

## Lower bounds for evolution strategies using VC-dimension

Olivier Teytaud, Hervé Fournier

► **To cite this version:**

Olivier Teytaud, Hervé Fournier. Lower bounds for evolution strategies using VC-dimension. Parallel Problem Solving from Nature, Sep 2008, Dortmund, Germany. 10 p., 2008. <inria-00287845>

**HAL Id: inria-00287845**

**<https://hal.inria.fr/inria-00287845>**

Submitted on 13 Jun 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Lower bounds for evolution strategies using VC-dimension

Olivier Teytaud<sup>1</sup> and Hervé Fournier<sup>2</sup>

<sup>1</sup> TAO (Inria), LRI, UMR 8623 (CNRS - Univ. Paris-Sud), Bât 490  
Univ. Paris-Sud 91405 Orsay, France.

`teytaud@lri.fr`

<sup>2</sup> Laboratoire PRiSM

CNRS UMR 8144 and Univ. Versailles St-Quentin en Yvelines  
45 av. des États-Unis, 78035 Versailles, France.

`herve.fournier@prism.uvsq.fr`

**Abstract.** We derive lower bounds for comparison-based or selection-based algorithms, improving existing results in the continuous setting, and extending them to non-trivial results in the discrete case. We introduce for that the use of the VC-dimension of the level sets of the fitness functions; results are then obtained through the use of Sauer’s lemma. In the special case of optimization of the sphere function, improved lower bounds are obtained by bounding the possible number of sign conditions realized by some systems of equations. The results include several applications to the parametrization of sequential or parallel algorithms of type  $(\mu \dagger \lambda)$ -ES.

**Keywords:** Evolution Strategies; Convergence rate; VC-dimension; Sign conditions.

## 1 Introduction

Evolution strategies (ES), defined by Rechenberg [14], are a family of optimization algorithms with nice robustness properties. Most ES use only comparisons between fitness values and not the fitness values themselves. This fact has been used in [17] in order to provide lower bounds that match some upper bounds known for evolutionary algorithms in the continuous domain [8, 2, 15], and [10] has shown the optimality of this comparison-based principle for some robustness criterion (see also [3, 19, 4]). In [17] is provided a new tool for proving lower bounds for evolutionary algorithms, but, as pointed out by the authors, some bounds are not tight and in particular: (i) the discrete case provides essentially trivial results; (ii) the bounds for the  $(\mu, \lambda)$ -ES are far too large. In this work, we propose improved lower bounds for evolution strategies of type  $(\mu \dagger \lambda)$ -ES (i.e. upper bounds on the convergence rates of these algorithms) in term of the VC-dimension of level sets of the fitness functions. In the special case of optimization of the sphere function, improved upper bound on the convergence rate of evolution strategies are presented; they are obtained by bounding the number of sign conditions realized by a system of equations.

The paper is organized as follows. Basic definitions and terminology of evolution strategies we consider are described in Section 2. Lower bounds on  $(\mu \dagger \lambda)$ -ES based on the branching factor, obtained in [17], are recalled in Section 3. Improved lower bounds on  $(\mu \dagger \lambda)$ -ES in term of VC-dimension are presented in Section 4. At last, practical implications of the theory are discussed in Section 5.

*Notations.* In all the paper,  $\log(x)$  denotes the logarithm with basis 2, i.e.  $\log(2) = 1$ . The set of integers  $\{1, 2, \dots, n\}$  is denoted by  $[[1, n]]$ .

## 2 Evolution Strategies of type $(\mu \dagger \lambda)$

We define in this section  $(\mu \dagger \lambda)$ -algorithms – we refer to Beyer and Schwefel [6] for a comprehensive introduction to evolution strategies.

The aim of a  $(\mu \dagger \lambda)$ -algorithm is to find the minimum of a function  $f$  (called the fitness function) defined over a domain  $D$ . This algorithm cannot evaluate the function  $f$  but has to work only with comparisons: given two points  $x$  and  $y$ , the algorithm has access to a black-box telling whether  $f(x) < f(y)$ ,  $f(x) = f(y)$  or  $f(x) > f(y)$ . Of course such an algorithm is not required to work for one fitness function but for a whole family of fitness functions. In the following we denote by  $\mathcal{F}$  the set of fitness functions we consider.

In the rest of the paper, we assume we never have a case of equality  $f(x) = f(y)$  among the generated points in order to avoid technical difficulties. Let  $\lambda$  and  $\mu$  denote two integers (subject

---

**Algorithm 1** SB- $(\mu, \lambda)$ -ES (resp. SB- $(\mu + \lambda)$ -ES), i.e. evolution strategies based on selection, working on a fitness function  $f$ . The real number  $\omega$  is a random seed, uniform in  $[0, 1]$ . We do not specify the generation of offsprings, because we work on the whole family of algorithms matching this framework.

