INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Some remarks on the optimization of Hölder functions with Genetic Algorithms

Evelyne LUTTON, Jacques LEVY VEHEL

N° 2627

Juillet 1995

PROGRAMME 5

Rapport de recherche

ISSN 0249-6399

Some remarks on the optimization of Hölder functions with Genetic Algorithms

Evelyne LUTTON, Jacques LEVY VEHEL

Programme 5 — Traitement du signal, automatique et productique
Projet FRACTALES

Rapport de recherche n° 2627 — Juillet 1995 — 35 pages

Abstract: We investigate the problem of Hölder functions optimization using Genetic Algorithms (GA). We first derive a relation between the Hölder exponent of the function, the sampling rate, and the accuracy of the optimum localization, both in the domain and the range of the function. This relation holds for any optimization method which work on sampled search spaces.

We then present a finer analysis in the case of the use of a GA, which is based on a deceptivity analysis. Our approach uses a decomposition on the Haar basis, which reflects in a natural way the Hölder structure of the function. It allows to relate the deceptivity, the exponent and some parameters of the GA (including the sampling precision). These results provide some indications which may help to make the convergence of a GA easier.

Key-words: Stochastic Optimization, Genetic Algorithms, Hölder functions, Deceptivity Analysis, Fractals.

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

Quelques remarques sur l'optimisation de fonction Höldériennes à l'aide d'algorithmes génétiques

Résumé : Nous étudions le problème de l'optimisation de fonctions Höldériennes à l'aide d'algorithmes génétiques. Nous établissons tout d'abord une relation entre l'exposant de Hölder de la fonction, la fréquence d'échantillonnage et la précision de localisation de l'optimum, en position et valeur. Cette relation est valable pour toute les méthode d'optimisation qui ont accès uniquement à un échantillonnage de l'espace de recherche.

Nous proposons ensuite, dans le cas de l'emploi d'un algorithme génétique, une analyse plus fine, qui est fondée sur la notion de déceptivité. Notre approche exploite une décomposition de la fonction f à optimiser sur la base de Haar, qui reflète directement la structure Höldérienne de f. Nous obtenons ainsi une relation qui lie la déceptivité, l'exposant de f et certains paramètres de l'algorithme génétique (dont la finesse d'échantillonnage). Ces résultats fournissent des indications qui pourraient dans certains cas faciliter la convergence de l'algorithme génétique.

Mots-clé : Optimisation Stochastique, Algorithmes Génétiques, Fonctions Höldériennes, Analyse de Déceptivité, Fractals.

1 Introduction

Two main factors make the optimization of certain functions difficult: local irregularity (for instance, non differentiability) resulting in wild oscillations, and the existence of several local extrema. Stochastic optimization methods were developed to tackle these difficulties: one of their characteristic features is that no a priori hypotheses are made on the function to be optimized; no continuity, nor derivability is required, and the function is not assumed to have only one local maximum (or minimum). This makes stochastic methods useful in numerous "difficult" applications (of course often at the expense of high computation times), as for example on inverse problems appearing in material optimization, image analysis, or process control.

In addition to theoretical investigations about their convergence properties, the main challenge in the field of stochastic optimization is to set the parameters of the methods so that they are the most efficient. This problem is of obvious practical interest but it also yields some theoretical insight on the behaviour of these optimization techniques.

It is difficult to derive rules for tuning the parameters without making any assumption on the studied function. On the other hand, if we are to make restrictive assumptions, they should not rule out "interesting" functions, as for instance non differentiable functions with many local extrema. In this work, we consider a class of functions which is both quite general, as it includes smooth functions as well as very irregular ones, and sufficiently constrained so as to obtain useful results. This class is that of Hölder functions, whose definition is recalled in section 2.

Essentially, Hölder functions are continuous functions which may have, up to a certain amount, wild variations. In particular, many non differentiable continuous functions, as long as their irregularity can be bounded in a certain sense, belong to this class. Hölder functions can not in general be optimized through usual, e.g. gradient based, methods. Some "fractal" functions, as for instance the Weierstrass one (see section 2) are Hölder functions which possess infinitely many local extrema. Such functions motivate the use of stochastic optimization methods and are thus a good test to assess their efficiency.

The first parameter which has to be set, for every discrete optimization method trying to locate the optimum of a continuous function, is the sampling accuracy ϵ. We will see in section 3 that the Hölder framework helps to fix a correct accuracy.

We then focus on Genetic Algorithms (GA), which belong to the pool of artificial evolution methods, i.e. methods inspired from natural evolution principles, and show that the Hölder framework allows to obtain more specific results. Evolutionary methods in general have been used since about thirty years, and are known as particularly efficient in numerous applications (see [14, 22, 2, 24, 18, 9, 6]). They have been widely studied in various domains, from a theoretical as well as from a practical point of view. As we are dealing here with discrete optimization methods, we are interested in GAs, whose characteristic feature, in comparison with other evolutionary techniques, is that they work on discrete search spaces. Theoretical analyses of GA are based on two different approaches:

- proofs of convergence based on Markov chain modeling: for example, Davis [7] has shown that a very low decreasing of the mutation probability p_m along the generations insures the theoretical convergence onto a global optimum,
- deceptive functions analysis, based on Schema analysis and Holland's original theory [15, 10, 11, 12], which characterizes the efficiency of a GA, and allows to shed light on GA-difficult functions.

  Deceptivity has been intuitively related to the biological notion of epistasis [6], which can be understood as a sort of "non-linearity" degree. Deceptivity depends on:

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

In this paper, we concentrate on the deceptivity approach, which, as we will show, is well suited to the analysis of Hölder functions.

Section 4 recalls some basic facts about deceptivity analysis. In section 5, a deceptivity analysis is made for Hölder functions. This analysis provides a mean to efficiently tune some of the GA parameters, which is derived in section 6. Tests on "fractal" functions are presented in section 7.

2 Hölder functions

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

$$d_Y(F(x), F(y)) \leq k.d_X(x, y)^h \qquad (x, y \in X) \tag{1}$$

for some constant $k > 0$.

The following results are classical:

Proposition 1 *If F is Hölder with exponent h, it is Hölder with exponent h' for all $h' \in]0, h]$.*

Proposition 2 *Let F be a Hölder function. Then F is continuous.*

Although a Hölder function is always continuous, it needs not be differentiable (see the example of Weierstrass functions below).

Intuitively (see figures 4 and 5), a Hölder function with a low value of h looks much more irregular than a Hölder function with a high value of h (in fact, this statement only makes sense if we consider the highest value of h for which (1) holds).

