



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Bitwise regularity coefficients as a tool for  
deception analysis of a Genetic Algorithm*

Benoît Leblanc, Evelyne Lutton

**N° 3274**

October 1st 1997

————— THÈME 4 —————



*R*apport  
de recherche







## Bitwise regularity coefficients as a tool for deception analysis of a Genetic Algorithm

Benoît Leblanc, Evelyne Lutton \*

Thème 4 — Simulation et optimisation  
de systèmes complexes  
Projet FRACTALES

Rapport de recherche n3274 — October 1st 1997 — 47 pages

**Abstract:** We present in this paper a theoretical analysis that relates an irregularity measure of a fitness function to the so-called GA-deception. This approach is a continuation of a work that has presented a deception analysis of Hölder functions. The analysis developed here is a generalization of this work in two ways: we first use a “bitwise regularity” instead of a Hölder exponent as a basis for our deception analysis, second, we perform a similar deception analysis of a GA with uniform crossover. We finally propose to use the bitwise regularity coefficients in order to analyze the influence of a chromosome encoding on the GA efficiency, and present experiments with bits permutations and Gray encoding.

**Key-words:** Genetic Algorithms, optimization, bitwise regularity, deception analysis, fractals, Hölder functions.

*(Résumé : tsvp)*

\* [Benoit.LebLANc@inria.fr](mailto:Benoit.LebLANc@inria.fr), [Evelyne.Lutton@inria.fr](mailto:Evelyne.Lutton@inria.fr), <http://www-rocq.inria.fr/fractales/>

## Les coefficients de régularité bit à bit comme outil d'analyse de déceptivité pour un Algorithme Génétique

**Résumé :** Nous présentons ici une analyse théorique qui permet de relier une mesure d'irrégularité d'une fonction de fitness à une notion de difficulté (ou déceptivité) pour les Algorithmes Génétiques (AG). Dans des travaux antérieurs nous avons développé une analyse de la déceptivité des fonctions Höldériennes, la présente analyse en est une généralisation selon deux voies : premièrement nous employons la "régularité bit à bit" au lieu d'un exposant de Hölder comme mesure d'irrégularité à la base de l'analyse de déceptivité, ensuite nous étendons l'analyse de déceptivité à un AG avec croisement uniforme. Cette approche nous permet de proposer l'emploi des coefficients de régularité bit à bit comme outil d'évaluation de l'influence du codage des chromosomes sur l'efficacité de l'AG. Nous produisons quelques expériences sur les permutations de bits et le codage de Gray.

**Mots-clé :** Algorithmes Génétiques, optimisation, régularité bit à bit, analyse de déceptivité, fractales, fonctions Höldériennes.

## 1 Introduction.

Theoretical investigations on Genetic Algorithms (GA) and Evolutionary Algorithms (EA) in general concern mainly convergence analysis (and convergence speed analysis on a locally convex optimum for EA), influence of the parameters, and GA-hardness analysis. For GA, our main concern here, these analyses are based on different approaches :

- Proofs of convergence based on Markov chain modeling [6, 3, 1, 20].
- Deceptive functions analysis, based on Schema analysis and Holland's original theory [14, 8, 9, 11], which characterizes the efficiency of a GA, and allows to shed light on "GA-hard" functions.
- Some rather new approaches are based on an explicit modelization of a GA as a dynamical system [16, 22].

Deception has been intuitively related to the biological notion of epistasis [5], which can be understood as a sort of "non-linearity" degree. It can also be related, to the so called "fitness landscape" analyses (see for example [19]). In any ways, it basically depends on :

- the parameter setting of the GA,
- the shape of the function to be optimized,
- the chromosome encoding , i.e. the "way" of scanning the search space.

In a previous work [17] it has been proven that some tools, that have been developed in the framework of fractal theory, can be used in order to refine a deception analysis of Genetic Algorithms. This work has mainly related an irregularity measure (Hölder exponent) of the function to be optimized to its deceptiveness. We first recall in section 2 these results, that allow to model the influence of some of the GA parameters. The main hypothesis of this previous analysis is that the fitness function can be considered as the sampling of an underlying continuous Hölder function. In section 3 we will then present a generalization of this work that considers another regularity measure of the fitness function, the bitwise regularity, and which does not support the Hölder hypothesis anymore.

The GA modeled in this framework is the so-called canonical GA, i.e. with proportionate selection (roulette wheel selection), one point crossover and mutation, at fixed rates  $p_c$  and  $p_m$  all along the GA run. This analysis first suggests that a simple bit reordering as an encoding change may decrease deception ; experiments are presented in section 4 that show the validity extents of this analysis. In section 5 we then present a similar theoretical analysis for a canonical GA with uniform crossover, which is an operator that is largely used in real world applications.

Besides the intuitive fact that it relates the irregularity of the fitness function to its "difficulty", one important application of this theoretical analysis is that it provides a mean to "measure" (of course to a certain extent, due to the intrinsic limitations of deception theory) the influence of the chromosome encoding. We present in section 6 some experimentations with the Gray encoding that prove the interest, of such an approach.

## 2 Background and previous work.

In this section we quickly remind the definitions of schemata, deception and Hölder exponents.

### 2.1 The canonical Genetic Algorithm.

In the frame of our study, we will restrict ourselves to the case of the *canonical* GA (also called *simple* GA), which aims to find the maximum of a real positive function (called the *fitness* function) defined on the space of binary strings (also called *chromosome*) of size  $l$ :

$$f : \Omega^l = \{0, 1\}^l \rightarrow \mathbb{R}^+$$

The CGA has the following steps:

1. Creation of an initial *population*, a set of *individuals* that are represented by a point of  $\Omega^l$ . The binary representation of an individual is its *genotype* while its corresponding point in the search space is its *phenotype*. The correspondence genotype/phenotype is an encoding/decoding, that may not always be a bijection.
2. Computation of the fitness value for each individual.
3. Selection of  $N$  individuals to build a *parental pool*.
4. Successive applications of the crossover and mutation to the parental pool resulting in the creation of the population of the *new generation*. Back to 2 if a stopping criteria has not been reached (generally a given number of generations).
5. End. The individual having the highest fitness value is extracted from the final population as the solution to the problem.

The **selection** step is based on a random shot with replacement of individuals in order to build the parental pool. Each individual has a selection probability equal to its relative fitness value (this method is called the *Roulette wheel selection*):

$$P(i) = \frac{fitness(i)}{\sum_{j=1}^N fitness(j)}$$

The **crossover**, applied with a probability  $p_c$  on couple of individuals, mixes two chromosomes of the parents in order to create the two new offsprings: a position on the chromosome is randomly chosen between 1 and  $(l-1)$  (each one with equal probability), then the strings are exchanged from this point. This is the *one point crossover* (see figure 1).

The **mutation** is then applied to all resulting new individuals. It acts in changing the value of each bit or *gene* of the binary string with a given probability  $p_m$ , usually very low.

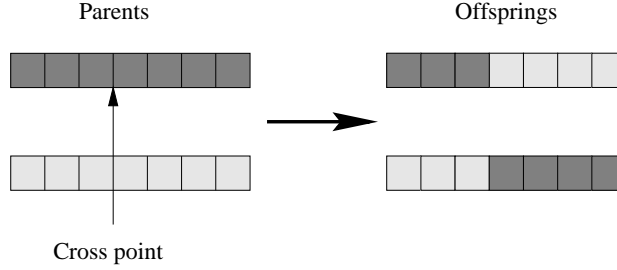


Figure 1: One point crossover.

## 2.2 Schemata.

Schemata has been widely studied in the field of GA, and are the basis of the deception analysis. A schema corresponds to a subset of the space  $\Omega^l = \{0, 1\}^l$  (the space of binary strings of length  $l$  for a GA using binary encoding), or more precisely a hyper-plan of  $\Omega^l$ . An additional symbol '\*', representing a wild card ('0' or '1') is used to represent a schema. For example, if  $l = 4$ , the strings  $i_1 = 0101$  and  $i_2 = 1101$  are the two elements of the schema  $H = *101$ . The order of a schema,  $O(H)$ , is defined as the number of fixed positions in  $H$ , and the defining length,  $\delta(H)$ , as the distance between the first and the last fixed positions of  $H$ . A fundamental theorem about schemata is the following:

### Theorem 1 (Schema theorem, Holland [14])

For a given schema  $H$  let:

- $m(H, t)$  be the expected number of representatives of the schema  $H$  in the population  $P(t)$  ( $t$  indexes the number of the generation):  $m(H, t) = |H \cap P(t)|$ .
- $\tilde{f}(H, t)$  be the mean fitness value of the representatives of  $H$  in the population  $P(t)$ :

$$\tilde{f}(H, t) = \frac{1}{|H \cap P(t)|} \sum_{i \in H \cap P(t)} f(i)$$

- $\bar{f}(t)$  be the mean fitness value of the individuals of  $P(t)$ :

$$\bar{f}(t) = \frac{1}{|P(t)|} \sum_{i \in P(t)} f(i)$$

- $p_c$  and  $p_m$  be respectively the (one point) crossover and mutation probabilities.

Then :

$$m(H, t + 1) \geq m(H, t) \frac{\tilde{f}(H, t)}{\bar{f}(t)} \left[ 1 - p_c \frac{\delta(H)}{l-1} - O(H)p_m \right]$$

### 2.3 Deception analysis.

A famous consequence of the schema theorem is that the schemata having a short defining length, a small order and a mean fitness better than the population mean fitness will be more and more represented in the successive generations (such schemata are called *building blocks*, [10]). This remark leads to the conclusion that if the global optimum of the fitness function  $f$  is the intersection of such good building blocks, a GA will easily find it. On the contrary, if the intersection of these building blocks is a secondary optimum, the population will preferably converge onto it, missing the global one. In this situation the GA will be considered to have failed<sup>1</sup> and  $f$  will be called *deceptive*.

More formally, Goldberg ([8], [9]) defined the *static deception*: The *selection* results in an expected greater mean fitness for the set of individuals selected for reproduction, than for the preceding population. But this mean value will be changed by the application of genetic operators. It follows that the GA can be considered as attracted toward the optima of a function  $f'$ , defined for each point of  $\Omega^l$  as its expected fitness value after the application of crossover and mutation. The function  $f$  will be called deceptive for a GA with a given parameter setting, if the global optima of  $f'$  and  $f$  differ. This function may be calculated with the help of the Walsh basis:

**Definition 1 (Walsh polynomials)**

*They form an orthogonal basis of the set of functions defined on  $\Omega^l$ :*

$$\psi_j(x) = \prod_{t=0}^{l-1} (-1)^{x_t j_t} = (-1)^{\sum_{t=0}^{l-1} x_t j_t} \quad (1)$$

Where  $x_t$  and  $j_t$  denote the values of the  $t^{\text{th}}$  bit of the binary decomposition of  $x$  and  $j$ .

The projection of a function  $f$  on this basis is:

$$f(x) = \sum_{j=0}^{2^l-1} w_j \psi_j(x) \quad \text{with} \quad w_j = \frac{1}{2^l} \sum_{x=0}^{2^l-1} f(x) \psi_j(x)$$

The coefficients  $w_j$  are called *Walsh coefficients* and are strongly related to schemata. Roughly, a given  $w_j$  is related to schemata having fixed bits at the position where  $j$  has '1' in its binary decomposition. Consequently, the *adjusted Walsh coefficients* ("adjusted" according to genetic operators) may be calculated (see [8, 18]):

$$w'_j = w_j \left(1 - p_c \frac{\delta(j)}{l-1} - 2p_m O(j)\right) \quad (2)$$

---

<sup>1</sup>Only in an optimization perspective. Recast in a more general context, the success of a GA may not only be related to its ability to find a global optimum at each trial, but rather to rapidly find good solutions.



Where  $O(j)$  denotes the number of '1' in the binary decomposition and  $\delta(j)$  the distance between the first and the last '1'.

$$f'(x) = \sum_{j=0}^{2^l-1} w'_j \psi_j(x) \quad (3)$$

Defining the two following sets (near optimal sets of  $f$  and  $f'$ ) for a given  $\epsilon$  :

$$N_\epsilon = \{x \in [0..2^l] / |f(x) - f^*| \leq \epsilon\}$$

and

$$N'_\epsilon = \{x \in [0..2^l] / |f'(x) - f'^*| \leq \epsilon' = \frac{f'^* - w_0}{f^* - w_0} \epsilon\}$$

the definition of static deception follows [9] :

**Definition 2** *A function-coding combination is **statically deceptive** at the level  $\epsilon$  when  $N_\epsilon - N'_\epsilon \neq 0$ .*

## 2.4 Deception analysis on Hölder functions.

The work presented in [17] aims to characterize the deception of a given function  $f$ , considered as the binary encoding of the sampling of a Hölder function on the interval  $[0, 1]$  :

**Definition 3 (Hölder function of exponent  $h$  [7])**

*Let  $(X, d_X)$  and  $(Y, d_Y)$  be two metric spaces. A function  $F : X \rightarrow Y$  is called Hölder function of exponent  $h > 0$ , if for each  $x, y \in X$  such that  $d_X(x, y) < 1$ , we have :*

$$d_Y(F(x), F(y)) \leq k \cdot d_X(x, y)^h \quad (4)$$

for some  $k > 0$ .

