Inside GA Dynamics: Ground Basis for Comparison

Proceedings of the 4^{th} Conference on Parallel Problems Solving from Nature. 1998. Pages 57-66.

Leila Kallel

CMAP – UMR CNRS 7641
Ecole Polytechnique
Palaiseau 91128, France
kallel@cmapx.polytechnique.fr

Revised version, March 16, 1999

Abstract. This paper reviews some of the commonly used methods to compare GA performance on different problems or to compare the adequation of GA parameters and operators. This points out the scaling issue when differently distributed fitness functions are compared. *Hamming fitness functions* are then proposed as a basis to relax the dependence of GA-performance-measures on fitness scale or distribution, and to account for any GA parameters bias. Different measures are also proposed, distinguishing convergence time from GA radial and effective trajectory in Hamming space.

1 INTRODUCTION

One of the major challenges in the field of Genetic Algorithms (GA) is to characterize which fitness functions are easy for the GA and which are not. Whereas the answer is straightforward in the case of the Hill Climber algorithm, given by the presence of local optima, no rigorous definition of the concept of difficulty is available in the framework of GAs. Yet, many papers implicitly use the notion of difficulty in many cases: When two instances of algorithm parameters are compared on some problem; when two different problems are compared for the same algorithm; or when tools are developed (and validated) in order to measure problem difficulty. Examples of such tools are fitness distance correlation [14], correlation length, operator correlation [22] , epistasis [6], schema variance [25], hyperplane ranking [5] ...

Early attempts to characterize difficulty [11, 10, 9] propose criteria based on: Isolation (needle in a haystack), Deception (schema analysis) and Multimodality. It seems clear that these criterion certainly contribute to problem difficulty for a GA. However, there exist examples showing that the latter two criterion (deception and multimodality) are neither necessary nor sufficient to make a problem difficult: Vose and Wright [29] present the extreme case of a fully non deceptive function that is difficult for a GA because of an exponential number (in terms of string length l) of stable suboptimal attractors. Wilson [31] proposes a deceptive function, yet straightforward to optimize for a GA. Horn and Goldberg [12] construct an easy maximally multimodal function. They also stress the fact that modality and schema-GA-easy analysis are not at all sufficient to describe the behavior of the GA. Finally, longpath problems [13] have been designed to show that unimodal functions can be difficult for a simple 1-bit-flip hill climber because of an exponential path length (and [15] proves that an instance of Rudolph's *long k-paths* is a longpath for $(\mu + \lambda)$-ES, and proposes a construction of candidate longpaths for GAs).
Other works unformally use the word 'difficult problem for GA' when the fitness landscape presents a large number of local optima with large basins of attraction and high barriers between them. This is the case of the NK-landscapes when K value increases. This problem is widely studied in [20], where problem difficulty is related to the repartition of local optima around the global optimum, indexed by their fitness and the size of their basin of attraction. This method, however, requires the knowledge of the whole fitness landscape and is rather time consuming.

On the other hand, many papers [7, 4] stress the nonsense of speaking of problem difficulty without considering the parameters of the algorithm at hand. Examples are given where the same landscape can be difficult or easy (in convergence speed) to optimize according to the parameters and operators of the algorithm. Along that same line, some tools have been proposed to *a priori* estimate the efficiency of different crossover operators ([18] reviews and compares these tools). As in the case of landscape analysis measures, these tools have been designed to capture key properties that may predict GA-behavior.

This paper addresses the key issue of *a posteriori* comparison of GA results: Indeed, before being able to *a priori* measure GA-hardness, one should be able to estimate *a posteriori* the notion of difficulty. But, given GA results on two different problems, there exist no rigorous comparison in order to decide which problem is the more difficult one.

After recalling some commonly used methods for comparing GA results, section 1.1 motivates the need for a basis to compare different problems that have different fitness distributions and different global optimum genotypes.
The *Hamming fitness* is then proposed as a relative basis to scale GA results (convergence time and trajectory), and to take into account eventual parameters biases (such as over-fitting initialization that generates close to optimal individuals).