---

**Initialize**  $I_0 \in \mathcal{I}$ ,  $S_{-1} = \emptyset$  and  $n = 0$

**while true do**

**Generate** an offspring  $O_n$  of  $\lambda$  distinct points:  $O_n = \text{generate}(I_n, \omega)$ .

**Selection:** Use the fitness  $f$  in order to partition  $O_n$  (resp.  $O_n \cup S_{n-1}$ ) in two sets  $S_n$  of cardinal  $\min(\mu, \text{Card}(O_n))$  and  $R_n$  such that

$$x \in S_n \text{ and } y \in R_n \Rightarrow f(x) < f(y).$$

We denote this by  $S_n = \text{select}(O_n, f)$  (resp.  $S_n = \text{select}(O_n \cup S_{n-1}, f)$ ).

**Update** the internal state:  $I_{n+1} = \text{update}(I_n, f, O_n) = \text{selectionUpdate}(I_n, S_n, R_n) \in \mathcal{I}$ .

$x_{n+1}^{(f)} = \text{proposal}(I_n)$

$n = n + 1$

**end while**

---



---

**Algorithm 2**  $(\mu, \lambda)$ -ES (resp.  $(\mu + \lambda)$ -ES) based on full ranking, working on a fitness function  $f$ . The real number  $\omega$  is a random seed, uniform in  $[0, 1]$ . Compared to Algorithm 1,  $S_n$  is now a vector of points, ordered with respect to their fitness values. This family of algorithms is more general than Algorithm 1, as we can use all the ranking information.

---

**Initialize**  $I_0 \in \mathcal{I}$ ,  $S_{-1} = \emptyset$  and  $n = 0$

**while true do**

**Generate** an offspring  $O_n$  of  $\lambda$  distinct points:  $O_n = \text{generate}(I_n, \omega)$ .

**Selection with ranking:** Use the fitness  $f$  in order to partition  $O_n$  (resp.  $O_n \cup S_{n-1}$ ) in a vector  $S_n = (x'_1, \dots, x'_{c_n})$  of cardinal  $c_n = \min(\mu, \text{Card}(O_n))$  (resp.  $c_n = \min(\mu, \text{Card}(O_n \cup S_{n-1}))$ ) and a set  $R_n$  such that

$$\forall i \in [[1, c_n]], \forall y \in R_n, f(x'_i) < f(y),$$

$$\text{and } \forall i \in [[1, c_n - 1]], f(x'_i) < f(x'_{i+1}).$$

We denote this by  $S_n = \text{select}(O_n, f)$  (resp.  $S_n = \text{select}(O_n \cup S_{n-1}, f)$ ).

**Update** the internal state:  $I_{n+1} = \text{update}(I_n, f, O_n) = \text{fullRankUpdate}(I_n, S_n, R_n) \in \mathcal{I}$ .

$x_{n+1}^{(f)} = \text{proposal}(I_n)$

$n = n + 1$

**end while**

---

in the non-elitist  $(\mu, \lambda)$  case to the condition  $\mu \leq \lambda$ ). A SB- $(\mu \dagger \lambda)$ -ES (Selection Based  $(\mu \dagger \lambda)$ -ES) is an algorithm working as follows. There is a set  $\mathcal{I}$  of internal states and an initial state  $I_0$ . At each iteration, the algorithm follows these three successive steps. First generate a set of  $\lambda$  points, called the *offspring*. Then select only the  $\mu$  best ones, i.e. the  $\mu$  points with lowest fitness values; in the case of a SB- $(\mu, \lambda)$ -ES, points generated at previous stages are forgotten and this

selection is only among the offspring, while an algorithm of type SB- $(\mu + \lambda)$ -ES selects the  $\mu$  best points among the offspring *and* the points selected at the previous step (hence these  $\mu$  selected points are always the  $\mu$  points with lowest fitness value found so far). At last the internal state is updated. General outlines of SB- $(\mu, \lambda)$ -algorithms (resp. SB- $(\mu + \lambda)$ -algorithms) are summarized 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*. Elitist strategies are usually faster on easy fitness functions, but less robust; therefore, non-elitist strategies are usually preferred.

At last we would like to explain a generalization of SB- $(\mu \dagger \lambda)$ -ES, called  $(\mu \dagger \lambda)$ -ES. Instead of just giving the best  $\mu$  points (i.e. the  $\mu$  points with the lowest fitness value), we can consider a selection procedure which returns the best  $\mu$  points *ordered with respect to their fitness*. More precisely, given the points  $(y_1, \dots, y_p)$  ( $O_n$  in the case of  $(\mu, \lambda)$ -ES or  $O_n \cup S_{n-1}$  in the case of  $(\mu + \lambda)$ -ES), it returns  $\mu$  distinct indices  $(i_1, \dots, i_\mu)$  such that  $f(y_{i_1}) < \dots < f(y_{i_\mu})$  and for all  $j \notin \{i_1, \dots, i_\mu\}$ ,  $f(y_{i_\mu}) < f(y_j)$ . We call *full ranking* this kind of "selection" [4, 3, 19]. The outline of these algorithms is summarized in Algorithm 2.