Although a Hölder function is always continuous, it needs not to be differentiable, and if it is Hölder with exponent  $h$ , it is Hölder with exponent  $h'$  for all  $h' \in ]0, h]$ . Intuitively, we may characterize a Hölder function of low exponent  $h$  as more "irregular" than a Hölder function of higher  $h$ .

It is possible to consequently establish a relation between  $h$  and  $|f - f'|$ . To reach this point the following basis is used :

**Definition 4 (Haar polynomials)**

*They form an orthogonal basis of the set of functions defined on  $\Omega^l$  :*

$$H_{2^q+m}(x) = \begin{cases} 1 & \text{for } (2m)2^{l-q-1} \leq x < (2m+1)2^{l-q-1} \\ -1 & \text{for } (2m+1)2^{l-q-1} \leq x < (2m+2)2^{l-q-1} \\ 0 & \text{otherwise in } \{0, \dots, 2^l - 1\} \end{cases} \quad (5)$$

With  $q \in [0..(l-1)]$ ,  $m \in [0..(2^q-1)]$  and  $j = 2^q + m \in [0..(2^l-1)]$ . The projection of  $f$  on this basis is :

$$f(x) = \sum_{j=0}^{2^l-1} h_j H_j(x) \quad \text{and} \quad h_j = \frac{1}{2^{l-q}} \sum_{x=0}^{2^l-1} f(x) H_j(x) \quad (6)$$

As the Haar coefficients may be bounded :

$$\forall j = 2^q + m, \quad |h_j| \leq \frac{k}{2} 2^{-h(q+1)} \quad (7)$$

the following theorem has been proved [17] :

**Theorem 2** *Let  $f$  be the sampling on  $l$  bits of a Hölder function of exponent  $h$  and constant  $k$ , defined on  $[0, 1]$ , and let  $f'$  be defined as in (3). Then :*

$$\forall x \in \{0, \dots, 2^l - 1\} \quad |f(x) - f'(x)| \leq k \cdot B(p_m, p_c, l, h)$$

with

$$B(p_m, p_c, l, h) = \frac{p_c}{l-1} 2^{-h} \left[ \frac{2^{-l(h+1)} - 1}{2^{-(h+1)}} + \frac{(1 - 2^{l-h})(2^{-hl} - 1) - l 2^{-hl}(1 - 2^{-h})}{(2^{-h} - 1)^2} \right] \\ + p_m \frac{2^{-h}}{(2^{-h} - 1)^2} [1 + 2^{-hl}(l 2^{-h} - l - 1)]$$

We can see that if  $B$  grows,  $f$  has more and more chances to be deceptive. Without going into details,  $B$  is a function of 4 real parameters having the following behavior :

- $B$  is decreasing with respect to  $h$ .
- $B$  is increasing with respect to  $p_m$ , and  $p_c$ .
- $B$  increases when  $l$ , when  $l$  is small, reaches a maximum for a value  $l_{max}$ , and decreases for  $l > l_{max}$ .

### 3 A bitwise regularity characterization.

The previous analysis is based on an irregularity characterization with respect to an underlying distance that is the Euclidean distance on  $[0, 1]$ . This approach is straightforward for fitness functions defined on  $\mathbb{R}$ , and in the general case it is always possible to consider the fitness function as the sampling of an underlying one-dimensional Hölder function. It is however less evident in this latter case that the Hölder exponent reflects in a simple way the irregularity of the fitness function (it may appear for example more irregular than it is in a multidimensional space). This is the reason why we present in this paper a similar irregularity analysis but with respect to the Hamming distance on the set of binary strings. Another justification is also that it is easier to represent the action of genetic operators with respect to the Hamming distance.

### 3.1 Bitwise regularity coefficients.

A grained Hölder exponent may be defined for a box of size  $\varepsilon$  centered on  $x$ ,  $B_\varepsilon(x)$ , of the  $\{0,1\}^l$  space:  $\varepsilon$  being expressed with respect to a distance proportional to the Hamming distance, for example:  $d(x,y) = \frac{1}{l} \cdot d_H(x,y)$  and  $B_\varepsilon(x) = \{y \in \{0,1\}^l / d(x,y) \leq \varepsilon\}$ :

$$\alpha_\varepsilon(x) = \frac{\log \mu(B_\varepsilon(x))}{\log \varepsilon}$$

$\mu$  is a measure of  $B_\varepsilon(x)$ , for example:

$$\begin{aligned} \mu(B_\varepsilon(x)) &= \sup_{y \in B_\varepsilon(x)} \{|f(x) - f(y)|\} \\ \text{then } \alpha_\varepsilon(x) &= \frac{\log \left[ \sup_{y \in B_\varepsilon(x)} \{|f(x) - f(y)|\} \right]}{\log \varepsilon} \end{aligned}$$

In the continuous case, the local behavior of  $f$  may be captured as  $\varepsilon \rightarrow 0$ , as we are dealing here with a discrete space, we can fix  $\varepsilon$  at the smallest positive value, i.e.:  $\frac{1}{l}$ . The box  $B_\varepsilon(x)$  may also be defined with respect to a particular “coordinate”, i.e. a fixed bit position ( $l - q - 1$ ), and for the Hamming distance between  $x$  and  $y$ :

$$B_{\frac{1}{l}}^q = \left\{ y \in \{0,1\}^l \mid d(x,y) = \frac{1}{l} \text{ and } y, x \text{ differ only on the } (l - q - 1)^{\text{th}} \text{ bit} \right\}$$

with  $B_{\frac{1}{l}}^q$ : “oriented” boxes of size 1. For these particular boxes:

$$\alpha_\varepsilon(x) = \frac{\log \left[ \sup_{y \in B_{\frac{1}{l}}^q(x)} \{|f(x) - f(y)|\} \right]}{-\log l}$$

An irregularity characterization with respect to these boxes is then:

$$\alpha^q = \inf_{x \in \{0,1\}^l} \alpha_\varepsilon^q(x) = \frac{\log \sup \{|f(x) - f(y)| \mid x, y \text{ differ only at position } (l - q - 1)\}}{-\log l}$$

We thus propose to compute the following coefficients, that naturally represents what we can call a bitwise regularity measure of the function  $f$ :

**Definition 5 (Bitwise regularity coefficients)** Let  $f$  be a function defined on  $\Omega^l$ :

$$\forall q \in \{0, \dots, l - 1\}, C_q = \sup_{x \in \Omega^l} \{|f(x) - f(x'_{l-q-1})|\}$$

with  $x'_{l-q-1}$  and  $x$  differing only with respect to one bit at the position  $(l - q - 1)$ .<sup>2</sup>

<sup>2</sup>The less significant bit being at position 0.

In other terms, the  $C_q$  coefficient represents the maximum fitness variation due to a bit flip at the position  $(l - q - 1)$ .

Therefore, we can show that :

$$\forall j = 2^q + m, \quad |h_j| \leq \frac{C_q}{2}$$

In the same way as in [17], with the help of the Haar basis, the following theorem has been established (see appendix A for a demonstration) :

**Theorem 3** *Let  $f$  be a function defined on  $\Omega^l$  with bitwise regularity coefficients  $(C_q)_{q \in \{0, \dots, l-1\}}$ , and let  $f'$  be defined as in (3). Then  $\forall x \in \Omega^l$  :*

$$|f(x) - f'(x)| \leq \frac{p_c}{l-1} \sum_{q=0}^{l-1} C_q \left( \frac{1 + 2^q(q-1)}{2^q} \right) + p_m \sum_{q=0}^{l-1} C_q(q+1) \quad (8)$$

Furthermore, this result still holds when the order of the  $C_q$  values is reversed, so the final bound is the one minimizing the preceding expression (see appendix B for a demonstration).

We also have to note that the bits do not have the same role in this bound expression. In fact their relative weight is strictly increasing with respect to the index  $q$ . Sorting them (either in increasing or decreasing order) would then minimize this bound, suggesting that the simple change of coding consisting in a bits permutation would make the function easier. This feature can be explained by the fact that the one point crossover disrupts more easily a combination of a few genes spread at each extremities of the chromosome than if these genes were grouped at one extremity. Reordering the bits in order to sort the bitwise regularity coefficients is then equivalent to group the most “sensible” genes at one extremity of the chromosome. Some experiments presented in the section 4 partially support this interpretation.

### 3.2 Bitwise regularity coefficients compared to Hölder exponent.

If we suppose that the fitness function  $f$  is the sampling on  $l$  bits of a Hölder function of exponent  $h$  and constant  $k$ , defined on  $[0, 1]$ , the bound of *theorem 3* is lower than the bound of *theorem 2*.

One can easily show, (see appendix C), that :

$$C_q \leq k \cdot 2^{-(q+1)h} \quad (9)$$

as we have :

$$|h_j| \leq \frac{C_q}{2} \quad \text{and} \quad |h_j| \leq \frac{k}{2} \cdot 2^{-(q+1)h}$$

and as the bound on  $|f - f'|$  is a linear function of the bounds on the  $|h_j|$ , it follows immediately that the bound of *theorem 3* is the lowest (see the figure 2 for a visual comparison of the bounds of Haar coefficients). Moreover, the estimation of the bitwise regularity coefficients is computationally cheaper than the estimation of the Hölder exponent and its associated constant  $k$ .

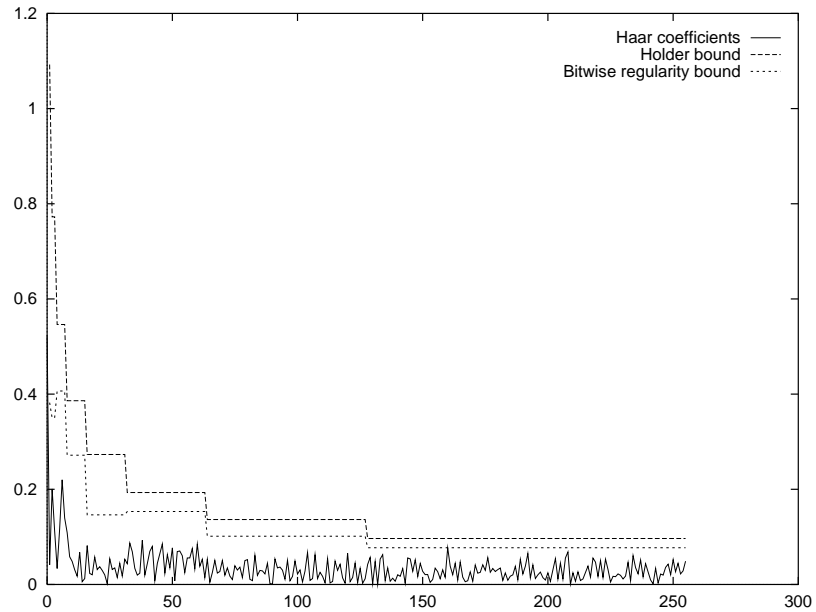


Figure 2: Haar coefficients of a Weierstrass function with Hölder exponent  $h=0.5$

## 4 First experiments.

The following experiments have been performed with the canonical GA. For each fitness function tested, the GA is ran many times and the following performances are recorded for each generation:

1. Average of the population mean fitness values.
2. Average of the best individual fitness values.
3. Proportion of the runs whose population contains a global optimum.

In the title of the figures displaying these performances, they are referenced as “Mean”, “Maximum” and “Proportion” followed by the name of the function.

The following GA parameters are also specified for each function :

$l$	encoding size (number of bits)
$N$	population size
$Gen$	number of generations for a run
$Runs$	number of runs
$p_c$	crossover probability
$p_m$	mutation probability

#### 4.1 Weierstrass functions

The fitness functions are the sampling of Weierstrass functions [7]:

$$W_{b,s}(x) = \sum_{i=1}^{\infty} b^{i(s-2)} \sin(b^i x) \quad \text{with } b > 1 \text{ and } 1 < s < 2$$

The Hölder exponent is directly given by  $h = (2 - s)$ . For the present case, we set  $b = 5$ .