1.1 Measures of GA performance : State of the art

Much attention has been paid in the GA literature to understand, to characterize or to predict GA behavior on a given fitness function. Such works often use some implicit classification of GA results. The most simple and common way to compare GA results on different problems or with different parameter settings is to run numerous experiments for each setting, and to compare GA results with some user-defined measure. Such measure can be the average on-line and off-line fitnesses [28], or the ratio of best fitnesses respectively found by GA and pure random search, after the same number of evaluations [26]. Many papers report the fraction of runs that find the global optimum. This measure is alternatively named *success rate* (1989, Tanese's doctoral dissertation) and *robustness of evolution* [3], with the difference that the latter paper considers the fraction of runs that find at least one –user-defined– *acceptable solution*, rather than the global optimum. The same paper also defines the *speed of evolution* as the number of generations necessary to discover an *acceptable solution*.
This trial-and-error experimental method is unfortunately time consuming. Nevertheless, the lack of efficient and robust measures of difficulty and/or adequation of settings for a given problem has made this experimental comparative method very popular since the early days of GAs.

As early as 1991, some attempt was made to formalize the properties that all comparison measures should follow [8] *(1) they should be normalized so that performance can be compared on different problems and with different optimizers, (2) they should reflect problem difficulty, and (3) they should indicate how far the optimizer got, not just whether or not it reached the global optimum.* However, a different point of view is encountered in [19] (1996) concerning SEARCH, a particular optimization algorithm: GA performance is not only related to some measure, but also to the way the measure evolves with some parameter (problem size, solution quality or success probability requirement). This is not the point of view considered here, and we stick to the requirements of [8] cited above.

Within the experimental comparison methods recalled above, one can distinguish two kinds of behavioral measures: those based on fitness values, and all others. Unfortunately, when different problems are compared, respective fitness distributions can be completely different. Hence, the ratio, or the amount of fitness improvement is completely unreliable. On the other hand, measures like the probability of success and the convergence time of successful runs have serious drawbacks. They do not give information about what happened during the run, and they do not take into account possible biases due to initialization or operators over-specialization to the problem at hand. For example, a given initialization can seriously bias the optimization process, by generating individuals very close in distance to the optimum of one of the compared problems (a caricatural example is given in section **??** with function g_1').

The goal of this paper is to design some comparison method that allows us to overcome these drawbacks. The section organizes as follows.

Section 2 specifies the context and goal of the paper. Section 3 presents the comparison approach: using *Hamming fitness* as a basis to scale the *behavioral measures* of GA runs when different problems, or different GA instances are involved. Examples of *behavioral measures* are proposed in Section 4 distinguishing time and trajectory. Finally, Section 5 instantiates the method on some well known problems of GA literature. Special attention is paid to the comparison of GA and other ESs dynamics on the difficult Baluja binary F1 problem.

2 MOTIVATION

Before presenting the comparison method with *Hamming fitness* basis, this section discusses two different points of view on difficulty.

Characterizing GA performance: Many papers [7, 4, 24] stress the nonsense of speaking of problem difficulty without considering all the parameters of the algorithm at hand: [4] presents two unimodal functions on the binary space. Depending on whether elitism is used or not, the same sampling technique (ex. steepest ascent hill climber) yields an exponential difference in convergence time. Authors explain that the potentialities of an operator are not necessarily exploited by the algorithm. Such differences are unmeasurable, of any landscape based indicator. Hence, considering the on-line behavior, relaxes this problem of the unused potentialities of the operator graph.

In that line, we proposes to consider the notion of difficulty as related to a given run of some GA instance. As a first step to refined the notion of GA difficulty, we propose a method to compare GA behaviors when applied to different fitness functions.

The aim of this paper is not to define an order relation in the set of possible GA behaviors, as the notion of what is important may vary widely across problem domains. Rather, it simply proposes to extract *scalable* information that may be used later on (in a manner dependent on the user's prerequisites), to construct a total or a partial GA-performance relation that suites a particular problem domain or user's prerequisites.

3 COMPARISON METHOD OF GA BEHAVIORS

Since GA behavior on *distance fitness* functions (the fitness is the distance [1] to optimum), is well understood, we propose to use it as an absolute basis to *scale* the behavioral measures, following the question: *What would this same algorithm have done –starting from the same population– if for each x the fitness function were simply the distance to optimum of f (or to the best individual found by the algorithm).*

More precisely, the comparison method consists in performing the following steps for fitness f (resp. g), then in comparing $\rho_m(f)$ and $\rho_m(g)$, where m is a behavioral measure of the algorithm as illustrated in Figure 4.2:

1. `Run GA with fitness` f $\longrightarrow$ `compute` $m(f)$
   Note s be the best found by the GA.