Note that both Algorithms 1 and 2 define a class of algorithms: in order to obtain an algorithm, one has to specify how generation of points is done, what is the set of internal states as well as the update function. A usual 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.

### 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 number of disjoint open balls of radius  $\varepsilon$  that one can put in the domain  $D$ . That is,  $N(\varepsilon)$  is the maximum integer  $n$  such that there exist  $n$  distinct points  $x_1, \dots, x_n \in D$  with  $\|x_i - x_j\| \geq 2\varepsilon$  for all  $i \neq j$ .

If each function  $f \in \mathcal{F}$  has one and only one optimum  $f^*$ , for any given optimization algorithm as in Algorithm 2, and for  $\varepsilon > 0$  and  $\delta > 0$ , we let  $n_{\varepsilon, \delta}$  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  $\varepsilon$ . I.e.  $n_{\varepsilon, \delta}$  is minimal such that for all  $n \geq n_{\varepsilon, \delta}$  and for all  $f \in \mathcal{F}$ ,

$$\Pr_{w \in [0,1]}(\|x_n^{(f)} - f^*\| \leq \varepsilon) \geq 1 - \delta.$$

For an algorithm of type  $(\mu \dagger \lambda)$ -ES working over a set  $\mathcal{F}$  of fitness functions, we define the *branching factor* of any algorithm as in Algorithm 2 as

$$K = \sup_{I \in \mathcal{I}, O} \text{Card}\{\text{update}(I, f, O) \mid f \in \mathcal{F}\}.$$

Notice that in the case of selection based algorithms (any algorithm fitting Algorithm 1):

$$K \leq \sup_O \text{Card}\{\text{select}(O, f) \mid f \in \mathcal{F}\}$$

where the supremum holds for:

- $O$  any set of  $\lambda$  points in the case of SB- $(\mu, \lambda)$ -ES;
- $O$  any set of  $\lambda + \mu$  points in the case of SB- $(\mu + \lambda)$ -ES.

A similar remark holds in the case of full ranking  $(\mu \dagger \lambda)$ -ES, except that a bound on  $K$  is given by the possible number of choices of selected points together with their order (with respect to their fitness values).

Let us recall the following result from Teytaud and Gelly [17] (restricted here to our purpose) relating the convergence rate and the branching factor of a  $(\mu \dagger \lambda)$ -ES.

**Theorem 1 (Lower bound on the convergence rate of  $(\mu \dagger \lambda)$ -ES.)** Consider a  $(\mu, \lambda)$ -ES or  $(\mu + \lambda)$ -ES as in Algorithm 2. Consider a set  $\mathcal{F}$  of possible 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  $\{f^* \mid f \in \mathcal{F}\} = D$ . Let  $\varepsilon > 0$  and  $\delta \in ]0, 1[$ . Let  $K$  be the branching factor of this algorithm. Then

$$n_{\varepsilon, \delta} \geq \left\lceil \frac{\log(1 - \delta)}{\log(K)} + \frac{\log(N(\varepsilon))}{\log(K)} \right\rceil.$$

In the following all logarithms are in base 2. We can define the convergence rate for both discrete and continuous domains thanks to the following unified definitions. The *convergence rate* of an algorithm for precision  $\varepsilon$  is defined as

$$\text{CR}_\varepsilon = \frac{\log N(\varepsilon)}{dn_{\varepsilon, \frac{1}{2}}}.$$

For a finite domain  $D = \{0, 1\}^d$ ,  $N(0) = 2^d$  and the median of the parallel running time is  $1/\text{CR}_\varepsilon$  for  $\varepsilon = 0$ . For both discrete and continuous domains again, we define the *normalized convergence rate* (normalized by the number of individuals generated per epoch) by

$$\text{NCR}_\varepsilon = \frac{\log N(\varepsilon)}{d\lambda n_{\varepsilon, \frac{1}{2}}}.$$

For a finite domain  $D = \{0, 1\}^d$  as above, the sequential running time is  $1/\text{NCR}_\varepsilon$  for  $\varepsilon = 0$ .  $\text{CR}_\varepsilon$  is relevant for quantifying the parallel convergence rate (i.e. the convergence rate when working on a parallel computer, with parallel evaluation of the offspring).  $\text{NCR}_\varepsilon$  is relevant for quantifying the sequential convergence rate, i.e. when individual are evaluated sequentially.

Theorem 1 can be reformulated with these unified definitions of convergence rates as follows. Consider a  $(\mu \dagger \lambda)$ -ES satisfying the hypothesis of Theorem 1. Let  $\alpha(\varepsilon) = 1/(1 - 1/N(\varepsilon))$ . Then

$$\text{CR}_\varepsilon \leq \frac{\log(K)\alpha(\varepsilon)}{d} \quad \text{and} \quad \text{NCR}_\varepsilon \leq \frac{\log(K)\alpha(\varepsilon)}{d\lambda}. \quad (1)$$

## 4 Sauer's lemma and VC-dimension

Teytaud and Gelly [17] applied the bounds obtained in Section 3 in the following way: the number of subsets of size  $\mu$  of a set of  $\lambda$  points, is at most  $\binom{\lambda}{\mu} \leq \binom{\lambda}{\lfloor \lambda/2 \rfloor} \leq (2^\lambda / \sqrt{2\pi\lambda})$  – see e.g. [7, p587] or [9] for these inequalities. This surely holds, but it is a worst case on possible selections: if the fitness functions are “nice”, many of these subsets cannot be realized. This is precisely quantified by Sauer's lemma in the theory of VC-dimension. In this section, we show how this allows to obtain more precise lower bounds on the convergence rate of  $(\mu \dagger \lambda)$ -ES.