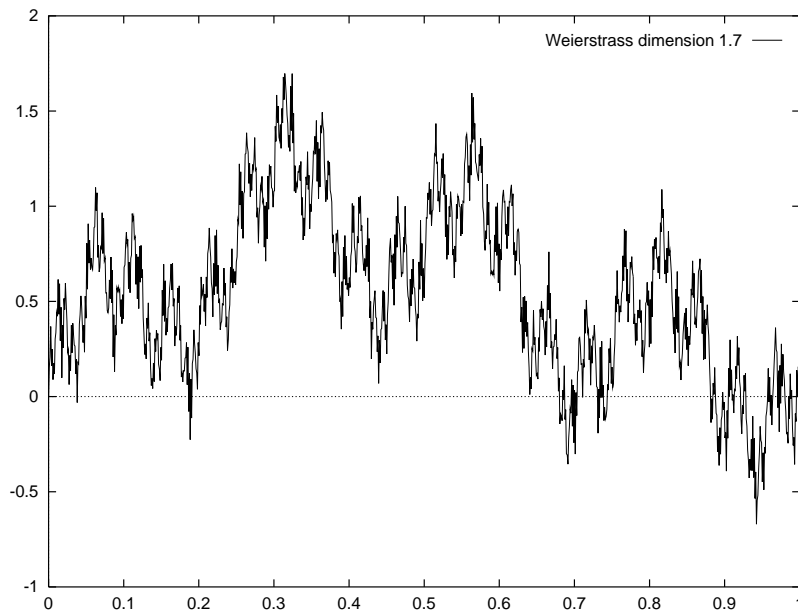
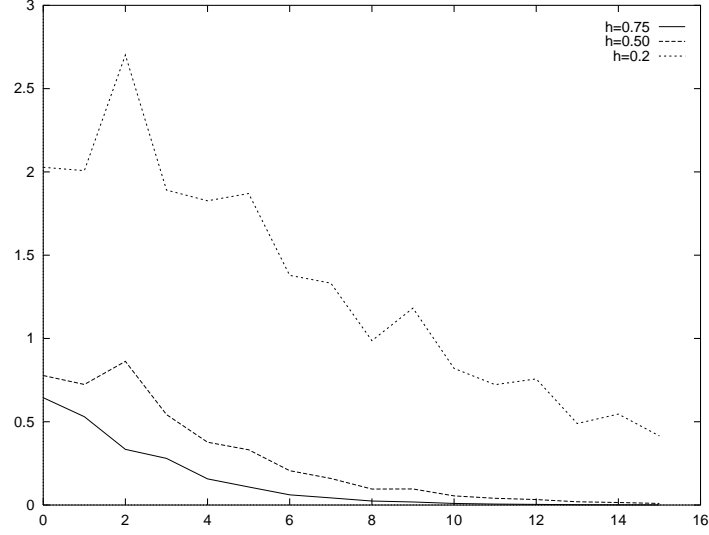


Figure 3: Weierstrass function with Hölder exponent  $h = 0.3$

The tests are performed with  $h = 0.75$  (*W75*),  $h = 1.5$  (*W50*) and  $h = 0.2$  (*W20*). The sampling domain is  $[0, 1]$ , and the classical encoding is the usual mapping:

$$\begin{aligned} \Omega^l &\rightarrow [0, 1] \\ b_{(l-1)}b_{(l-2)} \dots b_0 &\rightarrow x = \frac{1}{2^l} \sum_{i=0}^{l-1} b_i \cdot 2^i \end{aligned}$$

The figure 4 displays the  $C_q$  values obtained for different values of  $h$  with a 16-bits sampling (recall that  $C_0$  corresponds to the bit  $b_{15}$  and  $C_{15}$  to  $b_0$ ). The exact values are not important (they must be compared to the function maximum), and only the relative importance of the  $C_q$  are relevant. We can see that they are more or less decreasing, and they tend to be of the same order as  $s$  tends to 2.

Figure 4:  $C_q$  values for W75, W50 and W20.

$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
16	512	100	100	1.0	0.005

	W75	W50	W20
$B_c$	0.254	0.776	6.96
$B_m$	1.0	1.94	9.85

Table 1: Parameters and bound values for W75, W50 and W20.

The second encoding is derived from the first one with a permutation of the bits in order to increase the value of the bound (8):

$$b_{(l-1)}b_{(l-3)} \dots b_1b_0 \dots b_{(l-4)}b_{(l-2)}$$

Table 1 displays the parameters of the runs and the bound values:  $B_c$  stands for the bound corresponding to the classical encoding and  $B_m$  stands for the bound corresponding to the modified encoding. For all the functions,  $B_m$  is higher than  $B_c$  and the tests displayed in figures 5 to 7 show that a GA with the classical encoding show the best performances, as predicted (“C1” stands for the classical encoding and “C2” for the modified encoding).

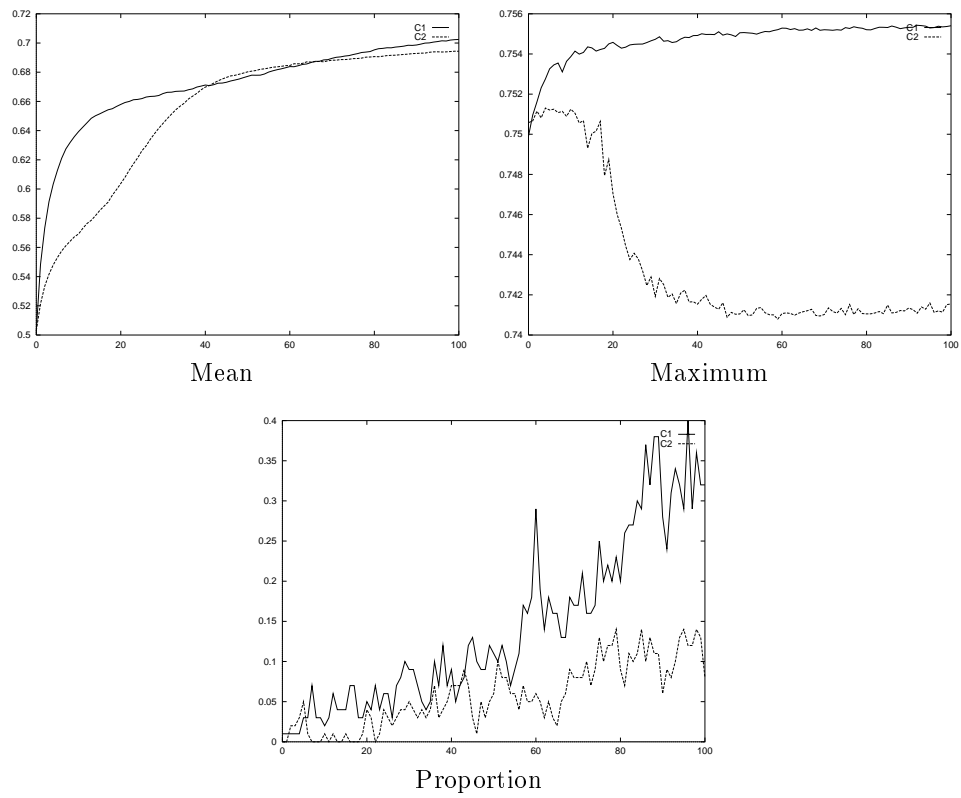


Figure 5: Performances on W75.



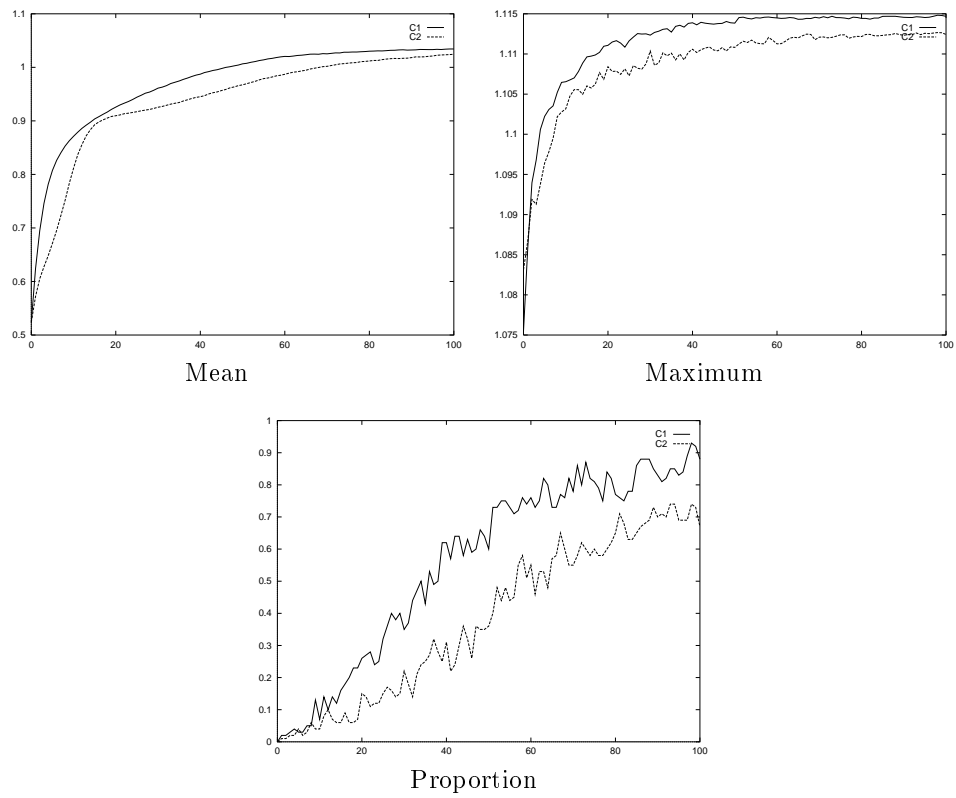


Figure 6: Performances on W50.

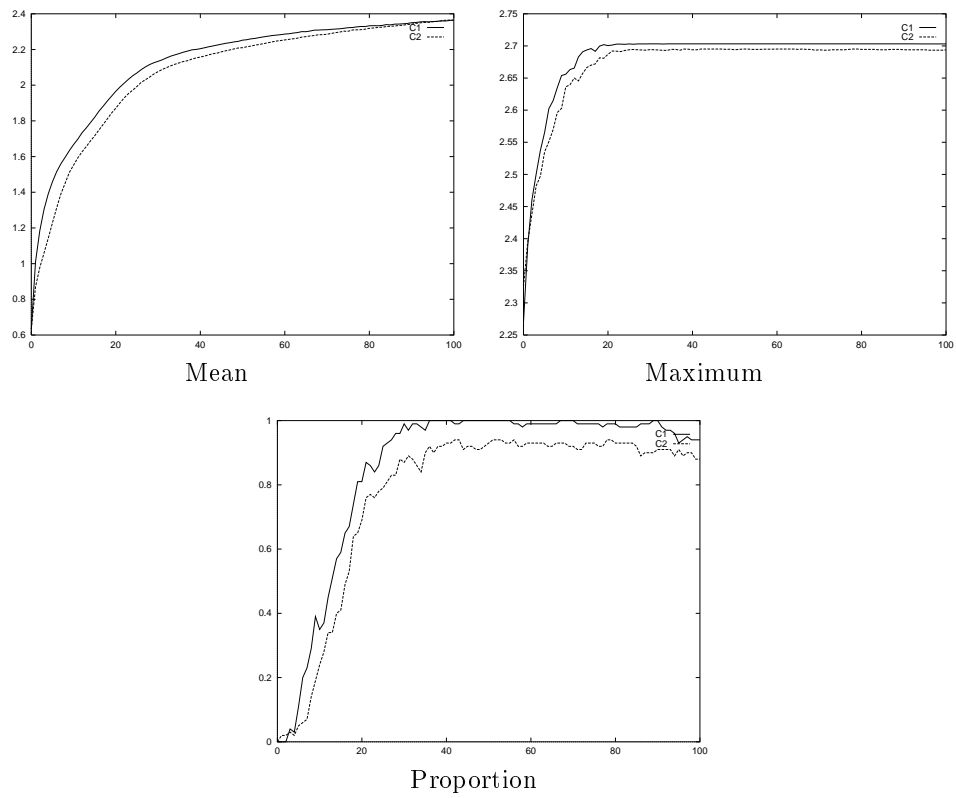


Figure 7: Performances on W20.

## 4.2 Function M2.

This function, from [12], is also the sampling of a continuous function defined on the interval  $[0, 1]$  (see figure 8):

$$\forall x \in [0, 1], \quad M2(x) = e^{-2(\ln 2)\left(\frac{x-0.1}{0.8}\right)^2} \sin^6(5\pi x)$$