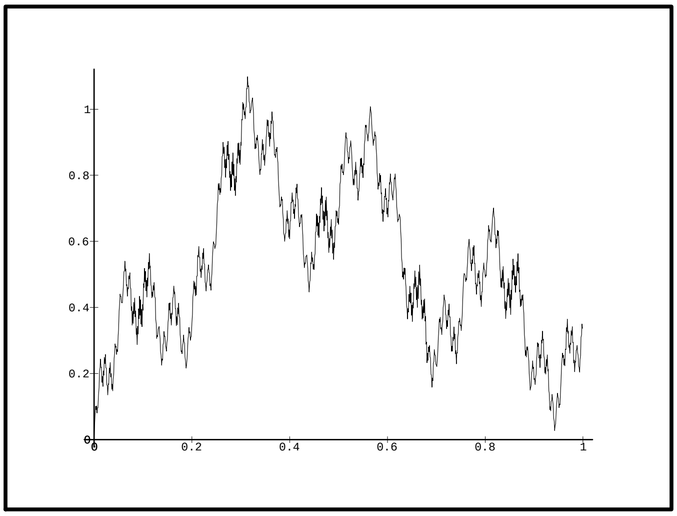

Figure 1: Weierstrass function of dimension 1.5.

The frame of Hölder functions, while imposing a condition that will prove useful for tuning the parameters of the GA, allows to consider very irregular functions, as the Weierstrass function displayed on figure 1 and defined by :

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

This function is nowhere differentiable, possesses infinitely many local optima, and may be shown to satisfy a Hölder condition with $h = s$ [8]. For such "monofractal" functions (i.e. functions having the same irregularity at each point), it is often convenient to talk in terms of box dimension d (sometimes referred to as "fractal" dimension), which, in this simple case, is $2 - h$.

Hölder functions appear naturally in some practical situations where no smoothness can be assumed and/or where a fractal behaviour arises (as for example to solve the inverse problem for IFS [21], in constrained material optimization [23], or in image analysis tasks [19, 4]). It is thus important to obtain even very preliminary clues that allow to tune the parameters of a stochastic optimization algorithm like GA, in order to perform an efficient optimization on such functions.

3 A general result on the sampling precision

We first address the problem of the tuning of the resolution (or sampling precision) in the general case, i.e. without assumption on the discrete optimization method used. This is indeed a crucial problem since if the sampling precision is inadequate, any optimization technique (even exhaustive search) may grossly fail to estimate the right position of the global optimum (see figure 2).

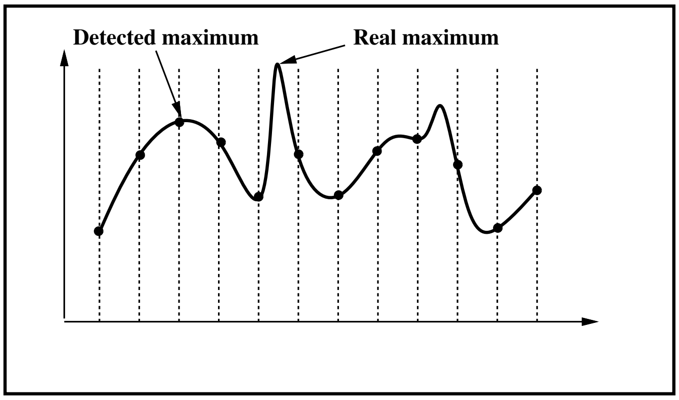

Figure 2: An inadequate sampling precision may mislead the optimization process.

In the case of a Hölder function, a very simple remark allows one to verify *a posteriori* that the chosen resolution ϵ has led to a correct estimate. The hypotheses are the following ones :

i) the function $F : \mathbf{R} \to \mathbf{R}$ is Hölder with exponent $h > 0$ and constant k (all results in this section remain true if F goes from $\mathbf{R}^n$ to $\mathbf{R}$, $n \in \mathbf{N}^*$),

ii) the discrete optimization method has a sampling precision of $\epsilon < 1$ (for instance, $\epsilon = \frac{1}{2^l}$ for a GA where l is the chromosome length). More precisely, the underlying continuous search space is sampled at regularly spaced points (x_n), with $|x_i - x_{i+1}| = \epsilon$ for all i,

iii) the discrete optimization method always gives the right answer on the discrete data : if x_m is found by the algorithm, then it is true that : $\forall i, \quad F(x_m) \geq F(x_i)$ (in case we are looking for a maximum).

This last hypothesis implies that the method is also able to locate the "true" second maximum in the discrete space, i.e. the point x'_m such that :

$$\forall i, i \neq m \Rightarrow F(x'_m) \geq F(x_i)$$

Condition **iii)** might seem to be a little too much to ask. However, our primary concern here is not to assess the quality of the optimization method itself. On the contrary, assuming that the method is perfect on discrete data, we want to find out whether it is possible to insure that a sampling precision can be set, which allows to locate the maximum in the continuous domain within a given accuracy. Moreover, it is known that some stochastic optimization methods, such as for instance Simulated Annealing [1] or GA [7], do converge with probability one to the global optimum in the discrete space under certain hypotheses. From a general point of view, we have found that GA often fulfill such a condition, and are even able to locate precisely x_m and x'_m using a sharing technique [13]. Finally, since we are working on a finite space of samples, it is always theoretically possible to assume that **iii)** is met.

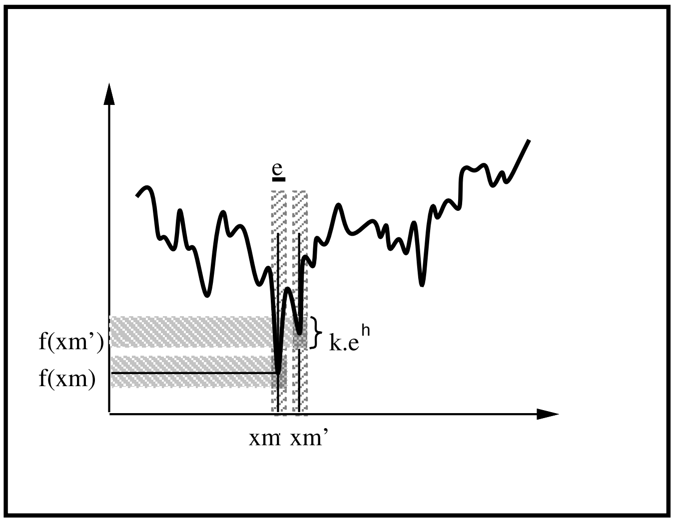

Figure 3: Sampling precision influence.