Given a function  $f$  defined over  $D$  and  $r > 0$ , let  $O_{f,r} = \{x \in D \mid f(x) < r\}$ . We define the *level sets*  $L_{\mathcal{F}}$  of a set  $\mathcal{F}$  of functions defined over the domain  $D$  as

$$L_{\mathcal{F}} = \{O_{f,r} \mid f \in \mathcal{F}, r > 0\}.$$

We now briefly recall the definition of VC-dimension and Sauer's lemma [18, 16] – our presentation is based on [13]. 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  $\mathcal{S}|_B = \{S \cap B \mid S \in \mathcal{S}\}$ . The VC-dimension of the set system  $\mathcal{S}$  defined over  $A$  is defined as  $\sup\{|B| \mid \mathcal{S}|_B = 2^B\}$  where  $2^B$  denotes the powerset 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  $\pi_{\mathcal{S}}(m) = \max\{|\mathcal{S}|_B| \mid B \subseteq A, |B| = m\}$ ; 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 Sauer's lemma which gives an upper bound on  $\pi_{\mathcal{S}}$  in terms of the VC-dimension of  $\mathcal{S}$ .

**Lemma 1 (Sauer’s lemma).** *For any set system  $\mathcal{S}$  of VC-dimension  $d$ , then for all integer  $m$ , it holds that  $\pi_{\mathcal{S}}(m) \leq \sum_{i=0}^d \binom{m}{i}$ .*

At last, let us recall the following classical bounds [13]:

$$\sum_{i=0}^d \binom{m}{i} \leq \min\left\{\left(\frac{em}{d}\right)^d, \lambda^d, 2^m\right\}. \quad (2)$$

Note that the trivial bound  $2^m$  is tight when  $m \leq d$ . The interesting case happens when  $m$  is large with respect to the VC-dimension  $d$ : the bound becomes polynomial in  $m$  in this case. This element is central for the difference between the results in this paper and results in [17].

#### 4.1 Non-elitist strategies

We first give an upper bound on the branching factor of a SB- $(\mu, \lambda)$ -ES in term of the VC-dimension of level sets.

**Lemma 2.** *Consider a SB- $(\mu, \lambda)$ -ES as described in Algorithm 1. Let  $V$  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  $K \leq \lambda^V$ .*

*Proof.* Given a set of  $\lambda$  points  $P = \{x_1, \dots, x_\lambda\}$  in the domain  $D$ , and  $f \in \mathcal{F}$ , let us define  $M_f(P)$  to be the subset  $Q$  of size  $\mu$  of  $P$  corresponding to the  $\mu$  points of  $P$  with lowest fitness values with respect to  $f$ . Note that the branching factor satisfies

$$K \leq \max_{P \subset D, |P|=\lambda} |\{M_f(P) \mid f \in \mathcal{F}\}|.$$

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

$$|\{M_f(P) \mid f \in \mathcal{F}\}| \leq \pi_{L_{\mathcal{F}}}(\lambda).$$

If the VC-dimension of  $L_{\mathcal{F}}$  is at most  $V$ , it follows from Sauer’s lemma and the bound given in Equation 2 that  $\pi_{L_{\mathcal{F}}}(\lambda) \leq \lambda^V$ . Thus  $K \leq \lambda^V$ .  $\square$

**Theorem 2 (SB- $(\mu, \lambda)$ -ES).** *Consider a SB- $(\mu, \lambda)$ -ES (Algorithm 1) in a domain  $D \subset \mathbb{R}^d$ , such that  $D = \{f^* \mid f \in \mathcal{F}\}$ . Let  $V$  be the VC-dimension of the level sets of  $\mathcal{F}$ . The normalized convergence rate of this algorithm satisfies  $\text{NCR}_\varepsilon \leq V \log(\lambda) \alpha(\varepsilon) / (d\lambda)$ , where  $\alpha(\varepsilon) = 1 / (1 - 1/N(\varepsilon))$ .*

*Proof.* The result easily follows from the upper bound on the branching factor given in Lemma 2, and from Theorem 1 as stated in Equation 1.  $\square$

We next give a couple of applications based on the VC-dimension of classical set systems [7].