2. `Run the same`[2] `GA with fitness` $\mathcal{H}_s$[3] $\longrightarrow$ `compute` $m(\mathcal{H}_s)$

3. `Compute ratios` $\rho_m = m(f)/m(\mathcal{H}_s)$

$\longrightarrow$ The *Hamming fitness* are proposed as an absolute basis to **scale GA behavioral measures** (for example convergence time or trajectory length), and to take into account eventual parameters biases (such as over-fitting initialization that generates close to optimal individuals).

---

[1] We suppose that the genotypic distance is implicitly *related* to GA operators, as for example the *Hamming* distance is related to the *1-bit-flip* mutation operator. The relevance of a distance in relation to genetic operators is addressed in the third chapter.

[2] GA with the same initial population and parameters as for f in step 1

[3] $\mathcal{H}_s(x) = l - d(x, s)$ is the Hamming fitness to s

4 BEHAVIORAL MEASURES OF GAs

In the following, we present some measures that characterize on-line GA behavior (during a given run) when maximizing a fitness function f with a unique global optimum on the binary space.

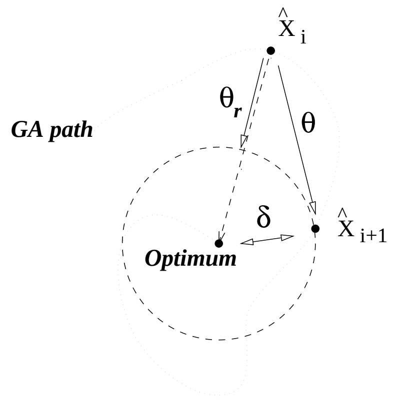
4.1 Definition of GA trajectory

GA *behavior* includes both temporal and spatial issues. If time is rather straightforward to define, it is not the case with spatial trajectory.
The effective trajectory of an GA is the succession of populations. To measure the length of this trajectory, one can use a distance between two populations that are considered as sets of elements of E. But in the context of optimization, we may be interested in the trajectory of the upper tail of the successive populations. In the following **GA trajectory will be reduced to the fittest, so far, individual known to the GA at each generation**. But the same method can be extended to different definitions of the GA's trajectory, by simply replacing the Hamming distance between individuals with the adequate distance between populations (for example the *Hausdorff* distance).

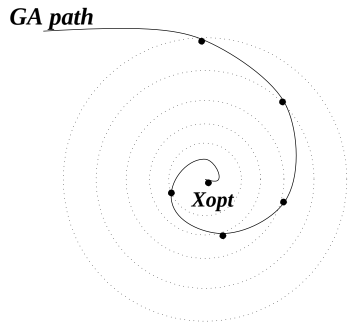
4.2 Behavioral measures

We define a number of *behavioral measures* (independent from fitness distribution) to characterize GA behavior and results on a given fitness function: In addition to convergence time, Figure **??** illustrates some geometrical quantities derived from the trajectory of the algorithm. It is not easy to compare two sets of such behavioral measures, as they are very sensitive to initialization bias (initialization with close to optimal individuals) and on the choice of the genotypic distance.

Let $\hat{X}_i$ be the best-so-far individual at generation i, N_c the number of generations to reach $\hat{X}_i$, and d_n the Hamming distance normalized to string length. The behavioral measures presented below, characterize on-line GA behavior (of a given run) to reach $\hat{X}_{N_c}$.



Behavioral measures of GA trajectory.



Typical GA path with a *Hamming* fitness: θ is minimal, not θ_r.

- **Trajectory length:** $\theta = \sum_{1,N_c-1} d_n(\hat{X}_i, \hat{X}_{i+1})$
- **Radial progress:** $\theta_\mathbf{r} = \sum_{1,N_c-1} |d_n(\hat{X}_i, \hat{X}_{N_c}) - d_n(\hat{X}_{i+1}, \hat{X}_{N_c})|$
- **Distance to best** : $\delta = \dfrac{\sum_{1,N_c} d_n(\hat{X}_i, \hat{X}_{N_c})}{N_c}$
- **Convergence time:** $\mathbf{N_c} =$ Number of generations

Note that θ and θ_r are not necessarily correlated:

θ_r can be minimal, with different θ values. This is the case for *Hamming fitness* functions, where experiments (of section 5) show that θ typically equals 5 to 10 times θ_r. Note that θ_r is minimal and equals $d_n(\hat{X}_1, \hat{X}_{N_c})$, if the GA comes closer to $\hat{X}_{N_c}$ at each step (Fig. 4.2-b).