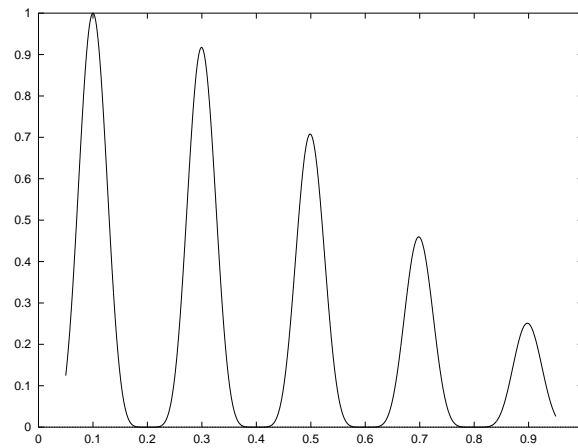


Figure 8: Function M2

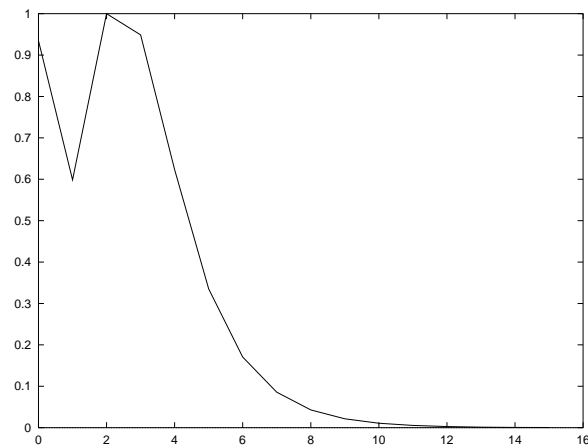
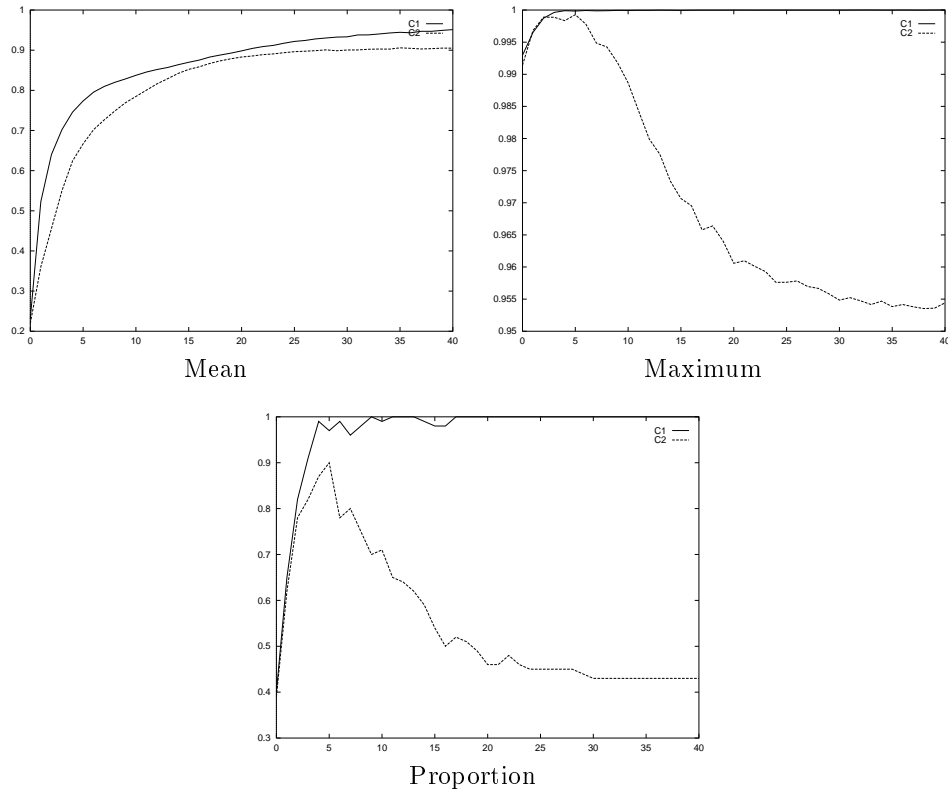


Figure 9: Bitwise regularity coefficients of function M2.

The  $C_q$  coefficients are displayed in figure 9. The same encodings as for the Weierstrass functions are tested. As for the Weierstrass functions the bound of the modified encoding  $B_c$  is higher than the bound of the classical encoding, and the performances presented in figure 10 show that the GA performs better in the last case, as predicted.



$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
16	200	100	40	1.0	0.005

GA parameters

$B_c$	$B_m$
0.676	2.03

Bound values

Figure 10: Tests on function M2.

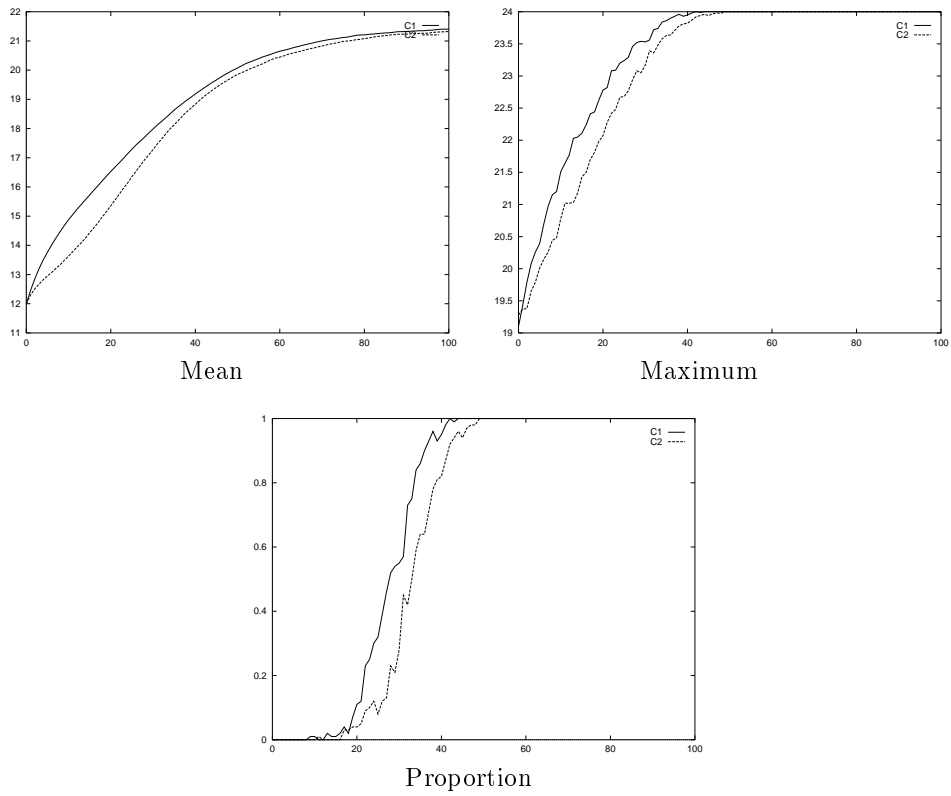
### 4.3 Function EPI6.

We constructed this function in order to create a strong dependency between genes (such functions are called *epistatic* [5]). It is the sum of 6 sub-functions *EPI* defined on 4 bits :

$$EPI(b_3b_2b_1b_0) = \begin{cases} \sum_{i=0}^2 (1 - b_i) & \text{if } b_3 = 0 \\ 1 + \sum_{i=0}^2 b_i & \text{if } b_3 = 1 \end{cases}$$

Each sub-function has one global optimum ( $EPI(1111) = 4$ ) and a local optimum with respect to the Hamming distance ( $EPI(0000) = 3$ ). The bitwise regularity coefficients  $C_q$  are then  $(4, 1, 1, 1)$ . *EPI6* being the concatenation of 6 functions *EPI*, its  $C_q$  coefficients are  $(4, 1, 1, 1, \dots, 4, 1, 1, 1)$ . *EPI6* has then one global optimum and  $(2^6 - 1) = 63$  local optima.

The bound for the modified encoding is lower than for the original one. We would then expect an improvement of the GA performances, but as we can see in figure 11, this is not the case. A possible explanation is that the second encoding breaks the proximity between the “control” bit and its “controlled” bits, then the relation between these 4 bits is more often disrupted by the one point crossover. This counter-example highlights the limits the limits of the previous deception analysis, that only takes into account the bitwise behavior of the fitness functions and that does not take into account “epistatic” behaviors.



$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
24	512	100	100	1.0	0.005

GA parameters

$B_c$	$B_m$
20.7	14.3

Bound values

Figure 11: Tests on function EPI6.

## 5 Deception analysis of a GA with uniform crossover.

As we have seen, the bound on  $|f - f'|$  derived from the bitwise regularity coefficients  $C_q$  depends on their relative order, due to the use of the one point crossover. The aim of this section is to establish analogous results for the uniform crossover [21], for which this positional bias no longer exists.

### 5.1 Definition and implications for schemata and deception.

The uniform crossover no longer relies on the use of crossing points. It produces a more uniform “mixing” of the genetic material: each gene of an offspring is randomly and independently chosen (with probability  $1/2$ ) from the two parents chromosomes. The other offspring inherits the complementary genes (the figure 12 illustrates this principle).

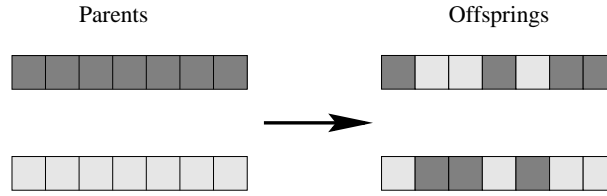


Figure 12: Uniform crossover.

The schema theorem may be adapted for this operator: with the one point crossover, the probability of schema disruption  $p_d$  is bounded:

$$p_d \leq \frac{\delta(H)}{l-1}$$

In the same manner, we can bound the disruption probability for the uniform crossover by observing that once the first fixed bit of the schema is allocated to one of the offsprings, the schema will always survive if all other fixed bits are allocated to the same offspring.

$$p_d \leq \left(1 - \left(\frac{1}{2}\right)^{O(H)-1}\right)$$

We then have the new bound for the schema theorem for a GA with uniform crossover:

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - p_c \left(1 - \left(\frac{1}{2}\right)^{O(H)-1}\right) - O(H)p_m\right] \quad (10)$$

As for the one point crossover  $p_d$  is conservatively set to  $\left(\frac{\delta(h)}{l-1}\right)$ , if we set  $p_d$  to the upper bound (8), then the new adjusted Walsh coefficients are :

$$w'_j = w_j \left[ 1 - p_c \left( 1 - \left( \frac{1}{2} \right)^{O(j)-1} \right) - 2p_m O(j) \right] \quad (11)$$

## 5.2 Implication for the bounds on $|f - f'|$ .

If we include the previous modification in the calculation of the bound of the theorem 2 we get the new expression (see appendix E for a demonstration) :

**Theorem 4** *Let  $f$  be the sampling on  $l$  bits of a Hölder function of exponent  $h$  and constant  $k$ , defined on  $[0, 1]$ , and let  $f'$  be defined as in (3). Then :*

$$\forall x \in \{0, \dots, 2^l - 1\} \quad |f(x) - f'(x)| \leq k \cdot B(p_m, p_c, l, h)$$

with

$$B(p_m, p_c, l, h) = p_c \frac{2^{-h}(2^{-hl} - 1)}{2^{-h} - 1} + p_m \frac{2^{-h}}{(2^{-h} - 1)^2} [1 + 2^{-hl}(l2^{-h} - l - 1)] \quad (12)$$

Furthermore, the bound (8) calculated for the uniform crossover (see appendix D for a demonstration), leads to the following theorem :

**Theorem 5** *Let  $f$  be a function defined on  $\Omega^l$  with bitwise regularity coefficients  $(C_q)_{q \in \{0, \dots, l-1\}}$ , and let  $f'$  be defined as in (3). If we consider the set  $S$  of permutations defined on the set  $\{0, \dots, l-1\}$ , then  $\forall x \in \Omega^l$  :*

$$|f(x) - f'(x)| \leq \min_{\sigma \in S} \left\{ p_c \sum_{q=0}^{l-1} C_{\sigma^{-1}(q)} + p_m \sum_{q=0}^{l-1} C_{\sigma^{-1}(q)} * (q+1) \right\} \quad (13)$$

We immediately see that the permutation that minimizes the upper bound is the one that sorts the  $C_{\sigma^{-1}(q)}$  in decreasing order.

Practically, if it is possible to get the  $C_q$  values (or good estimations), it is hard to draw conclusions from the value of the bound (13). But if we consider the effect of an encoding change on it, it is interesting to see if its variation is experimentally correlated to the performances of the GA. Intuitively, the hypothesis is formulated as follows: if an encoding change (such as a Gray code) induces a decrease of the bound (13), the GA should perform better with this new encoding, and conversely. We present experiments with the Gray code in the next section.



## 6 Experiments with uniform crossover.

The notations are the same as in section 4.

The Gray code is a largely used encoding change :

$$K : \Omega^l \rightarrow \Omega^l , \quad K(x) = g \quad \text{with}$$

$$g_i = \begin{cases} x_{(l-1)} & \text{if } i = (l-1) \\ (x_{(i+1)} \text{ XOR } x_i) & \text{if } (l-1) < i \leq 0 \end{cases}$$

### 6.1 Function $f_1$ .

This function is one of the *De Jong Five-Functions Test Bed* [15], turned into a maximization problem :  $F1(X) = (\max(f_1) - f_1(X))$ , with :

$$f_1(X) = \sum_{i=1}^3 \left( X^{(i)} \right)^2 \quad \text{with} \quad -5.12 \leq X^{(i)} \leq 5.12$$

This function is 3-dimensional and each component is defined on 10 bits. Four different mappings from  $\Omega^{10}$  to  $[-5.12, 5.12]$  have been experimented. Let  $x$  be any of the  $X^{(i)}$  coded on 10 bits  $b_9 b_8 \dots b_1 b_0$  :

- *Code1* : a classical signed integer binary encoding, mapped to  $[-5.12, 5.12]$  :

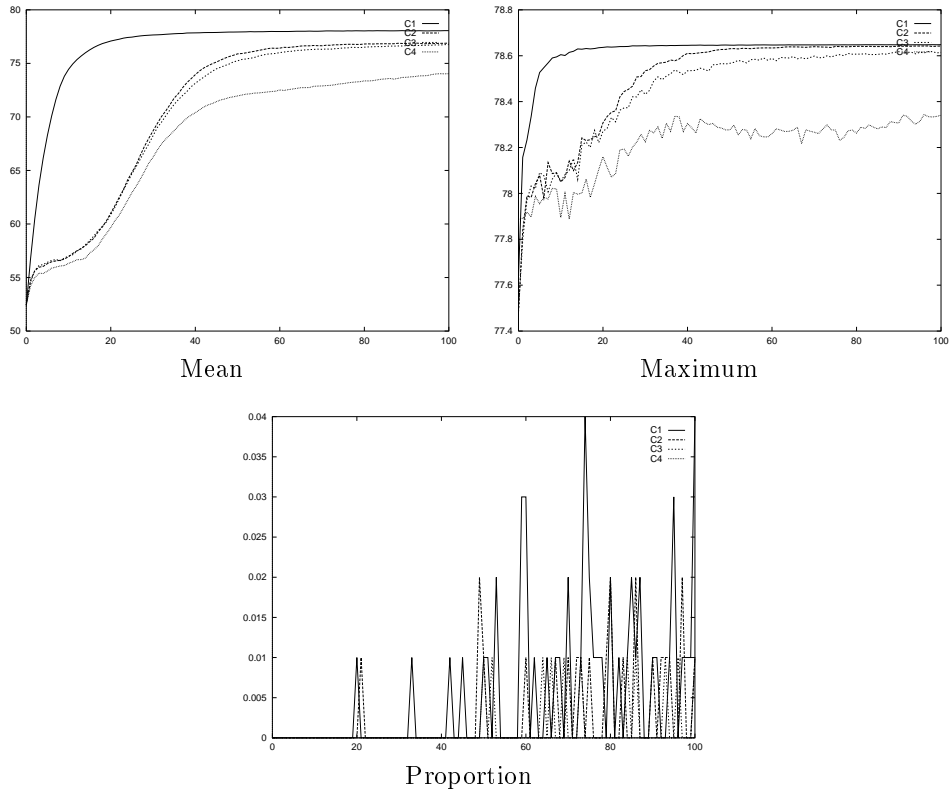
$$x = \frac{1}{100} (-1)^{b_9} \sum_{j=0}^8 2^j b_j$$

- *Code2* : an unsigned binary integer encoding, mapped to  $[-5.12, 5.12]$  :

$$x = \frac{1}{100} \left( \sum_{j=0}^9 2^j b_j - 512 \right)$$

- *Code3* : same as *Code1* but with Gray encoding for  $b_8 b_7 \dots b_1 b_0$ .
- *Code4* : same as *Code2* but with Gray encoding for  $b_9 b_8 \dots b_1 b_0$ .