*Corollaries in the continuous domain.* Let’s consider the case of the domain  $D = [0, 1]^d \subset \mathbb{R}^d$ . Then,  $\alpha(\varepsilon) \rightarrow 1$  as  $\varepsilon \rightarrow 0$ . The sphere functions is the the set of fitness functions  $\mathcal{F} = \{f_c \mid c \in D\}$  where  $f_c(x) = ((x_1 - c_1)^2 + \dots + (x_d - c_d)^2)^{1/2}$ ; the system of level sets has VC-dimension  $V = d + 1$  in this case. Quadratic functions with positive Hessian, where level sets are ellipsoids, have VC-dimension  $V \leq d + d(d + 1)/2$ . Functions with hyperrectangles as level sets satisfy  $V \leq 2d$ . In all these cases, algorithms of type SB- $(\mu, \lambda)$ -ES have convergence rate  $\text{CR}_\varepsilon = O(V \log(\lambda)/d)$ .

*Corollaries for bitstrings.* Let’s now consider the discrete case  $D = \{0, 1\}^d$ . For  $\varepsilon$  sufficiently small, the balls are singletons; it follows that  $\alpha(0) = 1 / (1 - 1/2^d)$ . First, let’s consider the ONEMAX function ( $x \mapsto \sum_{i=1}^d x_i$ ) and its symetries: the set of functions is

$$\{x \mapsto \sum_{i \in [1, d]} |x_i - \eta_i|; \eta \in \{0, 1\}^d\}.$$

Then, the VC-dimension of level sets (balls) satisfies  $V \leq d + 1$ ; therefore, the convergence rate  $\text{CR}_\varepsilon$  is at best, for  $\varepsilon$  sufficiently small,  $\alpha(0) \cdot (d + 1) \cdot \log(\lambda)/d$ . A second classical set of fitness functions on the same domain  $\{0, 1\}^d$  is linear functions. Since they are obtained by a restriction of linear functions in  $\mathbb{R}^d$ , their VC-dimension satisfies  $V \leq d + 1$ ; hence, the convergence rate  $\text{CR}_\varepsilon$  is at best, for  $\varepsilon$  sufficiently small,  $\alpha(0) \cdot (d + 1) \cdot \log(\lambda)/d$ . In particular, this includes the special case of the set of permutations of  $[[1, d]]$ . In the same way, sphere functions or quadratic positive definite functions in the discrete case have a VC-dimension bounded from above by the VC-dimension in the continuous case, leading to similar results.

## 4.2 Non-elitist strategies with full ranking

This subsection is organized as follows:

- First we will see to which extent lower bounds obtained for SB- $(\lambda, \mu)$ -ES are modified when we use the full ranking information and not only selection information (i.e. we move from Algorithm 1 to Algorithm 2);
- However, we show that the speed-up as a function of  $\lambda$  is at most logarithmic for  $\lambda$  large in the special case of the sphere function;
- At last, for the sphere function again, we show that for  $\lambda = 2d$ , some significant improvement holds:  $\text{CR}_\varepsilon$  moves from  $\Theta(1/d)$  when  $\lambda = 1$  to  $\Theta(1)$  for  $\lambda = 2d$ . This is the tightness of Theorem 3 for  $\lambda = 2d$  as discussed below.

**Keeping the full ranking information.** Consider the case of Algorithm 2 instead of Algorithm 1; we have a wider family of algorithms as we can use all the ranking information. There are evolutionary algorithms which use the full ranking information of the selected points and not only selection; for example, roulette-wheel with rank-based fitness assignment (stochastic sampling [4], rank-based fitness assignment [3, 19]), weighted recombination [11, 1] or BREDA [10]. In this case, 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

$$\text{NCR}_\varepsilon \leq (V \log \lambda + \mu \log \mu) \alpha(\varepsilon)/(d\lambda).$$

However, we can say better in the case where  $\mu$  is large with respect to the VC-dimension  $V$  of the level sets of the fitness functions.

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

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

*is at most  $2^{4Vn}$ .*

*Proof.* Let us denote by  $\gamma(n)$  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  $2^{p-1} < n \leq 2^p$ . Let  $\tilde{n} = 2^p$ . A possible order on  $\tilde{n}$  points is completely determined by the  $\tilde{n}/2$  points with smallest values with respect to  $f$ , multiplied by the number of possible orders on two sets of  $\tilde{n}/2$  points. Thus  $\gamma(\tilde{n}) \leq \tilde{n}^V \gamma(\tilde{n}/2)^2$ . Iterating this  $p$  times until we get sets of size 2, we obtain:

$$\gamma(\tilde{n}) \leq \tilde{n}^V \left(\frac{\tilde{n}}{2}\right)^{2V} \dots \left(\frac{\tilde{n}}{2^{p-1}}\right)^{2^{p-1}V}.$$

It follows that

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

Of course  $\tilde{n}/2^i = 2^{p-i}$ . Moreover,

$$\sum_{i=0}^{p-1} 2^i(p-i) = 2^{p+1} - p - 2 \leq 2\tilde{n} \leq 4n.$$

This gives  $\log \gamma(n) \leq 4Vn$ . □

**Theorem 3 (Full ranking  $(\mu, \lambda)$ -ES).** *Consider a  $(\mu, \lambda)$ -ES (Algorithm 2) in a domain  $D \subset \mathbb{R}^d$ , such that  $D = \{f^* \mid f \in \mathcal{F}\}$ . Let  $V$  be the VC-dimension of the level sets of  $\mathcal{F}$ . Then the normalized convergence rate of this algorithm satisfies*

$$\text{NCR}_\varepsilon \leq V(\log \lambda + 4\mu) \alpha(\varepsilon)/(d\lambda)$$

where  $\alpha(\varepsilon) = 1/(1 - 1/N(\varepsilon))$ .