The condition on ϵ is then easy to derive. Because of the Hölder property of F, we have :

$$\forall i, x \in]x_{i-1}, x_{i+1}[\ \Rightarrow \quad |F(x) - F(x_i)| \leq k\epsilon^h$$
$$\text{thus :} \quad F(x) \leq F(x_i) + k\epsilon^h$$

$$\text{if } \ i \neq m, \qquad F(x) \leq F(x'_m) + k\epsilon^h$$

This may be rewritten as:

$$\forall x, \quad x \notin]x_{m-1}, x_{m+1}[\ \Rightarrow \ F(x) \leq F(x'_m) + k\epsilon^h$$

Thus, if:

$$F(x_m) - F(x'_m) \geq k\epsilon^h \tag{3}$$

we get:

$$\forall x, \quad x \notin]x_{m-1}, x_{m+1}[\ \Rightarrow \ F(x) \leq F(x_m)$$

This means that if (3) is verified, we cannot be in the situation of figure 2, and thus that the position of the maximum is localized with the best possible precision, i.e. ϵ (see figure 3). The result is then expressed in the following

Proposition 3 *Under conditions* **i)**, **ii)** *and* **iii)** *above, we have:*

$$F(x_m) - F(x'_m) \geq k\epsilon^h \Rightarrow \begin{cases} x^\star \in]x_{m-1}, x_{m+1}[\\ \\ F(x^\star) \in]F(x_m) - k\epsilon^h, F(x_m) + k\epsilon^h[\end{cases} \tag{4}$$

where $x^\star$ is the position of the maximum in the continuous space.

This relation quantifies the intuitive guess that if h is low (i.e. if the function is very irregular), then $F(x_m)$ and $F(x'_m)$ should clearly differ in order to yield reliable information. Otherwise, because F has large oscillations, the absolute maximum of F could be in the neighborhood of x'_m instead of in that of x_m.

Practical implications of the proposition are presented in section 6. In the case of GA optimization, it is possible to go a little further and to find an *a posteriori* validation rule not only for the sampling precision, but also for the other parameters of the method. This is what we present in the next sections.

4 Deceptivity Analysis

Our approach is based on Goldberg's deceptivity analysis [10, 11], which uses a decomposition of the function to be optimized, f, on Walsh polynomials. This decomposition allows to define a new function f', which can be understood as a sort of "preference" given by the GA to the points of the search space during the search. This function f' is in some sense a simplified version of f, perceived by the GA. The GA is said to be deceptive when the global maxima of f and f' do not correspond to the same points of the search space.

4.1 Schema theory

More precisely, this approach is based on schema theory [9, 15]. A schema represents a sub-space of the search space, and quantifies the resemblance between its representing codes: for example the schema **01⋆⋆11⋆0** is a sub-space of the space of codes of 8 bits length ($\star$ represents a "wild-card", which can be 0 or 1).

The GA modelled in schema theory is a canonical GA which acts on binary strings, and for which the creation of a new generation is based on three operators:

- an elitist *selection*, where the fitness function steps in : the probability that a solution of the current population is selected is proportional to its fitness,
- the *genetic operators* : one point crossover and bit-flip mutation, randomly applied, with probabilities p_c and p_m.

Schemata allow to represent global information about the fitness function. It has to be understood that schemata are just tools which help to understand the codes structure. A GA thus works on a population of N codes, and implicitly uses informations on a certain amount of schemata.

We recall below the so called "schema theorem" which is based on the observation that the evaluation of a single code allows to deduce some knowledge about the schemata to which that code belongs.

Theorem 1 (Schema theorem) *(Holland)*
For a given schema H, let :

- *$m(H,t)$ be the relative frequency of the schema H in the population of the t^{th} generation,*
- *$f(H)$ be the mean fitness of the elements of H,*
- *$\mathcal{O}(H)$ be the number of fixed bits in the schema H, called the* order *of the schema,*
- *$\delta(H)$ be the distance between the first and the last fixed bit of the schema, called the* definition length *of the schema.*
- *p_c be the crossover probability,*
- *p_m be the mutation probability of a gene of the code,*
- *$\bar{f}$ be the mean fitness of the current population.*

Then :

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

The quantities $\delta(H)$ and $\mathcal{O}(H)$ help to model the influence of the genetic operators on the schema H : the longer the definition length of the schema is, the more frequently it is broken by a crossover (the schema theory has been developed for a one point crossover). In the same way, the bigger the order of H is, the more frequently H is broken by a mutation.

From a qualitative view point, this formula means that the "good" schemata, having a short definition length and a low order, tend to grow very rapidly in the population. These particular schemata are called *building blocks.*

The usefulness of the schema theory is twofold : first, it supplies some tools to check whether a given representation is well-suited for a GA. Second, the analysis of the nature of the "good" schemata, using for instance Walsh functions [9, 16], can give some ideas on the GA efficiency [6], via the notion of deceptivity that we describe below.

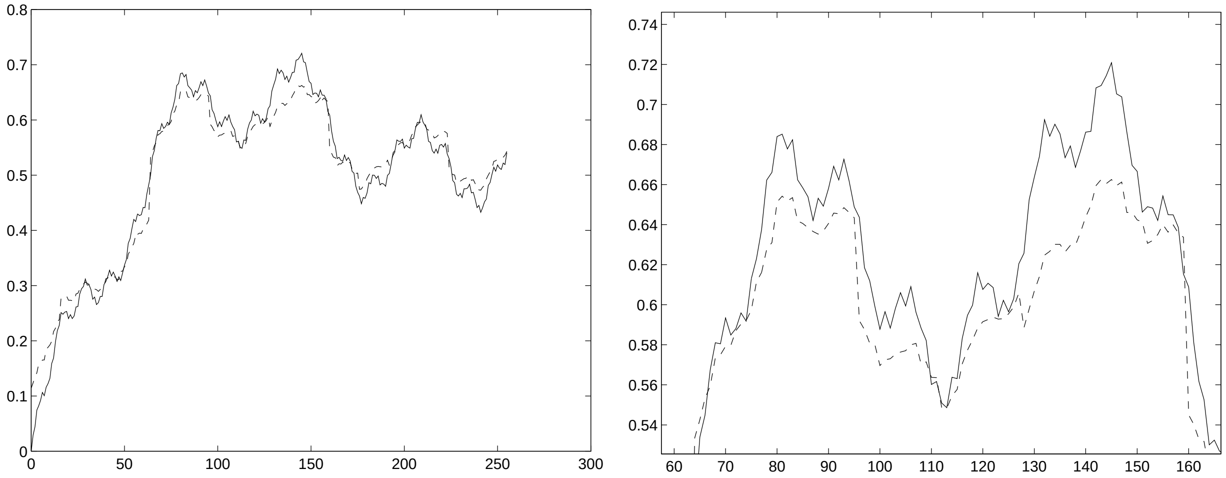

Figure 4: *Left:* f (continuous) and f' (dotted) for a Weierstrass function of dimension 1.2 sampled on 8 bits. *Right:* zoom on the region of the first two maxima: the function is not 0-deceptive although it is 0.03-deceptive.