In figure 13, we see that the bound is increasing with each new encoding, and that the performances of the GA decrease as predicted, though it is measured only through *Mean* and *Maximum*. The runs that found the global optimum were very rare, since a lot of solutions have a fitness value very close to the optimum (due the absence of scaling, the GA is unable to distinguish them).



$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
30	128	100	100	1.0	0.005

GA parameters

<i>Encoding</i>	<i>Code1</i>	<i>Code2</i>	<i>Code3</i>	<i>Code4</i>
<i>Bound</i>	131	210.3	235.2	313.6

Bound values

Figure 13: Tests on function F1.

## 6.2 Function $f_2$ .

This function is also one of the *De Jong Five-Functions Test Bed*, turned into a maximization problem:  $F2(X) = (\max(f_2) - f_2(X))$ , with:

$$f_2(X) = 100 * ((X^{(1)})^2 - X^{(2)})^2 + (1 - X^{(1)})^2$$

with  $X^{(i)} \in [-2.048, 2.048]$ .

It is a function defined on a 2-dimensional space, whose components are coded on 12 bits. The same 4 encodings as for  $F1$  are tested. The performances are displayed in figure 14.

Once again, a lot of points have fitness values very close to the optimum, so the ratio of populations containing it is more or less random. In fact, it would require far more than 4 digits to distinguish the *Maximum* performances.

The *Mean* performances follow, in order, the predictions of the bound, except for the comparison between *Code2* and *Code3*, for which the bound increase is the lowest, and the performances are roughly identical.

## 6.3 Function M2.

This function was introduced in the first experiments (section 4).

Two encodings are tested:

- *Code1*: a classical unsigned integer encoding mapped to  $[0,1]$
- *Code2*: Gray version of *Code1*.

Here the Gray encoding induces an increase of the bound and a decrease of the performances as predicted (see figure 15).

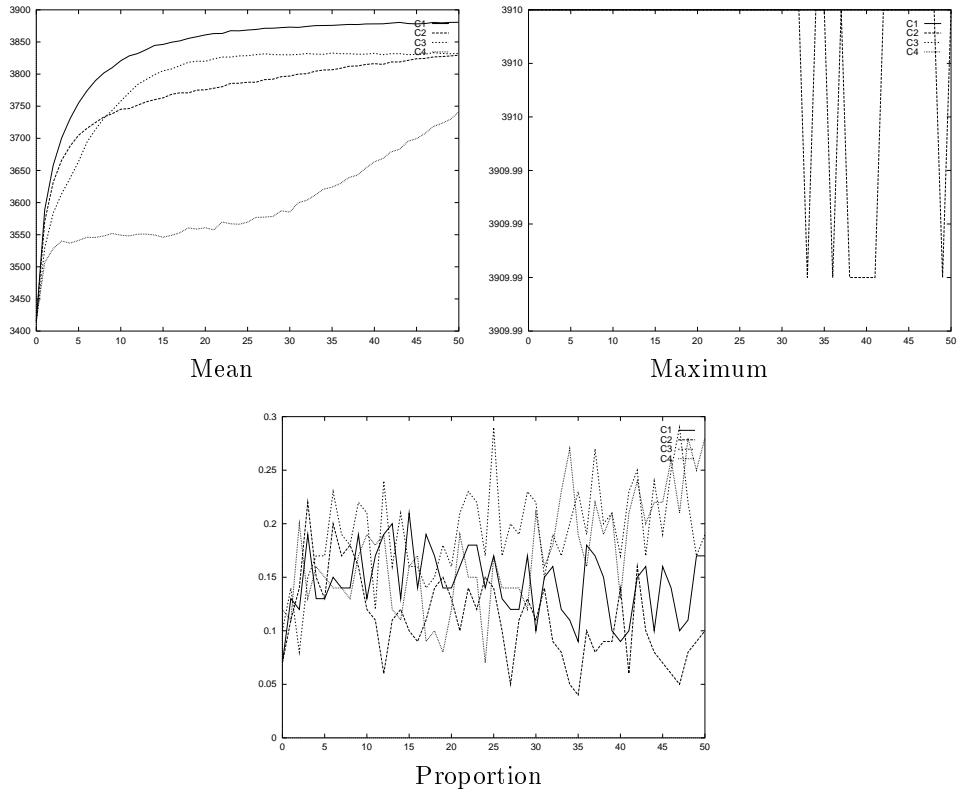
## 6.4 Function M7.

This function extracted from [4], is massively multimodal and deceptive. It is composed of sub-functions defined on 6 bits, which reach their maximum value for two mirror strings. Here we used 3 of them ( $l = 18$ ).

Two encodings are tested:

- *Code1*: the classical encoding.
- *Code2*: Gray version of *Code1*.

Here the Gray encoding induces a decrease of the bound and an increase of the performances, as predicted (see figure 16).



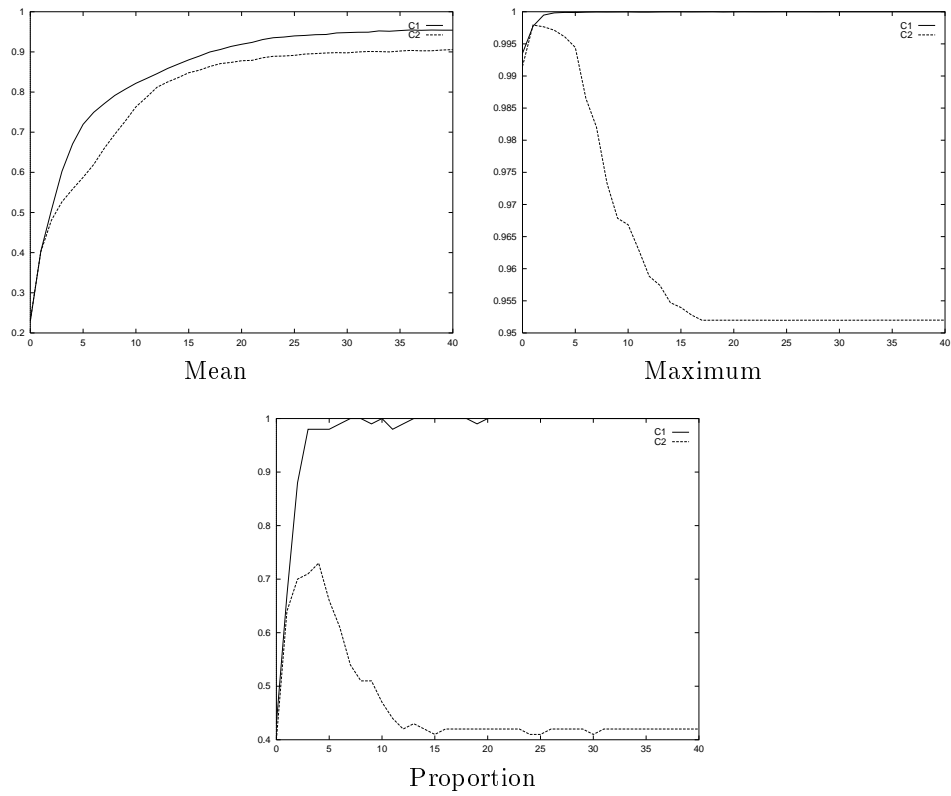
$l$	$N$	$Gen$	$Runs$	$p_c$	$p_m$
24	512	50	100	1.0	0.005

GA parameters

<i>Encoding</i>	<i>Code1</i>	<i>Code2</i>	<i>Code3</i>	<i>Code4</i>
<i>Bound</i>	13350	15610	16570	19070

Bound values

Figure 14: Tests on function F2.



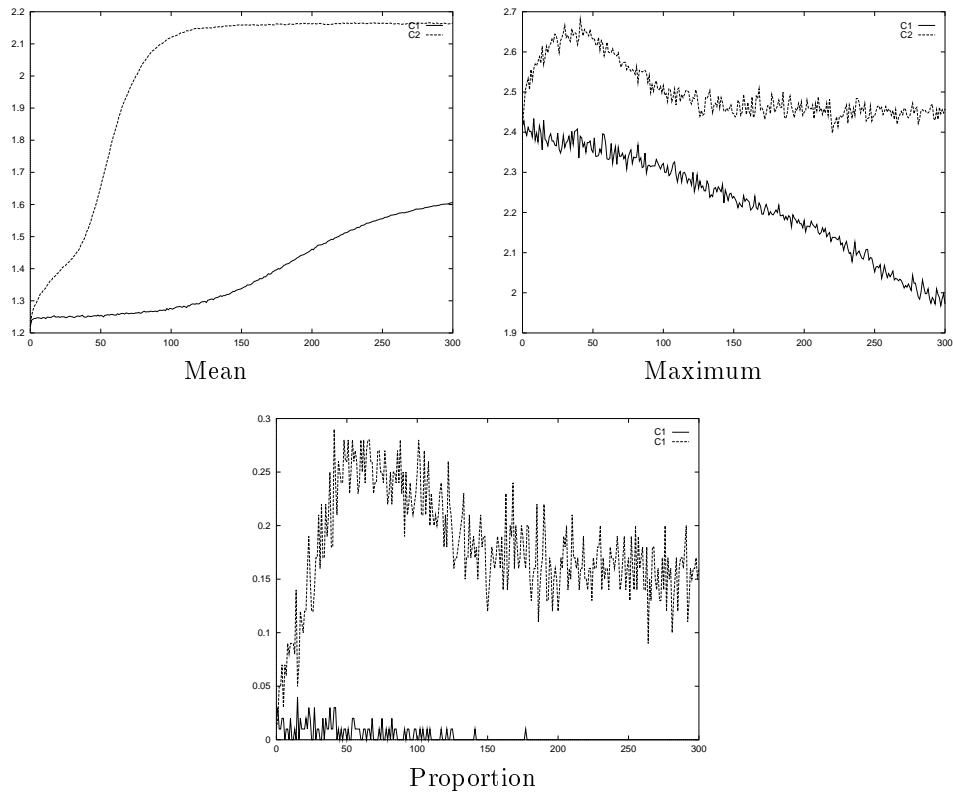
$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
16	200	40	100	1.0	0.005

GA parameters

$B_c$	$B_m$
4.86	5.76

Bound values

Figure 15: Tests on function M2.



$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
18	512	300	100	1.0	0.005

GA parameters

$B_c$	$B_m$
18.9	13.8

Bound values

Figure 16: Tests on function M7.

## 6.5 Function EPI6.

This function has been defined in the first experiments (section 4).

Two encodings are tested :

- *Code1* : the classical encoding.
- *Code2* : Gray version of *Code1*.

Here the Gray encoding induces an increase of the bound and a decrease of the performances, as predicted (see figure 17).

## 6.6 Function W20.

This function was introduced in the first experiments (section 4).