*Proof.* The branching factor of this algorithm is bounded by  $K \leq \lambda^V \gamma(\mu)$  where  $\gamma(\mu)$  is the possible number of orders on the  $\mu$  selected points with respect to fitness values. Lemma 3 shows that  $\log \gamma(\mu) \leq 4V\mu$ . Using Equation 1 yields the desired bounds. □

Selecting  $\lambda$  individuals among  $\lambda$  individuals is not meaningless if we keep the full ranking information. Then, the bound on the convergence rate per iteration becomes  $\text{CR}_\varepsilon = O(\lambda V/d)$ . This does not forbid  $\text{NCR}_\varepsilon = \Theta(1)$ . However, we now show that this bound can be improved in the case of the sphere function. Surprisingly, we *improve* the lower bound by considering specifically the sphere function; a similar phenomenon holds in [17], in which the authors show that some fast convergence rates are possible with specific unnatural fitness functions, *easier than the sphere function*.

**The case of the sphere function: complexity bounds for  $\lambda$  large.** For the sphere function and the Euclidean norm, we next give an upper bound on the convergence rate of a selection-based algorithm using full ranking.

**Proposition 1.** *Consider a  $(\mu, \lambda)$ -ES, as in Algorithm 2, optimizing the sphere function in a domain  $D \subset \mathbb{R}^d$ . Then  $\text{NCR}_\varepsilon \leq 2(\log \lambda)/\lambda \cdot \alpha(\varepsilon)$  where  $\alpha(\varepsilon) = 1/(1 - 1/N(\varepsilon))$ .*

*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.  $H_{p,q} = \{x \in \mathbb{R}^d \mid \|x - p\| = \|x - q\|\}$ .

At each iteration of the algorithm, an offspring of  $\lambda$  points  $\{x_1, \dots, x_\lambda\}$  is generated and the algorithm receives the sequence of indices of the  $\mu$  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  $1 \leq i < j \leq \lambda$ ; this sign is positive or negative since we assumed equality never occurs. The number of possible sign vectors  $s = (s_{i,j})_{1 \leq i < j \leq \lambda}$  is exactly the number of cells of the arrangement of hyperplanes  $\{H_{x_i, x_j} \mid 1 \leq i < j \leq \lambda\}$  in  $\mathbb{R}^d$ . But it is known that  $n$  hyperplanes in  $\mathbb{R}^d$  define at most  $n^d$  cells – see chapter 6 of [13]. Since there are  $\binom{\lambda}{2} \leq \lambda^2$  hyperplanes here, we obtain  $K \leq \lambda^{2d}$ . Applying Equation 1 yields the announced bounds on  $\text{CR}_\varepsilon$  and  $\text{NCR}_\varepsilon$ . □

When  $\varepsilon$  tends towards 0 and as  $N(\varepsilon) \rightarrow \infty$ , this gives  $\text{CR}_\varepsilon \leq 2 \log \lambda$ ; this shows that the upper bound given by Theorem 3 cannot be reached in this case.

Although Proposition 1 concerns the case of the sphere function, it can be applied to a more general setting. Indeed, it applies to systems where the number of sign conditions, i.e. the number of possible sign vectors  $(\text{sign}(f(x_i) - f(x_j)))_{1 \leq i < j \leq n}$  realized by any set of points  $x_1, \dots, x_n \in D$  and any fitness function  $f \in \mathcal{F}$ , can be efficiently bounded. This is in particular the case for polynomials of bounded degree – see [?] and chapter 10 of [13].



**The case of the sphere function: fast convergence rate with  $\lambda = 2d$ .** We point out here that for the specific case of the sphere function, a convergence rate  $\text{CR}_\varepsilon = \Theta(1)$  can be reached with  $\lambda = 2d$  in the domain  $[0, 1]^d$  by some algorithm of type full ranking  $(\mu, \lambda)$ -ES; this shows tightness of Theorem 3 for  $\mu = \lambda = 2d$ , within logarithmic factors of  $\lambda$ .

This convergence rate is easily obtained with Algorithm 3, which works as follows. It splits  $[0, 1]^d$  in the  $2^d$  cells delimited by the  $d$  hyperplanes of equations  $x_i = 1/2$ ; the full ranking of the  $2d$  points  $\{x_i = \eta \mid 1 \leq i \leq d, \eta \in \{0, 1\}\}$  allows to decide in which of these cells lies the optimum; then the algorithm proceeds recursively. This is quite similar to the Hooke and Jeeves algorithm [?].