On the other hand, the unrealistic case of minimal θ necessarily implies a minimal θ_r: θ is minimum and equals $d_n\hat{X}_1, \hat{X}_{N_c})$, if each time an allele of the optimum is found, it remains fixed until the end of the run. Hence, there are no more than l improvement steps.

5 EXPERIMENTS

To illustrate the approach, we propose to focus on comparing the same function with different instances of GA. In the last experiment, the difficult Baluja F1 problem is studied for different algorithms.
Otherwise mentioned, a generational GA with a population size of 100, linear ranking selection, crossover rate 0.6 and $1/l$ mutation at rate [4] per individual 0.4, is used. Stopping criterion is 1000 generations with no improvement. Results are averaged over 10 runs.
The definition of the problems used below are recalled in the appendix A.

• Longpath: large θ_r

As expected, longpath problem [13] with $1/l$-bit flip mutation (Table 1) shows a particularly high Radial trajectory ($\rho_{\theta_r} = 7$): the GA -partially following the path- makes big movements getting closer then further to the optimum. These movements are responsible for the increase in average distance to optimum ($\rho_\delta = 2$), as well as for the large time ratio($\rho_{N_c} = 39$).
Longpath problem with a $2/l$-bit flip mutation shows a different behavior: 1- Trajectory measures (ρ_θ, and ρ_{θ_r}) are rather close to one: GA trajectory is similar to that obtained with a Hamming fitness to optimum. In fact, the search begins as a linear function to the first point on the path (two bits distant from optimum). 2- Average distance to optimum ($\delta = 0.66$) is even smaller than one: Once it found the first point on the path, it takes the GA a large time –relatively to the times it would take with a Hamming fitness– to find the optimum.

	N_c	δ	θ	θ_r
longpath $1/l$-mut	2466(914)	0.2(0.08)	5.7(1.6)	2.7(1)
Hamming	63(9)	0.1(0.01)	2.9(0.6)	0.3(0.02)
ρ	**39**(15)	**1.9**(0.7)	**2**(0.6)	**7.5**(2.8)
longpath $2/l$-mut	271(142)	0.07(0.01)	3.5(0.7)	0.5(0.1)
Hamming	65(11)	0.1(0.01)	2.4	0.3(0.02)
ρ	**4.3**(2.4)	**0.6** (0.13)	**1.2** (0.4)	**1.5**(0.3)

Table 1. *GA results on the 91-bits longpath problem, using mutation at $1/l$ and $2/l$-bit-flip probability. ρ is the ratio of the two upper lines (longpath and Hamming measures).*

• F1g,F3g: influence of problem size

Gray coded F1 problem (defined in A) is very sensitive to the choice of problem and population size. For example, Tab. 2 shows that convergence time ratio ρ_{N_c} increases at a super-linear rate (wrt. problem size). Yet, F1g trajectory is -in average distance- as close to optimum as the Hamming run is ($\rho_\delta = 1$). Hence, the big radial trajectory ($\rho_{\theta_r} >= 3$) is not due to big deviations far away from optimum (as for longpaths), but due to small oscillating movements.

[4] The overall mutation rate per bit is hence $0.4/l$.

On the other hand, Gray F3 problem (defined in A) is an easy problem that have a close behavior to that of Hamming functions for different population and problem sizes (Tab. 2). The difference in convergence time $\rho_{N_c} = 2$ is mainly due to small oscillating movements as witnessed by radial trajectory $\rho_{\theta_r} = 3$.