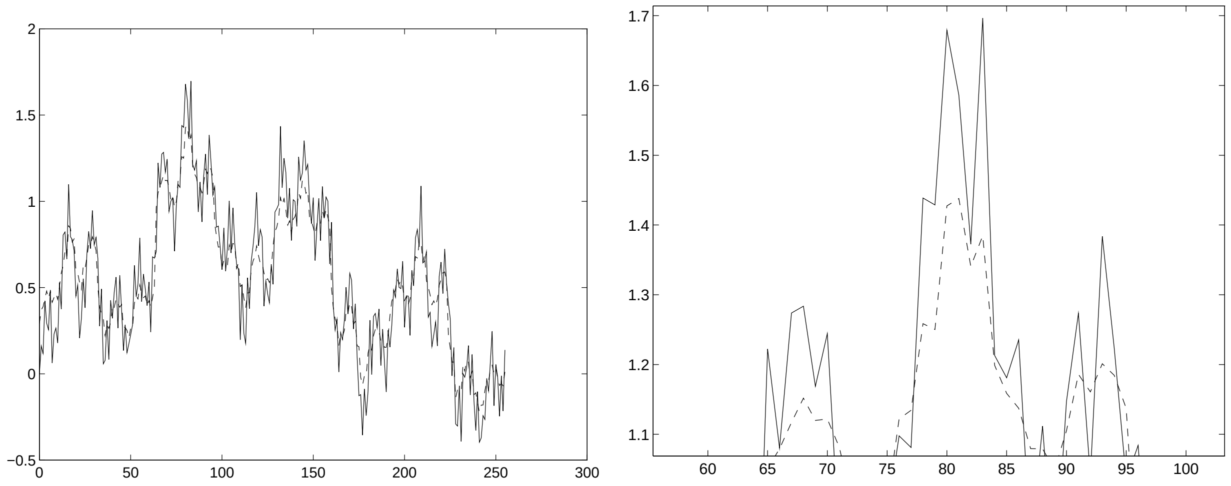

Figure 5: *Left:* f (continuous) and f' (dotted) for a Weierstrass function of dimension 1.7 sampled on 8 bits. *Right:* zoom on the region of the first two maxima: the function is 0-deceptive although it is not 0.05-deceptive.

4.2 Walsh polynomials and Deceptivity characterization

In order to test if a given function f is easy or difficult to optimize for a GA, one could verify the "building blocks" hypothesis:

1. identify the building blocks: i.e. compute all the mean fitnesses of the short schemata which are represented within a generation, and identify as building blocks the ones whose representation increases along the evolution,
2. verify whether the optimal solution belongs to these building blocks, to know if the building blocks may confuse the GA, or not.

However, this procedure is obviously computationally intractable. Instead, Goldberg has suggested to use a method based on a decomposition of f on the orthogonal basis of Walsh functions on $[0..2^l-1]$, where $[0..2^l-1]$ denotes the set of integers of the interval $[0, 2^l-1]$.

On the search space $[0..2^l-1]$, we can define 2^l Walsh polynomials as:

$$\Psi_j(x) = \prod_{t=0}^{l-1}(-1)^{x_t j_t} = (-1)^{\sum_{t=0}^{l-1} x_t j_t} \qquad \forall x, j \in [0..2^l-1]$$

x_t and j_t are the values of the t^{th} bit of the binary decomposition of x and j.

It is well known that these Walsh polynomials form an orthogonal basis of the set of functions defined on $[0..2^l-1]$, and we let $f(x) = \sum_{j=0}^{2^l-1} w_j \Psi_j(x)$ be the decomposition of the function f.

The deceptivity of f is characterized through the function f' [10, 11] defined as follows:

$$f'(x) = \sum_{j=0}^{2^l-1} w'_j \Psi_j(x) \qquad \text{with} \qquad w'_j = w_j(1 - p_c \frac{\delta(j)}{l-1} - 2p_m \mathcal{O}(j)) \tag{5}$$

The quantities δ and $\mathcal{O}$ are defined for every j in a similar way as for the schemata: $\delta(j)$ is the distance between the first and the last non-zero bits of the binary decomposition of j, and $\mathcal{O}(j)$ is the number of non-zero bits of j.

For $\epsilon \geq 0$ let:

$$N_\epsilon = \{x \in [0..2^l-1] / |f(x) - f^*| \leq \epsilon\} \quad \text{and} \quad N'_\epsilon = \{x \in [0..2^l-1] / |f'(x) - f'^*| \leq \epsilon' = \frac{f'^* - w_0}{f^* - w_0}\epsilon\}$$

where f^* (resp. f'^*) is the global optimum of f (resp. f'). Recall that w_0 is the mean value of both f and f'.

Definition 2 (ϵ-deceptivity) *f is said to be* ϵ**-deceptive** *if $N_\epsilon \not\subset N'_\epsilon$.*

Remark: ϵ-deceptivity is not monotonic: for some 0-deceptive functions, an ϵ may be found such that the function is not ϵ-deceptive. Reversely, for some non-0-deceptive functions, we may also find an ϵ' such that the function is ϵ'-deceptive. This fact is particularly obvious on figures 4 and 5. In the following we will only consider 0-deceptivity as our deceptivity criterion.

5 Haar polynomials for the deceptivity analysis of Hölder functions

In order to perform a valuable deceptivity analysis for Hölder functions, we have to replace the decomposition on the Walsh basis by a more suited one. This new basis should allow us to relate the deceptivity to the irregularity of the Hölder function, i.e. to its Hölder exponent. Indeed, it is intuitively obvious that the more irregular the function is (i.e. the lower the Hölder exponent is), the more deceptive it is likely to be. On figures 4 and 5 are drawn f and f' for Weierstrass functions of dimension 1.2 and 1.7, both sampled on 8 bits: the Weierstrass function of dimension 1.2 is here not deceptive while the Weierstrass function of dimension 1.7 is deceptive.

There exist simple bases which permit to characterize in a certain sense the irregularity of a function in terms of its decomposition coefficients. Wavelet bases possess such a property.