---

**Algorithm 3** An example of algorithm for which the convergence rate is  $\text{CR}_\varepsilon = \Theta(1)$  with  $\lambda = 2d$  on the sphere function with optimum in  $[0, 1]^d$ . In this algorithm  $e_i$  denotes the vector  $(0, \dots, 0, 1, 0, \dots, 0)$  with a unique 1 in position  $i$ .

---

$x = 0.5, \sigma = 0.5.$

**while true do**

    Generate  $\lambda = 2d$  distinct points equal to  $x \pm \sigma e_i$ .

    With the ranking information, decide in which octant of  $x + [-\sigma, \sigma]^d$  is the optimum. Move  $x$  to the center of this octant.

    Set  $\sigma \leftarrow \sigma/2$ .

**end while**

---

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

$$\text{For } \lambda = 2d : \text{CR}_\varepsilon = \frac{\log N(\varepsilon)}{d n_{\varepsilon, \frac{1}{2}}} = \Theta(1). \quad (3)$$

### 4.3 Elitist strategies

For the sake of completeness, we state the analog of previous results in the elitist case. The same technique as in Theorem 2 applies to the case of SB- $(\mu + \lambda)$ -ES.

**Corollary 1 (SB- $(\mu + \lambda)$ -ES).** *Consider a SB- $(\mu + \lambda)$ -ES as Algorithm 1 in a domain  $D \subset \mathbb{R}^d$ , such that  $D = \{f^* \mid f \in \mathcal{F}\}$ . Let  $V$  be the VC-dimension of the level sets of  $\mathcal{F}$ . The normalized convergence rate of this algorithm satisfies  $\text{NCR}_\varepsilon \leq V \log(\lambda + \mu) \alpha(\varepsilon)/(d\lambda)$ , where  $\alpha(\varepsilon) = 1/(1 - 1/N(\varepsilon))$ .*

*Proof.* An analog of Lemma 2 yields  $K \leq (\lambda + \mu)^V$  in this setting, since the selection is performed among a set of  $\lambda + \mu$  points instead of  $\lambda$  points at each iteration. Then the result follows from Equation 1.  $\square$

When  $\mu$  is very large this leads to  $\text{CR}_\varepsilon = O(V \log(\mu + \lambda))$ ; thanks to the use of Sauer's lemma, this improves the  $\lambda \log(\mu + \lambda)$  bound from [17] by removing the linear dependency in  $\lambda$ . Of course the last theorem is pertinent when  $\mu + \lambda \geq V$ ; otherwise, one should apply the bound  $K \leq \binom{\lambda + \mu}{\mu}$  used in [17]. One can also obtain an analog of Theorem 3 in the elitist setting with full ranking:

**Corollary 2 (Full ranking  $(\mu + \lambda)$ -ES).** *Consider a  $(\mu + \lambda)$ -ES (Algorithm 2) in a domain  $D \subset \mathbb{R}^d$ , such that  $D = \{f^* \mid f \in \mathcal{F}\}$ . Let  $V$  be the VC-dimension of the level sets of  $\mathcal{F}$ . Then the normalized convergence rate of this algorithm satisfies*

$$\text{NCR}_\varepsilon \leq V (\log(\lambda + \mu) + 4\mu) \alpha(\varepsilon)/(d\lambda)$$

where  $\alpha(\varepsilon) = 1/(1 - 1/N(\varepsilon))$ .  $\square$

A possible variant of full ranking  $(\mu + \lambda)$ -ES is obtained with  $\mu = \infty$ . In this type of algorithms, an offspring of  $\lambda$  points is generated at each step, and the full ranking of all points generated so far is given to the algorithm.

**Proposition 2 (Full ranking  $(\infty + \lambda)$ -ES).** *Under the hypothesis of Corollary 2, the normalized convergence rate of a  $(\infty + \lambda)$ -ES satisfies  $\text{NCR}_\varepsilon \leq \frac{4V}{d} \cdot \left(1 - \frac{1}{\log N(\varepsilon)}\right)^{-1}$ .*

*Proof.* The number of leaves of a computation tree after  $n$  steps is bounded from above by the number  $L$  of orders of  $\lambda n$  points with respect to some fitness function  $f \in \mathcal{F}$ . By Lemma 3, it holds that  $L \leq 2^{4V\lambda n}$ . It follows that  $2^{4V\lambda n_{\varepsilon, \delta}} \geq (1 - \delta)N(\varepsilon)$ . This gives  $n_{\varepsilon, 1/2}\lambda \geq (\log N(\varepsilon) - 1)/4V$ . Thus  $\text{NCR}_\varepsilon \leq \frac{4V}{d} \cdot \frac{\log N(\varepsilon)}{\log N(\varepsilon) - 1}$ .  $\square$

In the special case of sphere function, an analysis similar to the one of Proposition 1 can be done here; this allows to obtain the improved bound  $\text{NCR}_\varepsilon \leq \frac{\log N(\varepsilon)}{(\frac{1}{2}N(\varepsilon))^{1/2d}}$ .