	N_c	δ	θ	θ_r
45-F3g 100-GA	**2.1** (0.3)	**1.4** (0.16)	**2.3** (1.1)	**3** (0.47)
90-F3g 100-GA	**2** (0.3)	**1.3** (0.14)	**1.6** (0.44)	**2.9** (0.29)
90-F3g 200-GA	**2** (0.35)	**1.3** (0.13)	**1.8** (0.41)	**2.9** (0.52)
45-F1g 100-GA	**16.6** (14)	**1.1** (0.29)	**1.9** (0.87)	**3.1** (0.86)
90-F1g 100-GA	**47.4** (24)	**1.1** (0.15)	**1.8** (0.3)	**4.5** (0.4)
90-F1g 200-GA	**34.8** (27)	**1.1** (0.24)	**2.2** (0.86)	**5** (1.3)

Table 2. *Influence of problem size (*-F1), with different population sizes (*-GA). Only ratios of the measures are presented. At the difference with F1g, F3g ratios scarcely vary.*

• Ugly and Royal-Road problems: initialization bias

Two initialization procedures are compared on Ugly [30] and Royal Road (RR) [23] problems (see. description in A). The second initialization – defined and studied in [17] – is biased to enhance density of ones in a bitstring. Although both problems have the same optimum (all ones string), the resulting initialization bias is not the same:
The second initialization, generates individuals closer to optimum than the first, standard initialization. For instance, initial distance to optimum drops form about 0.35 (init1) to 0.15 (init2) (values given by θ_r values of the Hamming problems in Fig. 3).
As expected, when using the second initialization, convergence is a lot faster for both Royal Road and Ugly problems (Tab. 3). But, using the same initial population, *Hamming fitness* functions also converge faster, but only ρ_{N_c} for the Royal Road problem decreases. This is not surprising since the 2nd initialization enhances the emergence of the building blocks (blocks of 1s) of the RR function. On the other hand, the Ugly landscape is invariably deceptive with both initializations, hence starting closer to the optimum scarcely changes the time ratio ($\rho_{N_c} = 7$).

	N_c	δ	θ	θ_r
Ugly init1	**7.9** (4)	**0.9** (0.2)	**1.2** (0.5)	**2** (0.5)
Ugly init2	**6.8** (4.2)	**0.9**(0.3)	**2.4** (1.1)	**3.1** (0.9)
royal init1	**16.3** (8.7)	**1** (0.2)	**0.7** (0.2)	**1.3** (0.2)
royal init2	**7.2** (7.8)	**1.2** (0.2)	**1.8** (0.7)	**2.4** (1.1)

Table 3. *Influence of initialization on 90-bits Ugly and 64-bits royal problems. Only ratios are presented. The second initialization is biased to enhance density of ones in a bitstring. Note that we easily compare different problem sizes here.*

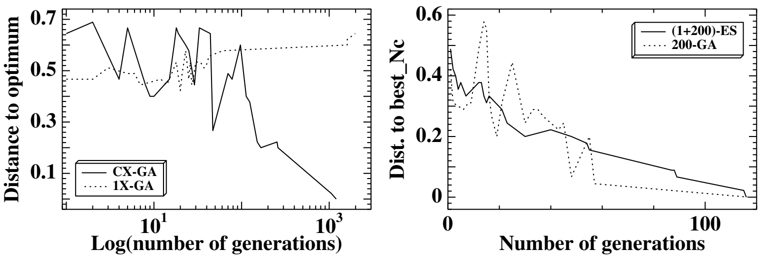
• F1b: comparison of different algorithms

Different algorithms are studied on Blauja's binary F1 problem with $l = 45$. Most algorithms (denoted F1b* in Tab. 4) fail to find the optimum. Only a GA using one-point complementary crossover (CX the crossover between a string and its bit-wise complement) finds the optimum. Note that this crossover seriously weakens the *Hamming fitness* run to optimum (807 generations!!), resulting in a small ρ_{N_c} of 2 in average.

Fig. 1-a shows GA trajectory over the course of the GA. 1C-GA gets stuck in a local optimum far away from the optimum. As witnessed by the particularly high $\rho_{\theta_r} = 7$, its trajectory makes big movements further and closer to its final best individual (Fig. 1-b). At the difference with GA, both evolution strategies have a close to Hamming trajectory distance (ratios are often close to one).
This suggests that the building blocks of the optimum are partially contained in the bitwise complements of some high fit sub-optimal points of the search space.