The wavelet transform (WT) of a function f consists in decomposing it into elementary space-scale contributions, associated to the so-called *wavelets* which are constructed from a single function, the *analyzing wavelet* ϕ, by means of translations and dilations. The WT of the function f is defined as:

$$T_\phi[f](b,a) = \frac{1}{a}\int_{-\infty}^{+\infty} \phi\left(\frac{x-b}{a}\right) f(x)dx$$

where $a \in \mathbf{R}^+$ is a scale parameter and $b \in \mathbf{R}$ is a space parameter. The analyzing wavelet ϕ is a square integrable function of zero mean, generally chosen to be well localized in both space and frequency.

Our approach is based on the use of the simplest wavelets, i.e. Haar wavelets, which are defined on the discrete space $[0..2^l - 1]$ as:

$$H_{2^q+m}(x) = \begin{cases} 1 & for \ \ (2m)2^{l-q-1} \leq x < (2m+1)2^{l-q-1} \\ -1 & for \ \ (2m+1)2^{l-q-1} \leq x < (2m+2)2^{l-q-1} \\ 0 & otherwise \ \ in \ \ [0..2^l-1] \end{cases}$$

with $q = 0, 1, \ldots, l-1$ and $m = 0, 1, \ldots, 2^q - 1$: q is the degree of the Haar function, related to the scale of the wavelet and m corresponds to its localization (see figure 6).

These functions form an orthogonal basis of the set of functions defined on $[0..2^l - 1]$. Any function f of $[0..2^l - 1]$ can be decomposed as:

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

5.1 Haar coefficients can be bounded

Suppose that the function f to be optimized is the sampling, with precision $\epsilon = \frac{1}{2^l}$, of a Hölder function F defined on $[0,1]$:

$$\forall x \in [0..2^l-1], \quad f(x) = F(\frac{x}{2^l})$$

Using the definition of the Haar functions $H_j, \ \ j = 2^q + m$, we write:

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

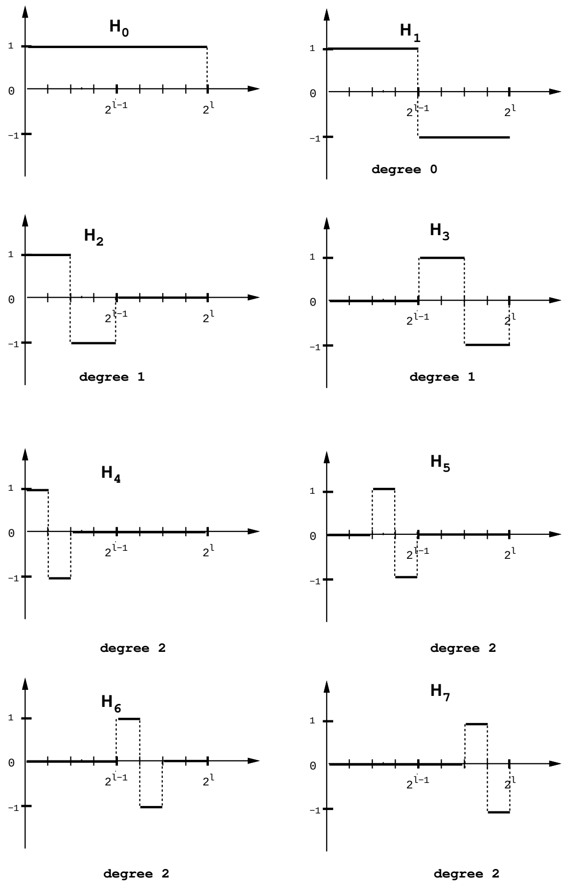

Figure 6: Haar functions for $l = 3$.

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

$$\Longleftrightarrow h_j = \frac{1}{2^{l-q}} \sum_{x=(2m)2^{l-q-1}}^{(2m+1)2^{l-q-1}} [F(\frac{x}{2^l}) - F(\frac{x}{2^l} + 2^{-q-1})]$$

Recall that :

$$\forall y \in [0,1[, \quad y + \eta \in [0,1[, \quad |F(y) \ - \ F(y+\eta)| \ \leq \ k|\eta|^h$$

then

$$\forall x \in [0..2^l - 1] \ \ \forall q \in [0..l-1], \quad |F(\frac{x}{2^l}) - F(\frac{x}{2^l} + 2^{-q-1})| \leq k2^{(-q-1)h}$$

We finally obtain the well-known bound for the Haar coefficients of a Hölder function :

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

This inequality is imaged on figure 7. The following remark is relevant for practical implementation: the optimal value of k (i.e. the lowest one) depends on the sampling precision. The curves of figure 7 are drawn with $k = 2.5$ for a Weierstrass function sampled on 12 bits, and with $k = 3$ for an FBM [1] sampled on 10 bits.

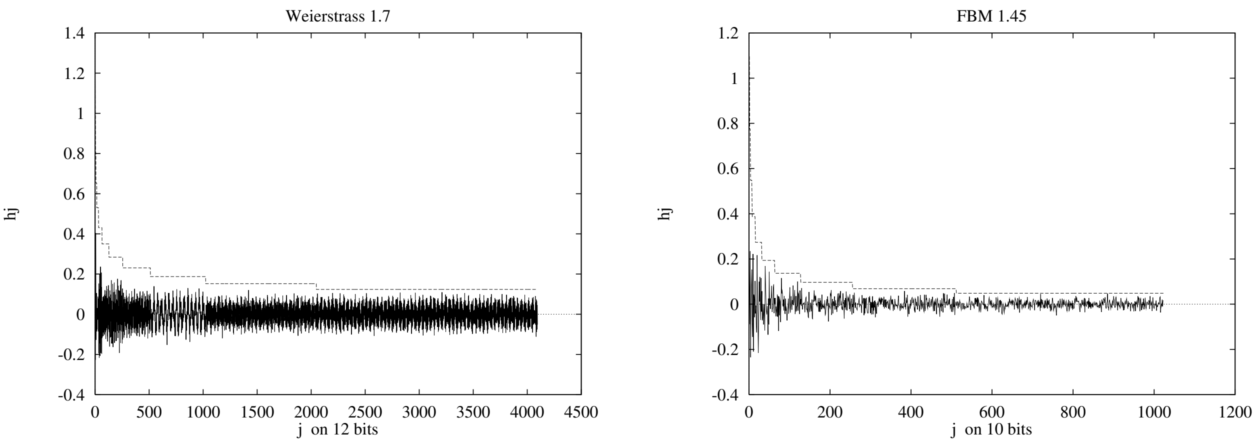

Figure 7: Haar coefficients (continuous) and bound (dotted) for a Weierstrass function of dimension 1.7 sampled on 12 bits (left) and an FBM of dimension 1.45 sampled on 10 bits (right).