Two encodings are tested :

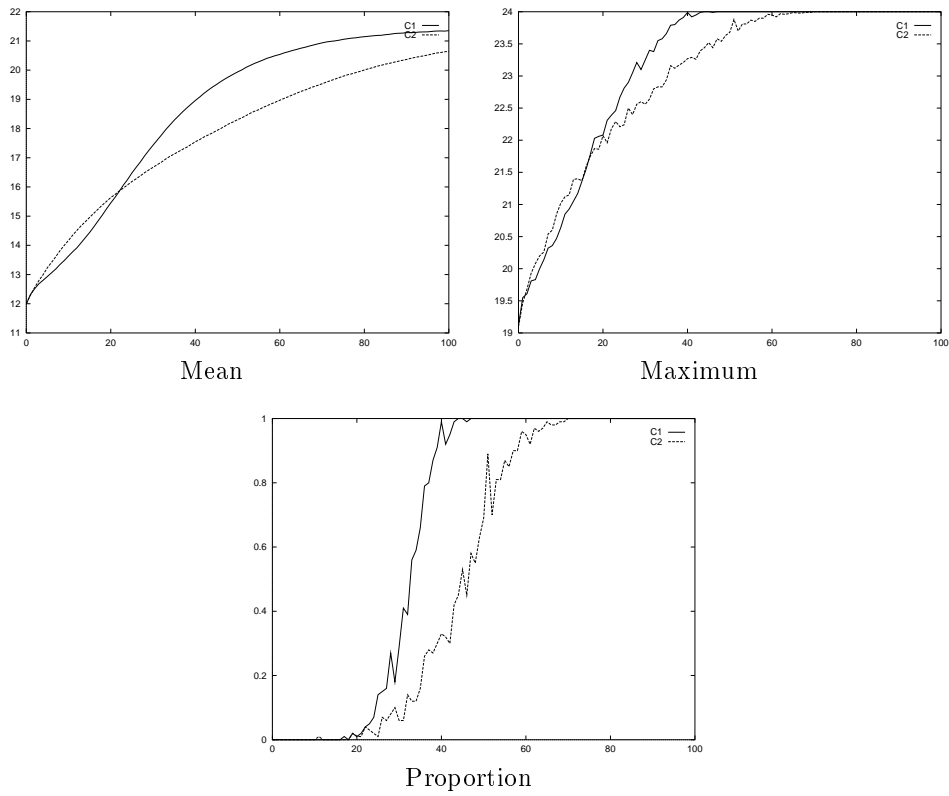
- *Code1* : the classical encoding.
- *Code2* : Gray version of *Code1*.

Here the bound slightly increases but the performances of that GA seem to be slightly better (see figure 18).

## 6.7 Function FBM50.

This fitness function is a sampling of a Fractional Brownian Motion [7] of Hölder exponent  $h = 0.5$ . The encodings are the same as for *M2*.

The decrease of the bound is very small, compared to the previous tests, (and except for the *Mean* at the end of the 50 generations), the performances slightly increase as predicted (see figure 19).



$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
24	512	100	100	1.0	0.005

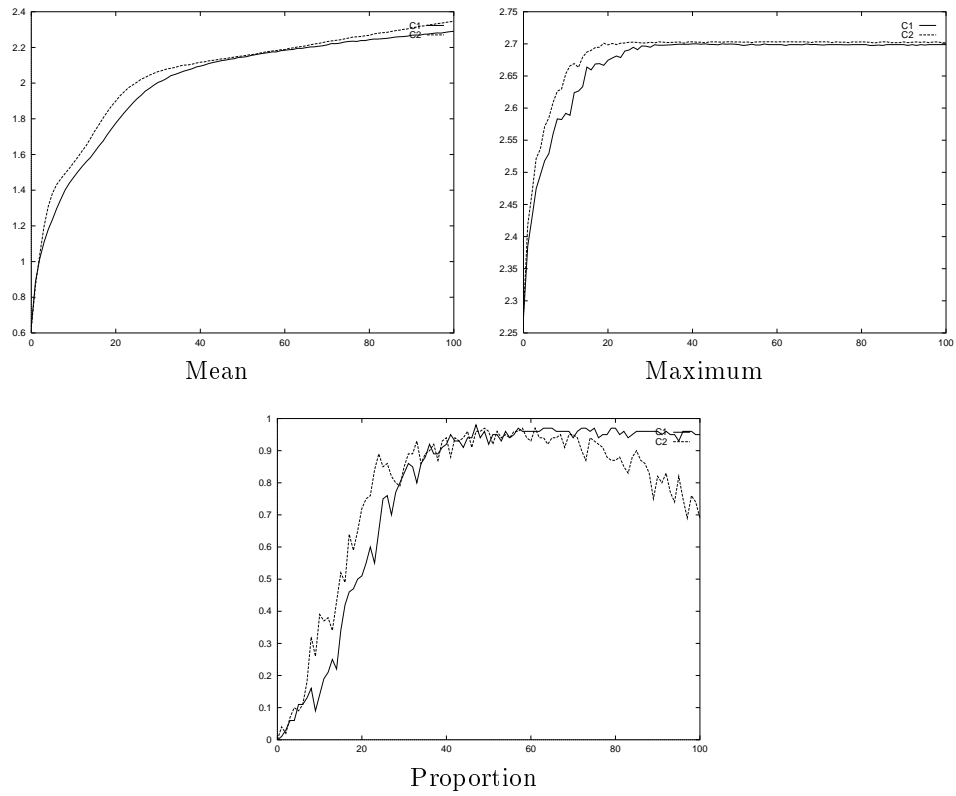
GA parameters

$B_c$	$B_m$
43.8	57.1

Bound values

Figure 17: Tests on function EPI6.





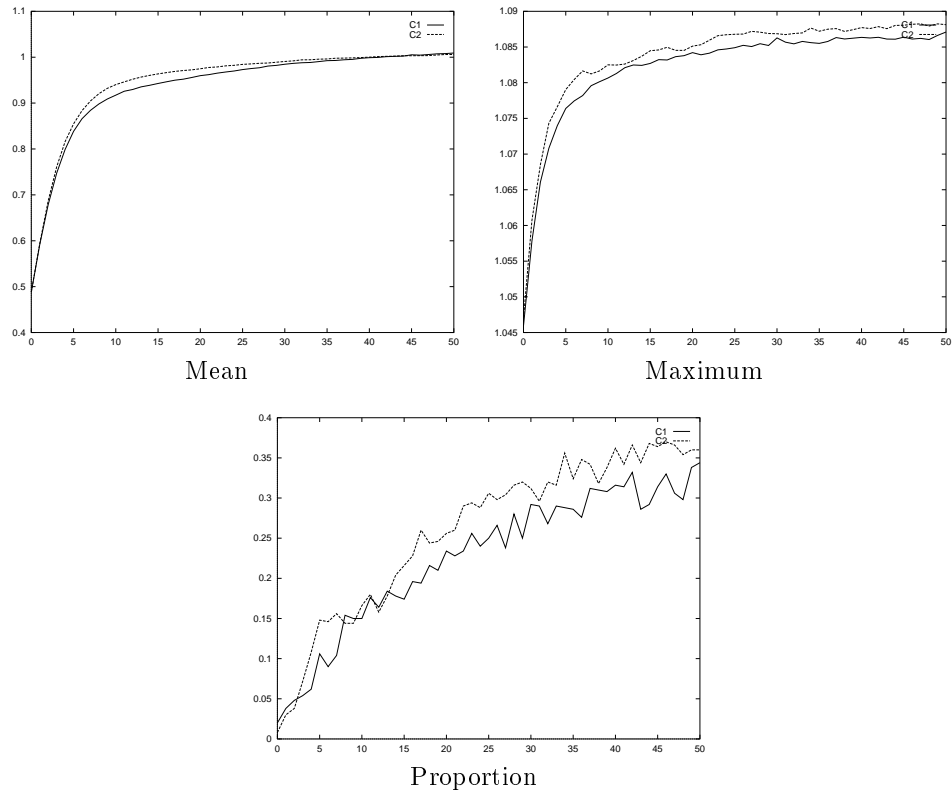
$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
16	512	100	100	1.0	0.005

GA parameters

$B_c$	$B_m$
21.6	22.65

Bound values

Figure 18: Performances on W20.



$l$	$N$	$Gen$	$Runs$	$P_c$	$p_m$
12	64	50	500	1.0	0.005

GA parameters

$B_c$	$B_m$
4.658	4.609

Bound values

Figure 19: Tests on function FBM50.

## 7 Conclusions.

The tests presented in section 6 show that the bound calculated from the bitwise regularity coefficients is a quite reliable tool to compare encodings as long as its variations are significant enough: when the bound variations are high, the GA behaves according to the predictions, when they are low (as for *Code2* to *Code3* of functions *F2*, for functions *W1.8* and *FBM50*) the GA behavior is less predictable.

These limitations can be explained in many ways. The one that seems to us the most appropriate is of the same nature as the *Static Building Blocks Hypothesis*, pointed out in [13]. If we consider cautiously the calculation of  $f'$ , which is the basis of the *static deception* analysis, we note that it is assumed that each allele is equally represented at each position. This viewpoint (detailed in appendix F), should be considered with care in order to continue the work presented here and suggests that a dynamical modelization of the GA behavior would be more appropriate. The nonuniform Walsh-schema transform [2] could be the basis of such an improvement.

## References

- [1] A. Agapie. Genetic algorithms : Minimal conditions for convergence. In *Artificial Evolution, European Conference, AE 97, Nimes, France, October 1997*,. Springer Verlag, 1997.
- [2] Clayton L. Bridges and David E. Goldberg. The nonuniform walsh-schema transform. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms.*, pages 13–22. Morgan Kaufmann publishers, San Mateo, 1991.
- [3] R. Cerf. *Artificial Evolution, European Conference, AE 95, Brest, France, September 1995, Selected papers*, volume Lecture Notes in Computer Science 1063, chapter Asymptotic convergence of genetic algorithms, pages 37–54. Springer Verlag, 1995.
- [4] K. Deb David E. Goldberg and J. Horn. Massive multimodality, deception and genetic algorithms. In B. Manderick R. Männer, editor, *Parallel Problem Solving from Nature (PPSN)*, volume 2, pages 37–46. Amsterdam : North-Holland, 1992.
- [5] Yuval Davidor. Epistasis variance : A viewpoint on ga-hardness. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms.*, pages 23–35. Morgan Kaufmann publishers, San Mateo, 1991.
- [6] Thomas E. Davis and Jose C. Principe. A Simulated Annealing Like Convergence Theory for the Simple Genetic Algorithm . In *Proceedings of the Fourth International Conference on Genetic Algorithm*, pages 174–182, 1991. 13-16 July.
- [7] K.J. Falconer. *Fractal Geometry : Mathematical Foundation and Applications*. John Wiley & Sons, 1990.
- [8] David E. Goldberg. Genetic algorithms and walsh functions: Part i, a gentle introduction. In *Complex Systems*, volume 3, pages 123–152. 1989.
- [9] David E. Goldberg. Genetic algorithms and walsh functions: Part ii, deception and its analysis. In *Complex Systems*, volume 3, pages 153–171. 1989.
- [10] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [11] David E. Goldberg. Construction of high-order deceptive functions using low-order walsh coefficients. IlliGAL Report 90002, University of Illinois at Urbana-Champaign, Urbana, IL 61801, December 1990.
- [12] David E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J.J. Grefenstette, editor, *Genetic Algorithms and their Applications*, pages 41–49. Lawrence Erlbaum Associates, Hillsdale, New-Jersey, 1987.

- 
- [13] John J. Grefenstette. Deception considered harmful. In L.D. Whitley, editor, *Foundations of Genetic Algorithms 2.*, pages 75–91. Morgan Kaufmann publishers, San Mateo, 1992.
- [14] John Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor, University of Michigan Press, 1975.
- [15] K.A. De Jong. *An analysis of the behavior of a class of genetic adaptative systems*. PhD thesis, University of Michigan, 1975.
- [16] J. Juliany and M. D. Vose. The genetic algorithm fractal. *Evolutionary Computation*, 2(2):165–180, 1994.
- [17] Evelyne Lutton and Jacques Lévy-Véhel. Some remarks on the optimization of hölder functions with genetic algorithms. Technical Report 2627, Institut National de Recherche en Informatique et en Automatique, Rocquencourt, France, 1995.
- [18] Andrew J. Mason. Partition coefficients, static deception and deceptive problems for non-binary alphabets. In R. K. Belew and L.B. Booker, editor, *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann publishers, San Mateo, 1991.
- [19] S. Rochet, G. Venturini, M. Slimane, and E.M. El Kharoubi. *Artificial Evolution, European Conference, AE 97, Nimes, France, October 1997, Selected papers*, volume Lecture Notes in Computer Science, chapter A critical and empirical study of epistasis measures for predicting GA performances : a summary. Springer Verlag, 1997.
- [20] G. Rudolph. Asymptotical convergence rates of simple evolutionary algorithms under factorizing mutation distributions. In *Artificial Evolution, European Conference, AE 97, Nimes, France, October 1997.*, Springer Verlag, 1997.
- [21] Gilbert Syswerda. Uniform crossover in genetic algorithms. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann publishers, San Mateo, 1989.
- [22] M.D. Vose. Formalizing genetic algorithms. In *Genetic Algorithms, Neural Networks and Simulated Annealing Applied to Problems in Signal and Image processing*. The institute of Electrical and Electronics Engineers Inc., 8th-9th May 1990. Kelvin Conference Centre, University of Glasgow.