	N_c	δ	θ	θ_r
F1b* (1+200)-ES : ρ	**32**(45)	**0.8** (0.2)	**1.8** (0.5)	**1.2** (0.2)
F1b* (1,200)-ES : ρ	**49** (42)	**0.78** (0.3)	**1.7** (0.48)	**1.2** (0.3)
F1b* 200-GA : ρ	**4.3** (1.8)	**1.8** (0.6)	**3.1** (1.5)	**6.9** (3)
F1b 100-CX-GA	1356 (490)	0.3 (0.05)	11 (1.9)	3.1 (0.6)
Hamming	807 (320)	0.06 (0.018)	1.3 (0.6)	0.3 (0.04)
ρ	**1.9** (1.1)	**5** (1.7)	**12** (8.8)	**11** (2.7)

Table 4. *Study of Baluja's binary F1 problem with $l = 45$. F1b* means all runs failed to find the optimum. In which case, measures are calculated with respect to $\hat{X}_{N_c}$. Last case shows results with complementary crossover (CX): 8 GA runs out of 20 found the optimum, and results are averaged among successful runs only.*



(a) (b)

Fig. 1. *Binary F1 problem (a) GA trajectory with one point and complementary crossover plotted in distance to optimum: one point crossover GA gets stuck in a local optimum. (b) ES and GA trajectories plotted in distance to local optimum: ES trajectory is almost monotone decreasing.*

5.1 SUMMARY AND FURTHER DIRECTIONS

Different measures have been proposed to characterize GA behaviors, involving both time and spatial GA trajectory. Hamming fitness functions are proposed as a basis to scale the measures, and to account for GA parameters biases. For instance, the ratio of the measure obtained with the fitness f and that obtained with the *Hamming fitness* function, directly takes into account eventual parameters bias or overfitting. As the comparison is independent from fitness values, optimizing f or $g(f)$ for any monotone increasing function g, give the same results. This method can be used to compare different problems at different problem sizes and with different algorithms. For example, from time point of view, 45-bits F1g problem is as difficult as 64-bits royal problem, and (4 times) more difficult than 2-bits mutation 91-bits longpath problem.
The case of F1b problem is very instructive: it reveals a caricatural overfitting when the complementary crossover GA is used. On the other hand, mutation only evolution strategies yield a low θ_r value, and globally a similar to *Hamming fitness* behavior. Yet, this situation reflects the presence of strong local optima and certainly not problem easiness. A close look at the solutions clearly

shows they vary in fitness and in genotype from one run to the other.

The method presented in this paper can easily be generalized to different representations, simply by replacing the *Hamming fitness* function with the appropriate *distance fitness* function. Alternatively the basis can be defined with a phenotypic distance. Then, a possible comparison of results obtained with genotypic and phenotypic basis, may give insights in the dynamic of the search linked to representational issues.

Though, the choice of Hamming functions as a basis is rather arbitrary, and may be too limited when used alone. It only allows to measure the difference of GA behavior from its behavior on some fixed reference basis. A better characterization of GA behavior can be obtained from the comparison with a full sequence of well known basis functions. In this case, we get a sequence of values for each considered measure, and we could identify the basis that yields the most GA-similar behavior. The NK-Landscapes [20], for example, can provide such a sequence indexed by K values (note that Hamming functions correspond to a null K value).

A APPENDIX: TEST CASES

OneMax problem: In the well-known *OneMax* problem, the fitness function (to maximize) is simply the number of 1's in the bitstring. Hence, the fitness is equal to l minus the distance to the global optimum:

$$f(x) = l - d(x, (1, \ldots, 1))$$

The **Zeromax** problem is defined analogously by maximizing

$$f(x) = l - d(x, (0, \ldots, 0))$$

The LongPath problem: The LongPath problem [13] was purposely designed to discourage standard hill-climbers (1-bit-mutation (1+1)-ES) though being gently unimodal.

The fitness landscape is composed of two different regions; the first one is a Zeromax landscape, and culminates at $(0, \ldots, 0)$; over that smooth hill, a path of exponential length in the length of the bitstring, climbs toward the global optimum by 1-bit steps. The Hamming distance between two non-adjacent points on the path is greater than 2: local hill-climber will have to climb all the way up the path, while global methods like GAs can take short-cuts to reach the optimum quickly. The optimum is at point $(1100 \ldots 00)$.

A generalization of longpaths is presented below with some examples.

The Long k-Path problem: Rudolph [27] presents a generalization of Horn's longpath to a class of fitness functions called *long k-paths*.