5.2 Deceptivity for Hölder functions

The use of a Haar decomposition for deceptivity analysis has already been proposed in [17], but it seems that the complete computation of the adjusted coefficients (i.e. the coefficients of the function f') was not explicit. We thus use here a transformation between Walsh and Haar bases to explicitly compute the adjusted Haar coefficients. Details of the computations are given in appendices A to D, and only the main steps are presented below.

We have (see appendix A):

$$H_j(x) = \frac{1}{2^q}\left(\sum_{k=0}^{2^q-1}(-1)^{\sum_{t=0}^{l-1} m_t k_t}\Psi_{2^{l-q-1}+k2^{l-q}}(x)\right) \quad \text{with} \quad j = 2^q + m, \quad q \in [0..l-1] \quad \text{and} \quad m \in [0..2^q - 1]$$

m_t and k_t represent the t^{th} bit of the binary decomposition of m and k: $m = \sum_{t=0}^{l-1} m_t 2^t$ and $k = \sum_{t=0}^{l-1} k_t 2^t$.

Conversely (see appendix B):

$$\Psi_j(x) = \sum_{m=0}^{2^q-1}(-1)^{\sum_{t=0}^{q-1} k_t m_t} H_{2^q+m}(x) \quad \text{with} \quad j = 2^{l-q-1}(1+2k), \quad k \in [0..2^q-1] \quad \text{and} \quad q \in [0..l-1]$$

We thus obtain the expression of Haar adjusted h'_j coefficients (see appendix C and D):

[1]FBM stands for Fractional Brownian Motion. For definition and properties of the Fractional Brownian Motion (FBM) see for instance [20]. As Weierstrass functions, paths of FBM (almost surely) verify a Hölder property, the irregularity being the same at each point. Thus an FBM with Hölder exponent h has box dimension equal to $2 - h$.

$$f'(x) = \sum_{j=0}^{2^l-1} h'_j H_j(x)$$

$$\begin{aligned} h'_{2^q+m} &= h_{2^q+m}[1 - \frac{p_c}{l-1}(1 + \frac{1+(q-2)2^q}{2^q}) - 2p_m(1+\frac{q}{2})] \\ &-\frac{p_c}{2^q(l-1)}\sum_{u=0}^{q-1}(1-2^{u+1})\sum_{r=0}^{2^{q-u-2}} h_{2^q+\sum_{t=0}^{u-1} m_t 2^t + (1-m_u)2^u + r2^{u+1}} \\ &-p_m \sum_{t=0}^{q-1} h_{2^q+m+(1-2m_t)2^t} \end{aligned}$$

We are now ready to compute an upper bound for the quantity $|f(x) - f'(x)|$:

$$\begin{aligned} |f(x) - f'(x)| &= |\sum_{j=1}^{2^l-1}(h_j - h'_j)H_j(x)| \\ &\leq \sum_{j=1}^{2^l-1}|h_j - h'_j||H_j(x)| \end{aligned}$$

Notice that for $x \in [0..2^l-1]$:

$$H_j(x) \neq 0, \quad j = 2^q + m \quad \Longleftrightarrow \quad E(\frac{x}{2^{l-q-1}}) = 2m \;\; or \;\; E(\frac{x}{2^{l-q-1}}) = 2m+1$$

where $E()$ represents the integer part of x.

For a fixed x, and for each q, there exists a single value m_x of m such that $H_{2^q+m}(x) \neq 0$, and:

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

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

with m_x such that $E(\frac{x}{2^{l-q-1}}) = 2m_x$ or $E(\frac{x}{2^{l-q-1}}) = 2m_x + 1$ and thus

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

The bounds on the Haar coefficients of order q yield, after some computation:

$$\forall m, \qquad |h_{2^q+m} - h'_{2^q+m}| \;\; \leq \;\; k2^{-h(q+1)}[\frac{p_c}{2^q(l-1)}[(1+(q-1)2^q] + p_m(1+q)]$$

Thus:

$$\begin{aligned} |f(x) - f'(x)| &\leq k\sum_{q=0}^{l-1}(2^{-h(q+1)}[\frac{p_c}{2^q(l-1)}(1+(q-1)2^q) + p_m(1+q)]) \\ &\leq k\;\frac{p_c}{l-1}(\sum_{q=0}^{l-1}2^{-h(q+1)-q}(1+(q-1)2^q)) + k\;p_m(\sum_{q=0}^{l-1}2^{-h(q+1)}(1+q)). \end{aligned}$$

We may now state

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 (5). Then:*

$$\forall x \in [0..2^l - 1], \qquad |f'(x) - f(x)| \leq k * B(p_m, p_c, l, h) \tag{6}$$

with

$$\begin{aligned} B(p_m, p_c, l, h) = \quad & \frac{p_c}{l-1} 2^{-h} \left[\frac{2^{-l(h+1)} - 1}{2^{-(h+1)} - 1} + \frac{(1 - 2^{1-h})(2^{-hl} - 1) - l2^{-hl}(1 - 2^{-h})}{(2^{-h} - 1)^2} \right] \\ + \quad & p_m \frac{2^{-h}}{\left(2^{-h} - 1\right)^2} \left[1 + 2^{-hl}(l2^{-h} - l - 1) \right] \end{aligned}$$

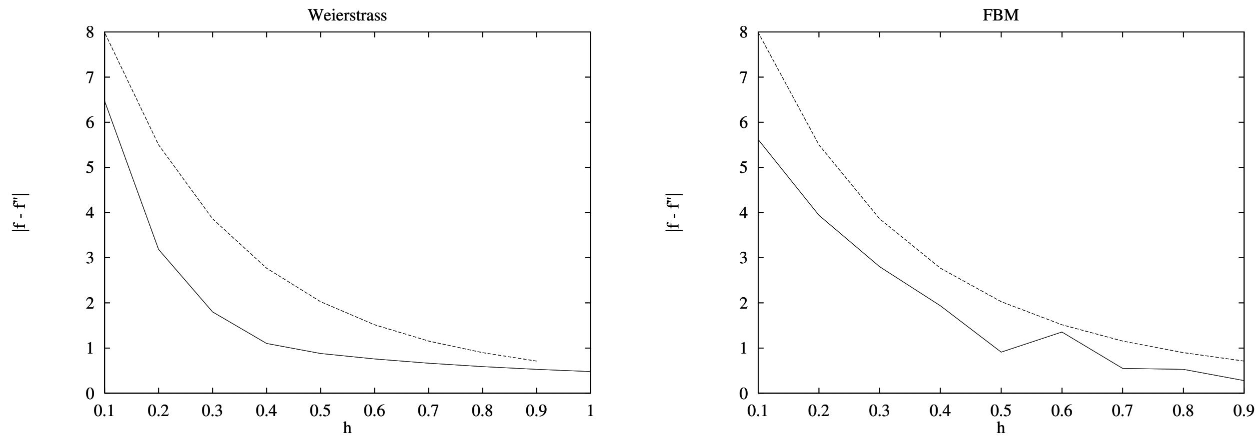

Figure 8: $B(p_m, p_c, l, h)$ (dotted) and computed maximal differences between f and f' (continuous) as function of h, for Weierstrass functions (left), and FBM's (right), $l = 8$ bits, $p_c = 0.9$, $p_m = 0.25$.