## A Bounding $|f - f'|$ with the $C_q$ coefficients.

From the definition of the Haar functions, we can write for  $j = 2^q + m$  :

$$h_j = \frac{1}{2^{l-q}} \left[ \sum_{x=(2m) * 2^{l-q-1}}^{(2m+1) * 2^{l-q-1}-1} f(x) - \sum_{x=(2m+1) * 2^{l-q-1}}^{(2m+2) * 2^{l-q-1}-1} f(x) \right]$$

Furthermore, we notice that if  $x \in \{(2m) * 2^{l-q-1}, \dots, (2m+1) * 2^{l-q-1} - 1\}$ , its binary profile belongs to the schema  $S_{2^q+m}^+$  :

$$S_{2^q+m}^+ = m_{q-1} \dots m_0 0 * \dots *$$

Where  $m_{q-1} \dots m_0$  denotes the binary profile of  $m$  :  $m = \sum_{i=0}^{q-1} m_i * 2^i$ .

In the same way, if  $x \in \{(2m+1) * 2^{l-q-1}, \dots, (2m+2) * 2^{l-q-1} - 1\}$ , then its binary profile belongs to the schema  $S_{2^q+m}^-$  :

$$S_{2^q+m}^- = m_{q-1} \dots m_0 1 * \dots *$$

We notice that for each element  $x$  in  $S_{2^q+m}^+$  corresponds a  $x'_{l-q-1}$  in  $S_{2^q+m}^-$  differing only with respect to position  $(l-q-1)$  (0 for  $x$  and 1 and  $x'_{l-q-1}$ ). Then we can write :

$$h_j = \frac{1}{2^{l-q}} \sum_{x \in S_{2^q+m}^+} f(x) - f(x')$$

And from the  $(C_q)$  definition, we get :

$$\begin{aligned} \forall j = 2^q + m, \quad |h_j| &\leq \frac{1}{2^{l-q}} 2^{l-q-1} C_q \\ \forall j = 2^q + m, \quad |h_j| &\leq \frac{C_q}{2} \end{aligned}$$

From the expression of the corrected Haar coefficients, given in [17] :

$$\begin{aligned} h'_{2^q+m} = & h_{2^q+m} \left[ 1 - \frac{p_c}{l-1} \left( 1 + \frac{1 + (q-2)2^q}{2^q} \right) - 2p_m \left( 1 + \frac{q}{2} \right) \right] \\ & - \frac{p_c}{2^q(l-1)} \sum_{u=0}^{q-1} (1 - 2^{u+1}) \sum_{r=0}^{2^{q-u}-1} h_{2^q + \sum_{t=0}^{u-1} m_t 2^t + (1-m_u)2^u + r 2^{(u+1)}} \\ & - p_m \sum_{t=0}^{q-1} h_{2^q+m+(1-2m_t)2^t} \end{aligned}$$

we obtain, after a few calculations :

$$|h_{2^q+m} - h'_{2^q+m}| \leq C_q \left[ \frac{p_c}{2^q(l-1)}(1 + 2^q(q-1)) + p_m(1+q) \right]$$

From [17], we know that :

$$|f(x) - f'(x)| \leq \sum_{q=0}^{l-1} |h_{2^q+m_x} - h'_{2^q+m_x}|$$

With  $m_x = E(\frac{x}{2^{l-q}})$ . We finally obtain :

$$|f(x) - f'(x)| \leq \frac{p_c}{l-1} \sum_{q=0}^{l-1} C_q \left( \frac{1 + 2^q(q-1)}{2^q} \right) + p_m \sum_{q=0}^{l-1} C_q * (q+1)$$

## B Encoding change: bits permutation.

We consider here the class of permutations  $\sigma$  on the integers  $\{0, \dots, (l-1)\}$  and note  $\sigma(x)$  the integer whose binary profile is the permuted binary profile of  $x$ .

### B.1 Implications for the Haar and Walsh basis.

We define  $H_{2^q+m}^\sigma$  as :

$$H_{2^q+m}^\sigma(x) = \begin{cases} +1 & \text{for } \sigma^{-1}(x) \in S_{2^q+m}^+ \\ -1 & \text{for } \sigma^{-1}(x) \in S_{2^q+m}^- \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

(See appendix A for the definitions of  $S_{2^q+m}^+$  and  $S_{2^q+m}^-$ .) Then we have :

$$H_{2^q+m}^\sigma(x) = H_{2^q+m}(\sigma^{-1}(x))$$

The  $(H_j^\sigma)_{j \in \{0 \dots 2^l-1\}}$  functions also form a basis for the functions defined on  $\Omega^l$  :

$$\begin{aligned} \sum_{x \in \Omega^l} H_i(x) H_j(x) &= \begin{cases} 2^q & \text{if } i = j = 2^q + m \\ 0 & \text{otherwise} \end{cases} \\ \Rightarrow \sum_{x \in \Omega^l} H_i(\sigma^{-1}(x)) H_j(\sigma^{-1}(x)) &= \begin{cases} 2^q & \text{if } i = j = 2^q + m \\ 0 & \text{otherwise} \end{cases} \\ \Rightarrow \sum_{x \in \Omega^l} H_i^\sigma(x) H_j^\sigma(x) &= \begin{cases} 2^q & \text{if } i = j = 2^q + m \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The associated coefficients are :

$$h_{2^q+m}^\sigma = \frac{1}{2^{l-q}} \sum_{x=0}^{2^l-1} f(x) H_j^\sigma(x)$$

And in the same way as we obtained (see appendix A)  $h_{2^q+m} \leq \frac{C_q}{2}$ , we also get :

$$h_{2^q+m}^\sigma \leq \frac{C_{\sigma^{-1}(q)}}{2}$$

For the Walsh basis, we directly have :

$$\begin{aligned} \Psi_j(\sigma^{-1}(x)) &= \prod_{t=0}^{l-1} (-1)^{\sigma^{-1}(x)_t j_t} \\ &= \prod_{t=0}^{l-1} (-1)^{x_t \sigma(j)_t} \\ &= \Psi_{\sigma(j)}(x) \end{aligned}$$



## B.2 Expression of Haar- $\sigma$ coefficients as a function of the Walsh coefficients and conversely.

We first establish the expression of the Haar basis in the Walsh basis. From [17], we have :

$$H_{2^q+m}(\sigma^{-1}(x)) = \frac{1}{2^q} \sum_{k=0}^{2^q-1} (-1)^{\sum_{t=0}^{l-1} m_t k_t} \Psi_{2^{l-q-1}+k2^{l-q}}(\sigma^{-1}(x))$$

$$H_{2^q+m}^\sigma(x) = \frac{1}{2^q} \sum_{k=0}^{2^q-1} (-1)^{\sum_{t=0}^{l-1} m_t k_t} \Psi_{\sigma(2^{l-q-1}+k2^{l-q})}(x)$$

Conversely, from [17], if we note :

$$i = 2^{l-q-1}(1 + 2k) \quad \text{et} \quad k \in [0 ; 2^q - 1]$$

we get :

$$\Psi_i(\sigma^{-1}(x)) = \sum_{m=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} k_t m_t} H_{2^q+m}(\sigma^{-1}(x))$$

$$\Psi_{\sigma(i)}(x) = \sum_{m=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} k_t m_t} H_{2^q+m}^\sigma(x)$$

Following the same calculations as presented in [17] (appendix C), we finally get identical expressions, up to the subscript  $\sigma$  :

$$\begin{aligned} h_j^\sigma &= \sum_{k=0}^{2^q-1} w_{\sigma(2^{l-q-1}+(1+2k))} (-1)^{\sum_{t=0}^{q-1} m_t k_t} \\ w_{\sigma(i)} &= \frac{1}{2^q} \sum_{m=0}^{2^q-1} h_{2^q+m}^\sigma (-1)^{\sum_{t=0}^{q-1} m_t k_t} \end{aligned}$$

### B.3 Implications on adjusted coefficients and the bound on $|f - f'|$ .

As in [17](appendix D), we may establish the adjusted coefficients  $h_j^{\sigma'}$  as a functions of the  $h_j^{\sigma}$  coefficients. The calculations are identical (up to the  $\sigma$  subscript) until the following expression :

$$h_{2^q+m}^{\sigma'} = \frac{1}{2^q} \sum_{k=0}^{2^q-1} \left[ 1 - p_c \frac{\delta(\sigma(2^{l-q-1}(1+2k)))}{l-1} - 2p_m O(\sigma(2^{l-q-1}(1+2k))) \right] \sum_{m'=0}^{2^q-1} h_{2^q+m'}^{\sigma} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t}$$

We remind that  $\delta(i)$  denotes the distance between the first and the last '1' in the binary expression of  $j$ , and  $O(j)$  the number of '1'. Then, in order to continue the calculations in a same manner as in [17], the following properties are sufficient :

$$\begin{aligned} \delta(\sigma(i)) &= \delta(i) \\ O(\sigma(i)) &= O(i) \end{aligned}$$

For every  $i$ , the number of '1' bits is invariant for any permutation, so the second equality is always true. But the distance  $\delta(i)$  is preserved only if the permutation preserves the relative order. Such a permutation is the permutation that reverse the order ( $\sigma(l-1) = 0$ ,  $\sigma(l-2) = 1$ ,  $\dots$ ,  $\sigma(0) = l-1$ ). Consequently, the theorem 3 only still holds if we apply this mirror permutation.

## C Bound for the $C_q$ coefficients with $h$ and $k$ .

One can make the assumption that any fitness function may be defined with the help of a function  $F$  such as :

$$\forall x \in \Omega^l, \quad f(x) = F\left(\frac{\text{integer}(x)}{2^l}\right)$$

$$\text{With} \quad \text{integer}(x) = \sum_{i=0}^{l-1} x_i * 2^i$$

and  $F : [0, 1] \rightarrow \mathbb{R}^+$  Hölder with exponent  $h$  and constant  $k$ .

Recall that :

$$C_q = \sup_{x \in \Omega^l} \{|f(x) - f(x'_{l-q-1})|\}$$

Moreover  $\forall x \in \Omega^l$  :

$$|f(x) - f(x'_{l-q-1})| = \left| F\left(\frac{\text{integer}(x)}{2^l}\right) - F\left(\frac{\text{integer}(x'_{l-q-1})}{2^l}\right) \right|$$

And if we note  $X_q = \min\{\text{integer}(x), \text{integer}(x'_{l-q-1})\}$  we have :

$$\begin{aligned} |f(x) - f(x'_{l-q-1})| &= \left| F\left(\frac{X_q}{2^l}\right) - F\left(\frac{X_q + 2^{l-q-1}}{2^l}\right) \right| \\ &= \left| F\left(\frac{X_q}{2^l}\right) - F\left(\frac{X_q}{2^l} + 2^{-(q+1)}\right) \right| \end{aligned}$$

$F$  being a Hölder function of exponent  $h$  and constant  $k$ , we can write :

$$\begin{aligned} |f(x) - f(x'_{l-q-1})| &\leq k 2^{-(q+1)h} \\ \sup_{x \in \Omega^l} \{|f(x) - f(x'_{l-q-1})|\} &\leq k * 2^{-(q+1)h} \end{aligned}$$

Thus :

$$C_q \leq k 2^{-(q+1)h}$$

## D Bound for $|f - f'|$ with the $C_q$ coefficients for uniform crossover.

We first establish the new bound on  $|h_j^{\sigma'} - h_j^{\sigma'}|$ . From the appendix B.2, for all permutation  $\sigma$ , if we note ( $j = 2^q + m$ ) and ( $i = 2^{l-q-1}(1 + 2k)$ ), we have:

$$\begin{aligned} h_j^{\sigma'} &= \sum_{k=0}^{2^q-1} w'_{\sigma(2^{l-q-1}+(1+2k))} (-1)^{\sum_{t=0}^{q-1} m_t k_t} \\ w'_{\sigma(i)} &= \frac{1}{2^q} \sum_{m=0}^{2^q-1} h_{2^q+m}^{\sigma'} (-1)^{\sum_{t=0}^{q-1} m_t k_t} \end{aligned}$$

For the uniform crossover the adjusted Walsh coefficients are (that do not depend any more on  $\delta(j)$ ):

$$w'_j = w_j \left[ 1 - p_c \left( 1 - \left( \frac{1}{2} \right)^{O(j)-1} \right) - 2p_m O(j) \right]$$

Then we get :

$$\begin{aligned} h_{2^q+m}^{\sigma'} &= \sum_{k=0}^{2^q-1} w_{\sigma(2^{l-q-1}+(1+2k))} (-1)^{\sum_{t=0}^{q-1} m_t k_t} \left[ 1 - p_c \left( 1 - \left( \frac{1}{2} \right)^{O(\sigma(2^{l-q-1}+(1+2k)))-1} \right) \right. \\ &\quad \left. - 2p_m O(\sigma(2^{l-q-1} + (1 + 2k))) \right] \end{aligned}$$