## 5 Summary of results

Let's apply the results obtained in the previous section to the simple framework of the domain  $D = [0, 1]^d$  with the Euclidean norm. Lower bounds obtained in this setting are summarized in Figure 1. Higher values mean better possible convergence rates. However, it is not known when these convergence rates can be achieved. Indeed, results marked with (\*) in Figure 1 are improved in the special case of the sphere function in Section 4.2: this shows that at least in this case, general bounds on convergence rate derived from VC-dimension are not tight. Discussion of these results follows.

	SB- $(\mu, \lambda)$ -ES	SB- $(\mu + \lambda)$ -ES	Full ranking $(\mu, \lambda)$ -ES	Full ranking $(\mu + \lambda)$ -ES	Full ranking $(\infty + \lambda)$ -ES
CR	$V \log(\lambda)/d$	$V \log(\mu + \lambda)/d$	$V(\log(\lambda) + \mu)/d$ (*)	$V(\log(\lambda + \mu) + \mu)/d$	$4V\lambda/d$
NCR	$V \log(\lambda)/(d\lambda)$	$V \log(\mu + \lambda)/(d\lambda)$	$V(\log(\lambda) + \mu)/(d\lambda)$ (*)	$V(\log(\lambda + \mu) + \mu)/(d\lambda)$	$4V/d$
Shown in	Theorem 2	Corollary 1	Theorem 3	Corollary 2	Proposition 2

**Fig. 1.** Upper bound on the (normalized) convergence rate in the case of Euclidean norm in the domain  $[0, 1]^d$ , when the level sets of fitness functions have VC-dimension  $V$ .

*Asymptotic speed-up in the case of selection only, non-elitist.* In the case of evolution strategies based on selection only (algorithms of type SB- $(\mu, \lambda)$ -ES), the linear speed-up of selection-based evolution strategies shown in [5] cannot be obtained for  $\lambda$  large enough. Asymptotically, the speed-up obtained with such an algorithm is at most logarithmic as shown in Theorem 2.

*Selection based algorithms vs. full ranking.* When moving from selection based algorithms of type SB- $(\mu, \lambda)$ -ES to full ranking  $(\mu, \lambda)$ -ES, upper 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  $\lambda$ .

However, we know from Proposition 1 that the speed-up is at most logarithmic for a full ranking  $(\mu, \lambda)$ -ES in the special case of sphere function – see also the discussion following Proposition 1. This raises the following question: for which kind of fitness functions is it interesting to keep the full ranking information?

On the other hand, in the special case of the sphere function, we have seen that a linear speed-up is can be achieved for  $\lambda$  linear in the dimension  $d$  in the full ranking case. This may suggest that for parallel evolution strategies evaluating  $\lambda$  points at once, using a number of processors linear in the dimension may be a reasonable choice.

A related intriguing question is the convergence rate that can be reached for selection based algorithms (i.e. without keeping the full ranking information) for the sphere function. In particular, in the case  $\lambda = \Theta(d)$  in dimension  $d$ , is it possible to achieve a convergence rate  $\text{CR}_\varepsilon = \Theta(1)$ , as in Equation 3? To the best of our knowledge, this is an open problem.

*Elitist vs. non-elitist.* Bounds obtained do not show a strong improvement between elitist and non-elitist strategies: bounds on the convergence rate are of the same order, in both selection based and full ranking settings. This is simply explained by the fact that any algorithm of type SB- $(\mu + \lambda)$ -ES can be simulated by an algorithm of type SB- $(\mu', \lambda')$ -ES with  $\mu' = \mu$  and  $\lambda' = \mu + \lambda$  (and in the same way, a full ranking  $(\mu + \lambda)$ -ES can be simulated by a full ranking  $(\mu', \lambda')$ -ES).

**Acknowledgements** Many thanks to Anne Auger, Nikolaus Hansen and Fabien Teytaud for constructive talks. 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  $(1, \lambda)$ -SA-ES using the theory of  $\varphi$ -irreducible markov chains. *Theoretical Computer Science*, 334:35–69, 2005.
3. 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*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.
4. 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.
5. 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.
6. Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies: a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.
7. L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic Theory of Pattern Recognition*. Springer, 1997.
8. 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.
9. W. Feller. *An introduction to Probability Theory and its Applications*. Wiley, 1968.
10. S. Gelly, S. Ruetten, 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):26p, 2007.
11. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 11(1), 2003.
12. R. Hooke and T. A. Jeeves. "direct search" solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.
13. Jiří Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer, 2002.
14. I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.
15. 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.
16. G. Rudolph. Convergence rates of evolutionary algorithms for a class of convex objective functions. *Control and Cybernetics*, 26(3):375–390, 1997.
17. N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory (A)*, 13:145–147, 1972.
18. O. Teytaud and S. Gelly. General lower bounds for evolutionary algorithms. In *proceedings of PPSN*, pages 21–31, 2006.
19. 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.
20. 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*, San Mateo, CA, 1989. Morgan Kaufman.