Since for all admissible values of l, p_m, p_c, B is an increasing function of h, this relation implies that the smaller h is (i.e. the more irregular the function is), the more different the functions f and f' may be, thus the more deceptive f is likely to be. This first fact bears some analogy with the results stated in section 3, and is confirmed by numerical simulations displayed on figure 8.

A fine analysis of the function $B(p_m, p_c, l, h)$ is rather uneasy, because B defines a hyper-surface of $\mathbf{R}^5$, but the following results may be stated (see figures 9 to 13, which are 3D "cuts" of that surface).

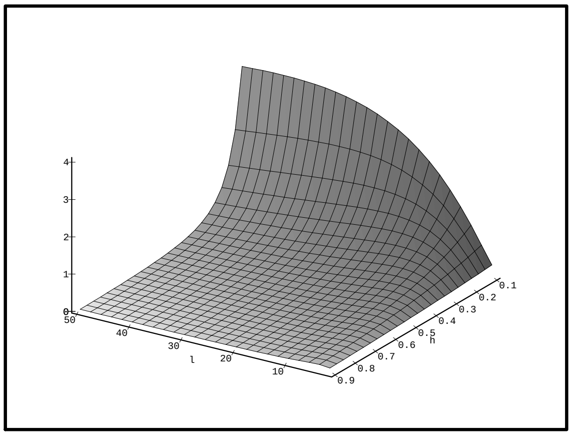

Figure 9: $B(p_m, p_c, l, h)$ as a function of (l, h) for $p_m = 0.01$, $p_c = 0.7$.

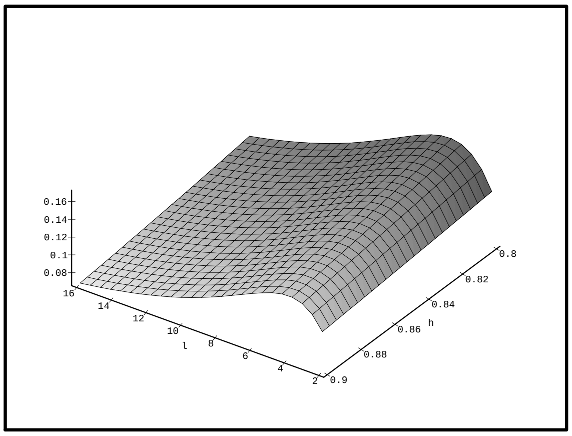

Figure 10: Zoom on $B(p_m, p_c, l, h)$ for $p_m = 0.01$, $p_c = 0.7$ and large values of h.

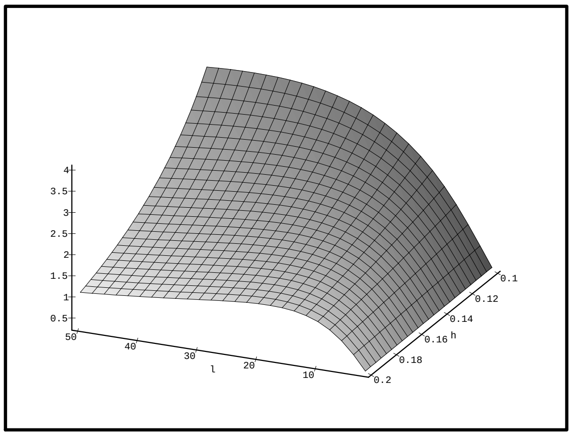

Figure 11: Zoom on $B(p_m, p_c, l, h)$ for $p_m = 0.01$, $l = 8$ bits and small values of h.

- Dependence on l (figures 9, 10 and 11):

  $B(p_m, p_c, l, h)$ has the following asymptotic behaviour when $l \to \infty$:

$$\lim_{l \to \infty} B(p_m, p_c, l, h) = p_m \frac{2^{-h}}{(2^{-h} - 1)^2}$$

  This limit does not depend on p_c [2] (see figure 13). We also have:

$$B(p_m, p_c, 2, h) = p_c 2^{-2h-1} + p_m (2^{-h} + 2^{1-2h})$$

[2] This fact is due to the definition of the mutation and crossover probabilities: *each gene* of the chromosome is mutated with probability p_m, while the crossover probability is defined on a whole chromosome. Thus when l tends to infinity, for fixed mutation and crossover probabilities, mutation becomes more and more important with respect to crossover. It may also be argued that the one point crossover as it is defined here is meaningless when l is infinite.

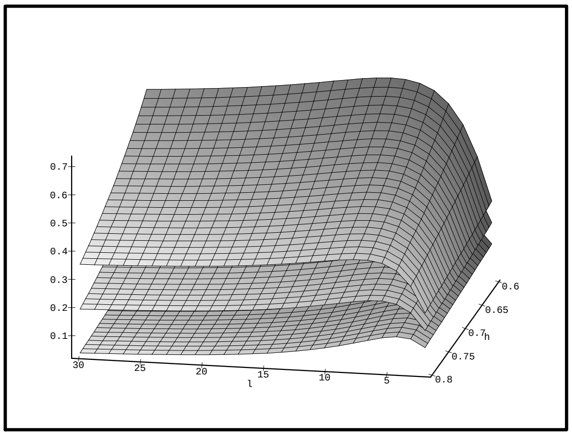

Figure 12: Influence of p_m: $B(p_m, p_c, l, h)$ as a function of (l, h) for $p_c = 0.7$ fixed, and $p_m = 0.001, 0.05$ and 0.1.

