-O(1/d)\right)$ in this case. Hence, Algorithm 3 achieves an almost linear speed-up when λ moves from $O(1)$ to $2d$.

This means that, with $2d$ processors, the number of function evaluations required for halving the approximation error is $\Theta(d)$ (as well as for the $(1+1)$-ES [16]); or, in other words, with $2d$ processors, the number of iterations required for halving the error is $\Theta(1)$.

On the other hand, the asymptotic speed-up for λ large (and d fixed) is known to be at most logarithmic by Proposition 8.

6 Final remarks

It could seem to be a weakness that bounds on the convergence rate obtained by VC-dimension arguments are weaker when the function is more "complex" (i.e., when the VC-dimension of its level sets is higher). However, it may be possible that these bounds cannot be improved in the general case. Indeed, one can wonder if it is possible to build *ad hoc* fitness functions matching the bounds obtained by VC-dimension arguments. Such constructions were given in [26] to match lower bounds on the convergence rate of algorithms obtained from the sole branching factor. On the other hand, we know that the bounds obtained from VC-dimension can be far from optimal for some specific sets of fitness functions: for the sphere function, the bound obtained for Full Ranking (μ, λ)-ES is greatly improved by the sign pattern technique (Section 5).

In the case of evolution strategies based on selection only (algorithms of type SB-(μ, λ)-ES), the linear speed-up observed in [6] cannot be obtained for λ large enough. Asymptotically, the speed-up obtained with such an algorithm is at most logarithmic as shown in Theorem 4. However, we show that the speed-up is nearly linear for up to $2d$ processors on the sphere function in dimension d.

When moving from algorithms of type Selection Based (μ, λ)-ES to Full Ranking (μ, λ)-ES, lower bounds on the convergence rate obtained here in the general case do not forbid a

strong improvement asymptotically; essentially, the speed-up that could be achieved moves from logarithmic to linear in λ. However, we know from Proposition 8 that the speed-up is at most logarithmic for a Full Ranking (μ, λ)-ES in the special case of the sphere function – see also the discussion following Proposition 8. This raises the following question: For which kind of fitness functions is it interesting to keep the full ranking information?

A related intriguing question is what convergence rate can be reached for selection based algorithms (i.e., without keeping the full ranking information) for the sphere function. In particular, is it possible to achieve a constant convergence rate with λ linear in the dimension, as in Equation 11? To the best of our knowledge, this is an open problem.

Acknowledgments

We would like to thank Anne Auger, Nikolaus Hansen and Fabien Teytaud for constructive talks, and the anonymous referees for their helpful comments. This work was partially supported by the Pascal Network of Excellence.

References

[1] Dirk V. Arnold. Optimal weighted recombination. In *Foundations of Genetic Algorithms 8*, volume 3469 of *Lecture Notes in Computer Science*, pages 215–237. Springer-Verlag, Berlin Heidelberg, 2005.

[2] Anne Auger. Convergence results for (1,λ)-SA-ES using the theory of φ-irreducible Markov chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[3] Anne Auger, Marc Schoenauer, and Olivier Teytaud. Local and global order 3/2 convergence of a surrogate evolutionary algorithm. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 857–864, New York, NY, USA, 2005. ACM.

[4] Thomas Bäck, Frank Hoffmeister, and Hans-Paul Schwefel. Extended selection mechanisms in genetic algorithms. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 92–99, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[5] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 14–21, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.

[6] H.-G. Beyer. Toward a theory of evolution strategies: On the benefit of sex – the $(\mu/\mu, \lambda)$-theory. *Evolutionary Computation*, 3(1):81–111, 1995.

[7] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies: a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

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

[9] Stefan Droste. Not all linear functions are equally difficult for the compact genetic algorithm. In *Proc. of the Genetic and Evolutionary Computation COnference (GECCO 2005)*, pages 679–686, 2005.

[10] Stefan Droste, Thomas Jansen, and Ingo Wegener. A rigorous complexity analysis of the (1+1) evolutionary algorithm for separable functions with boolean inputs. *Evolutionary Computation*, 6(2):185–196, 1998.

[11] S. Gelly, S. Ruette, and O. Teytaud. Comparison-based algorithms are robust and randomized algorithms are anytime. *Evolutionary Computation Journal (MIT Press), Special issue on bridging Theory and Practice*, 15(4):411–434, 2007.

[12] D. Halperin. Arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 24, pages 529–562. CRC Press LLC, Boca Raton, FL, 2004.

[13] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

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

[15] Jens Jägersküpper. Analysis of a simple evolutionary algorithm for minimization in euclidean spaces. In *30th International Colloquium on Automata, Languages, and Programming (ICALP 2003)*, Springer LNCS 2719, pages 1068–1079, 2003.

[16] Jens Jägersküpper. Analysis of a simple evolutionary algorithm for minimization in euclidean spaces. *Theoretical Computer Science (special issue on ICALP 2003)*, 379(3):329–347, 2007.

[17] Jens Jägersküpper and Carsten Witt. Rigorous runtime analysis of a $(\mu + 1)$-ES for the sphere function. In *GECCO*, pages 849–856, 2005.

[18] Jiří Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer, 2002.

[19] Alberto Moraglio and Riccardo Poli. Topological crossover for the permutation representation. In *GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 332–338, New York, NY, USA, 2005. ACM.

[20] Frank Neumann. Expected runtimes of evolutionary algorithms for the eulerian cycle problem. *Computers & OR*, 35(9):2750–2759, 2008.

[21] I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.

[22] Lajos Rónyai, László Babai, and Murali K. Ganapathy. On the number of zero-patterns of a sequence of polynomials. *Journal of the American Mathematical Society*, 14(3):717–735, 2001.

[23] G. Rudolph. Convergence rates of evolutionary algorithms for a class of convex objective functions. *Control and Cybernetics*, 26(3):375–390, 1997.

[24] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Ser. A*, 13(1):145–147, 1972.

[25] Dirk Sudholt and Carsten Witt. Runtime analysis of binary PSO. In *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 135–142, New York, NY, USA, 2008. ACM.

[26] O. Teytaud and S. Gelly. General lower bounds for evolutionary algorithms. In *Proceedings of PPSN*, pages 21–31, 2006.

[27] Olivier Teytaud and Hervé Fournier. Lower bounds for evolution strategies using vc-dimension. In *PPSN*, pages 102–111, 2008.

[28] V.N. Vapnik and A.Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, XVI(2):264–280, 1971.

[29] Darrell Whitley. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 116–121, San Mateo, CA, 1989. Morgan Kaufman.

[30] Carsten Witt. Theory of randomised search heuristics in combinatorial optimisation: an algorithmic point of view. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 3551–3592, New York, NY, USA, 2009. ACM.