The long k-path $P(l, k)$ is defined by a recursion on the length l of the strings, with $P(1, 1) = \{0, 1\}$ as the base path. Given the long k-path $P(l, k)$, $P(l + k, k)$ is constructed as follows: A sub-path S_0 of $P(l + k, k)$ is created by taking the path $P(l, k)$ and by prepending k zeros to each point on $P(l, k)$. A sub-path S_1 of $P(l + k, k)$ is created by taking $P(l, k)$ *in reverse order* and prepending k ones to each point on the reversed path. A third path B consists of $k - 1$ points created by prepending the following substrings of length k to the last point in the path $P(l, k)$: $0 \ldots 01$, $0 \ldots 011$, $\ldots$, $001 \ldots 11$ and $01 \ldots 11$. The resulting long k-path $P(l + k, k)$ on strings of length $l + k$ is then defined by the concatenation of S_0, the "bridge path" B and S_1. Horn's longpath is the long k-path for $k = 2$.

As an example, consider the construction of the path $P(7, 3)$. Starting from $P(1, 3) = \{0, 1\}$, we obtain the sequence $P(4, 3)$:

$$\underbrace{0000_1 \quad 0001_2}_{S_0} \quad \underbrace{0011_3 \quad 0111_4}_{Bridge} \quad \underbrace{1111_5 \quad 1110_6}_{S_1} \tag{A.1}$$

from which the path $P(7, 3)$ is constructed:

$$\begin{array}{lll} 0000000_1 & 0011110_7 & 1111110_9 \\ 0000001_2 & 0111110_8 & 1111111_{10} \\ 0000011_3 & & 1110111_{11} \\ 0000111_4 & & 1110011_{12} \\ 0001111_5 & & 1110001_{13} \\ 0001110_6 & & 1110000_{14} \\ & & \\ S_0 & \text{Bridge} & S_1 \end{array} \tag{A.2}$$

Gray-coded Baluja functions F1 and F3: Consider the functions of k variables $(x_1, \ldots x_k)$, with $x_i \in [-2.56, 2.56]$ [1]:

$$F_1(\mathbf{x}) = \frac{100}{10^{-5} + \sum_{i=0}^{k-1} |y_i|}, y_0 = x_0 \text{ and } y_i = x_i + y_{i-1} \text{ for } i = 1, \ldots, k-1$$

$$F_3(\mathbf{x}) = \frac{100}{10^{-5} + \Sigma_i |.024*(i+1) - x_i|}$$

They reach their maximum value of 10^7 at point $(0, \ldots 0)$. The Gray-encoded versions of F_i, with 100 variables encoded on 9 bits each are considered.

The 4 peaks problem: In the FourPeaks problem [2], the fitness is the maximum of the length of the sequence of 0's starting at the first bit position and the length of the sequence of 1's ending at the last position, plus a reward of n if both sequences are larger than a given threshold T. There are two global optima, made of a block of 0's followed by a block of 1's, of lengths T and $n-T$ or $n-T$ and T. However, there are two local optima, the all 1's and all 0's bitstrings, from which it is difficult for a GA to escape. The most difficult instance of that problem (i.e. $T = n/4$) is used here.

The Ugly problem: The Ugly problem [30] is defined from an elementary deceptive 3-bit problem (F(x) = 3 if x = 111; F(x) = 2 for x in $0\star\star$, and F(x) = 0 otherwise). The – deceptive – full problem is composed of 300 concatenated elementary deceptive problems. The maximum is the all 1's bitstring.

The Royal Road problem: The 64-bits Royal Road problem used here was conceived to study into details the combination of features most adapted to GA search (laying a *Royal Road*). A precise definition as well as an analysis of the unexpected difficulties of this problem can be found in [23].