Furthermore, we have :  $O(\sigma(2^{l-q-1} + (1 + 2k))) = O(2^{l-q-1} + (1 + 2k)) = O(k) + 1$

$$\begin{aligned} h_{2^q+m}^{\sigma'} &= \sum_{k=0}^{2^q-1} \sum_{m'=0}^{2^q-1} h_{2^q+m'}^{\sigma} \frac{(-1)^{\sum_{t=0}^{q-1} (m_t+m'_t) k_t}}{2^q} \left[ 1 - p_c \left( 1 - \left( \frac{1}{2} \right)^{O(k)} \right) - 2p_m (O(k) + 1) \right] \\ h_{2^q+m}^{\sigma'} &= \frac{1}{2^q} \sum_{k=0}^{2^q-1} \left[ 1 - p_c \left( 1 - \left( \frac{1}{2} \right)^{O(k)} \right) - 2p_m (O(k) + 1) \right] \sum_{m'=0}^{2^q-1} h_{2^q+m'}^{\sigma} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t) k_t} \\ h_{2^q+m}^{\sigma'} &= \frac{1}{2^q} \sum_{m'=0}^{2^q-1} h_{2^q+m'}^{\sigma} \sum_{k=0}^{2^q-1} \left[ 1 - p_c \left( 1 - \left( \frac{1}{2} \right)^{O(k)} \right) - 2p_m (O(k) + 1) \right] (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t) k_t} \end{aligned}$$

And finally :

$$\begin{aligned} h_{2^q+m}^{\sigma'} &= h_{2^q+m}^{\sigma} (1 - p_c - 2p_m) \\ &\quad + \frac{p_c}{2^q} \sum_{m'=0}^{2^q-1} h_{2^q+m'}^{\sigma} \sum_{k=0}^{2^q-1} \left( \frac{1}{2} \right)^{O(k)} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t) k_t} \end{aligned}$$

$$-\frac{2p_m}{2^q} \sum_{m'=0}^{2^q-1} h_{2^q+m'}^\sigma \sum_{k=0}^{2^q-1} O(k) (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t}$$

The term depending on  $p_m$  is calculated in [17]:

$$h_{2^q+m}^\sigma (-2p_m(1 + \frac{q}{2})) - p_m \sum_{t=0}^{q-1} h_{2^q+m+(1-2m_t)2^t}^\sigma$$

If we note :

$$P^q(m, m') = \sum_{k=0}^{2^q-1} \left(\frac{1}{2}\right)^{O(k)} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t}$$

Then :

$$\begin{aligned} P^{(q+1)}(m, m') &= \sum_{k=0}^{2^{q+1}-1} \left(\frac{1}{2}\right)^{\sum_{t=0}^q k_t} (-1)^{\sum_{t=0}^q (m_t+m'_t)k_t} \\ &= \sum_{k=0}^{2^{q+1}-1} \left(\frac{1}{2}\right)^{k_q} \left(\frac{1}{2}\right)^{\sum_{t=0}^{q-1} k_t} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t} (-1)^{(m_q+m'_q)k_q} \\ &= \sum_{k=0}^{2^q-1} \left(\frac{1}{2}\right)^0 \left(\frac{1}{2}\right)^{\sum_{t=0}^{q-1} k_t} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t} (-1)^{(m_q+m'_q)*0} \\ &\quad + \sum_{k=2^q}^{2^{q+1}-1} \left(\frac{1}{2}\right)^1 \left(\frac{1}{2}\right)^{\sum_{t=0}^{q-1} k_t} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t} (-1)^{(m_q+m'_q)*1} \\ &= \sum_{k=0}^{2^q-1} \left(\frac{1}{2}\right)^{\sum_{t=0}^{q-1} k_t} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t} \\ &\quad + \sum_{k=2^q}^{2^{q+1}-1} \frac{1}{2} \left(\frac{1}{2}\right)^{\sum_{t=0}^{q-1} k_t} (-1)^{\sum_{t=0}^{q-1} (m_t+m'_t)k_t} * (-1)^{(m_q+m'_q)} \\ &= P^q(m, m') + \frac{1}{2} (-1)^{(m_q+m'_q)} P^q(m, m') \\ P^{(q+1)}(m, m') &= P^q(m, m') \left(1 + \frac{1}{2} (-1)^{(m_q+m'_q)}\right) \end{aligned}$$

We also have :

$$P^1(m, m') = 1 + \frac{1}{2} (-1)^{(m_0+m'_0)}$$

Thus :

$$\begin{aligned}
P^q(m, m') &= \prod_{t=0}^{q-1} \left( 1 + \frac{1}{2}(-1)^{(m_t+m'_t)} \right) \\
&= \prod_{t=0/m_t \neq m'_t}^{q-1} \frac{1}{2} \prod_{t=0/m_t = m'_t}^{q-1} 3 \frac{1}{2} \\
&= \prod_{t=0}^{q-1} \frac{1}{2} \prod_{t=0/m_t = m'_t}^{q-1} 3
\end{aligned}$$

If we note  $d_H(m, m')$  the Hamming distance between  $m$  and  $m'$  ( $d_H(m, m') = \sum_{t=0/m_t \neq m'_t}^{q-1} 1$ ), we finally have :

$$P^q(m, m') = \frac{1}{2^q} 3^{d_H(m, m')}$$

The expression of  $h_{2^q+m}^{\sigma'}$  thus becomes :

$$h_{2^q+m}^{\sigma'} = h_{2^q+m}^{\sigma} (1 - p_c - 2p_m(1 + \frac{q}{2})) + \frac{p_c}{2^q} \sum_{m'=0}^{2^q-1} h_{2^q+m'}^{\sigma} \frac{1}{2^q} 3^{d_H(m, m')} - p_m \sum_{t=0}^{q-1} h_{2^q+m+(1-2m_t)2^t}^{\sigma}$$

We now have to bound  $|h_{2^q+m}^{\sigma} - h_{2^q+m}^{\sigma'}|$ :

$$\begin{aligned}
|h_{2^q+m}^{\sigma} - h_{2^q+m}^{\sigma'}| &= |h_{2^q+m}^{\sigma} (p_c + 2p_m(1 + \frac{q}{2})) - \frac{p_c}{2^q} \sum_{m'=0}^{2^q-1} h_{2^q+m'}^{\sigma} \frac{1}{2^q} 3^{d_H(m, m')} \\
&\quad + p_m \sum_{t=0}^{q-1} h_{2^q+m+(1-2m_t)2^t}^{\sigma} | \\
|h_{2^q+m}^{\sigma} - h_{2^q+m}^{\sigma'}| &\leq |h_{2^q+m}^{\sigma}| (p_c + 2p_m(1 + \frac{q}{2})) + \frac{p_c}{2^q} \sum_{m'=0}^{2^q-1} |h_{2^q+m'}^{\sigma}| \frac{1}{2^q} 3^{d_H(m, m')} \\
&\quad + p_m \sum_{t=0}^{q-1} |h_{2^q+m+(1-2m_t)2^t}^{\sigma}|
\end{aligned}$$

Now we know from the appendix B.2 that :

$$\forall n \in \{0, \dots, 2^q - 1\}, \quad |h_{2^q+n}^{\sigma}| \leq \frac{C_{\sigma^{-1}(q)}}{2}$$

Thus :

$$|h_{2^q+m}^{\sigma} - h_{2^q+m}^{\sigma'}| \leq \frac{C_{\sigma^{-1}(q)}}{2} (p_c + 2p_m(1 + \frac{q}{2})) + \frac{p_c}{2^q} \frac{C_{\sigma^{-1}(q)}}{2} \sum_{m'=0}^{2^q-1} \frac{1}{2^q} 3^{d_H(m, m')} + p_m \sum_{t=0}^{q-1} \frac{C_{\sigma^{-1}(q)}}{2}$$

Setting  $s = 3^{d_H(m, m')}$ , we may write :

$$\begin{aligned} \sum_{m'=0}^{2^q-1} \frac{1}{2^q} 3^{d_H(m, m')} &= \frac{1}{2^q} \sum_{s=0}^q \binom{q}{s} 3^s 1^{(q-s)} \\ &= \frac{1}{2^q} (3 + 1)^q \\ &= 2^q \end{aligned}$$

Thus :

$$|h_{2^q+m}^\sigma - h_{2^q+m}^{\sigma'}| \leq C_{\sigma^{-1}(q)} [p_c + p_m(1 + q)]$$

Finally, reporting this to  $|f - f'|$  :

$$|f(x) - f'(x)| \leq \sum_{q=0}^{l-1} |h_{2^q+m_x}^\sigma - h_{2^q+m_x}^{\sigma'}|$$

$$|f(x) - f'(x)| \leq p_c \sum_{q=0}^{l-1} C_{\sigma^{-1}(q)} + p_m \sum_{q=0}^{l-1} C_{\sigma^{-1}(q)} (1 + q)$$

## E Bound for $|f - f'|$ with the $(h, k)$ coefficients for uniform crossover.

Under uniform crossover, we have (see appendix D):

$$\begin{aligned} |h_{2^{q+m}} - h'_{2^{q+m}}| &\leq |h_{2^{q+m}}^\sigma| \left( p_c + 2p_m \left(1 + \frac{q}{2}\right) \right) + \frac{p_c}{2^q} \sum_{m'=0}^{2^q-1} |h_{2^{q+m'}}| \frac{1}{2^q} 3^{d_H(m, m')} \\ &\quad + p_m \sum_{t=0}^{q-1} |h_{2^{q+m+(1-2m_t)2^t}}| \end{aligned}$$

Furthermore from [17] we know that :

$$\forall j = 2^q + m, \quad |h_{2^{q+m}}| \leq \frac{k}{2} 2^{-h(q+1)}$$

Then :

$$\begin{aligned} |h_{2^{q+m}} - h'_{2^{q+m}}| &\leq \frac{k}{2} 2^{-h(q+1)} \left( p_c + 2p_m \left(1 + \frac{q}{2}\right) \right) + p_c \frac{k}{2} 2^{-h(q+1)} \frac{1}{2^q} \sum_{m'=0}^{2^q-1} \frac{1}{2^q} 3^{d_H(m, m')} \\ &\quad + p_m \sum_{t=0}^{q-1} \frac{k}{2} 2^{-h(q+1)} \end{aligned}$$

Now (see appendix D):

$$\sum_{m'=0}^{2^q-1} \frac{1}{2^q} 3^{d_H(m, m')} = 2^q$$

Then :

$$\begin{aligned} |h_{2^{q+m}} - h'_{2^{q+m}}| &\leq \frac{k}{2} 2^{-h(q+1)} (p_c + 2p_m(1 + \frac{q}{2})) + p_c \frac{k}{2} 2^{-h(q+1)} + p_m \frac{k}{2} 2^{-h(q+1)} * q \\ |h_{2^{q+m}} - h'_{2^{q+m}}| &\leq k \left[ 2^{-h(q+1)} (p_c + p_m(1 + q)) \right] \end{aligned}$$

And finally :

$$\begin{aligned} |f(x) - f'(x)| &\leq \sum_{q=0}^{l-1} |h_{2^{q+m_x}}^\sigma - h_{2^{q+m_x}}^{\sigma'}| \\ |f(x) - f'(x)| &\leq k \sum_{q=0}^{l-1} 2^{-h(q+1)} (p_c + p_m(1 + q)) \\ |f(x) - f'(x)| &\leq k * p_c * \sum_{q=0}^{l-1} 2^{-h(q+1)} + k * p_m * \sum_{q=0}^{l-1} 2^{-h(q+1)} (1 + q) \\ |f(x) - f'(x)| &\leq k \left[ p_c \frac{2^{-h}(2^{-hl} - 1)}{2^{-h} - 1} + p_m \frac{2^{-h}}{(2^{-h} - 1)^2} (1 + 2^{-hl}(l2^{-h} - l - 1)) \right] \end{aligned}$$



## F The adjusted Walsh coefficients calculation revisited.

We focus here on the calculation of the adjusted Walsh coefficients that define the functions  $f'$  [9]. We recall that for a schema  $h$ , the schema average fitness is :

$$f(h) = \sum_{j \in J(h)} w_j \psi_j(\beta(h)) \quad (15)$$

with  $J(h) = \{j : \exists i : h \subseteq h_i(j)\}$ , and :

$$\beta(h_i) = \begin{cases} 0 & \text{if } h_i = 0, * \\ 1 & \text{if } h_i = 1 \end{cases}$$

Attention is paid to the expected value of involved Walsh coefficients when disruption occurs for the schema  $h$ . Any schema  $h'$  that shares fixed position with  $h$  may replace it (we note  $j_p(h') = j_p(h)$ , where  $j_p$  denote the partition number of a schema). If we look at each term of the summation in 15, Goldberg wrote this expected value (that we note  $w_j^d$ ) as :

$$w_j^d = w_j \sum_{h' : j_p(h') = j_p(h)} \psi_j(\beta(h')) \quad (16)$$

And as :

$$\sum_{h' : j_p(h') = j_p(h)} \psi_j(\beta(h')) = 0$$

one comes to the conclusion that the average value of the Walsh coefficient after crossover disruption (that we note  $w_j^d$ ) can be considered as null. However, it has to be noticed that the simplification (16) (that makes the calculation possible) is valid only when the allele frequencies are equal for each position. A rigorous expression (but much more difficult to estimate!) would be :

$$w_j^d = w_j \sum_{h' : j_p(h') = j_p(h)} \psi_j(\beta(h')) P_c(h, h') \quad (17)$$

where  $P_c(h, h')$  denotes the probability to obtain the schema  $h'$  after the disruption of the schema  $h$ . If we use the expression 17, this expectation is null only if the  $P_c$  are equal. This assumption make sense only in a population where all alleles are equally represented at each position (as we would expect for a randomly generated population), but it becomes invalid as the bits frequencies evolve during the GA run. This is one of the reasons why this definition of deception is called *static* deception.



---

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN  
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399