References

1. S. Baluja. An empirical comparison of seven iterative and evolutionary function optimization heuristics. Technical Report CMU-CS-95-193, Carnegie Mellon University, 1995.
2. S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithms. In A. Prieditis and S. Russel, editors, *Proceedings of ICML95*, pages 38–46. Morgan Kaufmann, 1995.
3. Robert J. Collins and David R. Jefferson. Selection in massively parallel genetic algorithm. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*, pages 249–256. Morgan Kaufmann, 1991.
4. J. Culberson and Jonathan Lichtner. On searching 'alpha'-ary hypercubes and related graphs. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 263–290. Morgan Kaufmann, 1996.
5. K. Mathias D. Whitley and L. Pyeatt. Hyperplane ranking in simple genetic alghoritms. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 231–238. Morgan Kaufmann, 1995.
6. Y. Davidor. Epistasis variance: a viewpoint on ga-hardness. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 23–35. Morgan Kaufmann, 1991.
7. K. A. DeJong, W. M. Spears, and D. M. Gordon. Using markov chains to analyse gafos. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 115–137. Morgan Kaufmann, 1995.
8. S. Forrest and M. Mitchell. The performance of genetic algorithms on walsh polynomials: Some anomalous results and their explanation. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*, pages 182–189. Morgan Kaufmann, 1991.
9. S. Forrest and M. Mitchell. Relative building-block fitness and the building-block hypothesis. *Foundations of Genetic Algorithms 2*, pages 109–126, 1993.
10. D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.
11. D. E. Goldberg. Making genetic algorithms fly:a lesson from the wright brothers. *Advanced Technology for Developpers*, 2:1–8, February 1993.

12. J. Horn and D.E. Goldberg. Genetic algorithms difficulty and the modality of fitness landscapes. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 243–269. Morgan Kaufmann, 1995.
13. J. Horn, D.E. Goldberg, and K. Deb. Long path problems. In Y. Davidor, H.-P. Schwefel, and R. Manner, editors, *Proceedings of the* 3^{rd} *Conference on Parallel Problems Solving from Nature*, pages 149–158. Springer Verlag, 1994.
14. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.
15. L. Kallel and B. Naudts. Candidate longpaths for the simple genetic algorithm. In *Foundations of Genetic Algorithms 5*. Morgan Kaufmann, 1998. to appear.
16. L. Kallel, B. Naudts, and M. Schoenauer. On functions with a fixed fitness distance relation. Technical Report 399, CMAP, Ecole Polytechnique, 1998.
17. L. Kallel and M. Schoenauer. Alternative random initialization in genetic algorithms. In Th. Bäeck, editor, *Proceedings of the* 7^{th} *International Conference on Genetic Algorithms*, pages 268–275. Morgan Kaufmann, 1997.
18. L. Kallel and M. Schoenauer. A priori predictions of operator efficiency. In J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution'97*, LNCS. Springer Verlag, October 1997.
19. H. Kargupta and D. E. Goldberg. Serch, blackbox optimisation, and sample complexity. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 291–324. Morgan Kaufmann, 1996.
20. S. A. Kauffman. *Lectures in the Sciences of Complexity*, volume I of *SFI studies*, chapter Adaptation on Rugged Fitness Landscape, pages 619–712. Addison Wesley, 1989.
21. S. J. Louis and G. J. E. Rawlins. Predicting convergence time for genetic algorithms. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 141–161. Morgan Kaufmann, 1993.
22. B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*, pages 143–150. Morgan Kaufmann, 1991.
23. M. Mitchell and J.H. Holland. When will a genetic algorithm outperform hill-climbing ? In S. Forrest, editor, *Proceedings of the* 5^{th} *International Conference on Genetic Algorithms*, page 647, 1993.
24. B. Naudts and L.Kallel. Some facts about so called ga-hardness measures. Technical Report 379, CMAP, Ecole Polytechnique, Mars 1998. submitted.
25. N. J. Radcliffe and P. D. Surry. Fitness variance of formae and performance prediction. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 51–72. Morgan Kaufmann, 1995.
26. S. Rochet, G. Venturini, M. Slimane, and E.M El Kharoubi. A critical and empirical study of epistasis measures for predicting ga performances: a summary. In J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Artificial Evolution'97*, LNCS, pages 287–299. Springer Verlag, 1997.
27. G. Rudolph. Convergence of non-elitist strategies. In Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, editors, *Proceedings of the First IEEE International Conference on Evolutionary Computation*, pages 63–66. IEEE Press, 1994.
28. J. D. Schaffer, R. A. Caruana, L. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In J. D. Schaffer, editor, *Proceedings of the* 3^{rd} *International Conference on Genetic Algorithms*, pages 51–60. Morgan Kaufmann, 1989.
29. Michael D. Vose and Alden H. Wright. Stability of vertex fixed points and applications. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 103–11. Morgan Kaufmann, 1995.
30. D. Whitley. Fundamental principles of deception in genetic search. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991.
31. S.W. Wilson. Ga-easy does not imply steepest-ascent optimizable. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*. mk, 1991.