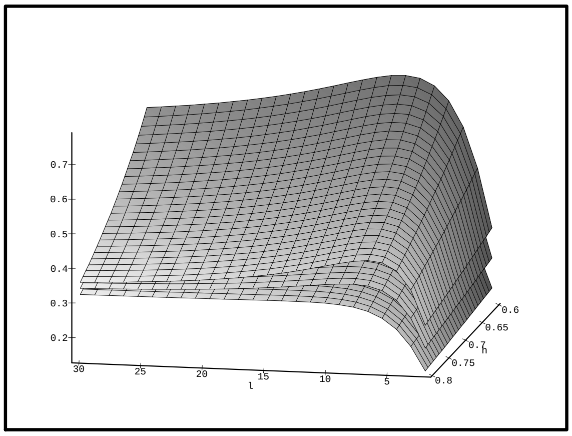

Figure 13: Influence of p_c: $B(p_m, p_c, l, h)$ as a function of (l, h) for $p_m = 0.01$ fixed, and $p_c = 0.1, 0.5$ and 0.9.

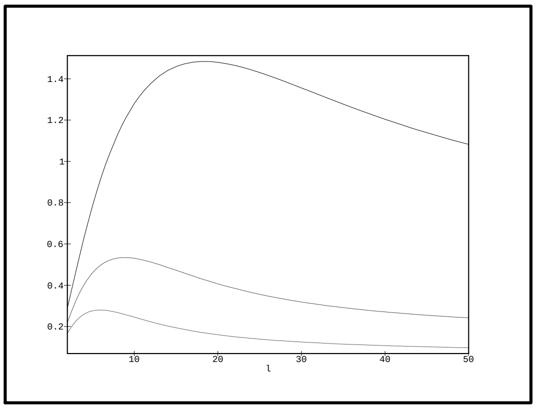

Figure 14: $B(p_m, p_c, l, h)$ as a function of l, $p_m = 0.01$, $p_c = 0.7$, and for different values of h (up: 0.2, middle: 0.4, down: 0.6).

$B(p_m, p_c, l, h)$ increases with l for small values of l, and then decreases for larger values of l. It may be proved that the parameterized curves $B(p_m, p_c, \bullet, h)$ admit one and only one maximum at l_{max} in $[2, \infty[$. l_{max} increases when h decreases, i.e. when the function f becomes more and more irregular (see figure 9 and 14).

A sufficient condition for non-deceptivity is $B(p_m, p_c, l, h) = 0$, which is in general not possible. A qualitative approach is then to keep B as small as possible. In that respect, a strategy to set the optimal value of l is the following one:

- try to find a small value $l < l_{max}$ which is a tradeoff between a sufficiently fine sampling to correctly capture the optimum (according to section 3), while trying to limit the number of samples,
- if no "small" values can be found, take a large value $l > l_{max}$, compatible with computational requirements.

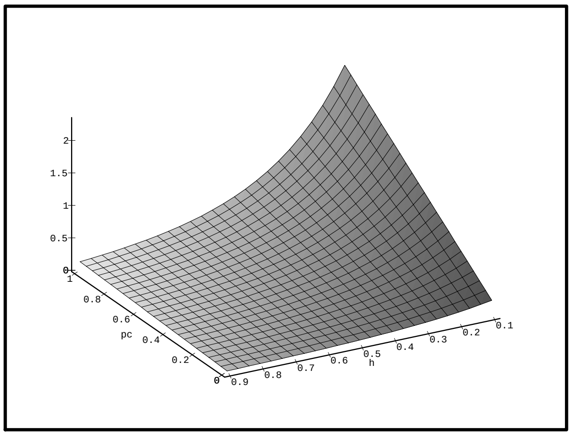

Figure 15: $B(p_m, p_c, l, h)$ as a function of (p_c, h) for $l = 8$ bits, $p_m = 0.01$.

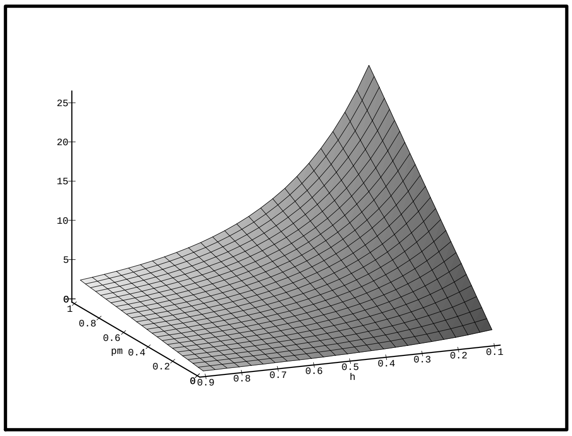

Figure 16: $B(p_m, p_c, l, h)$ as a function of (p_m, h) for $l = 8$ bits, $p_c = 0.7$.

- Dependence on p_c and p_m (figures 15 and 16) :

  Deceptivity decreases as p_c and p_m decrease. This effect is more important for small values of h than for large values of h. Note also that deceptivity is less influenced by p_c than by p_m, and that the influence of p_m increases when h is small and when l is large. For example, if $h = 0.5$ and $l = 8$ bits, the influence of p_m on the deceptivity is about 15 times more important that the influence of p_c. From a practical point of view, it means that decreasing p_m is much more efficient than decreasing p_c, in order to reduce deceptivity. This fact also confirms the interest of the mutation probability decrease technique, especially for irregular functions. Mutation probability decrease has been theoretically justified for a simple model of GA, without crossover, with Markovian approaches (see [7]), and its practical efficiency has been experienced. Formula (6) shows that decreasing the mutation probability tightens the bound on $|f - f'|$, thus probably decreasing the deceptivity of the function, i.e. making the convergence of the GA easier.

6 Practical non-deceptivity criterion

Formula (6) provides a relation involving mutation and crossover probabilities, which may help to set these probabilities in order to make the convergence of the GA easier. Notice however that this relation only gives a bound, which needs not be optimal.

We present a simple sufficient non-deceptivity condition in the same spirit as in section 3. Formula (6) points out that the function f' is located inside a strip of extent $2kB$ around f. Suppose that we have detected the two first optima of the functions (using for example a GA with sharing), i.e. x_1^* and x_2^* corresponding to the values f_1^* and f_2^* (see figure 17).

If the following relation holds :

$$|f_1^* - f_2^*| > 2 * k * B(p_m, p_c, l, h)$$

then we are insured that the maximum of the function f' will be near to the one of f (i.e. near f_1^*). This situation is depicted in figure 18. In this case, the function is not deceptive. If that relation does not hold (figure 19), we cannot say anything about the deceptivity of the function.

The extent of the strip around f may be tuned by changing the values of the parameters p_c, p_m and l, with the constraint that l must be larger than a fixed threshold, deduced from Proposition 1, that insures

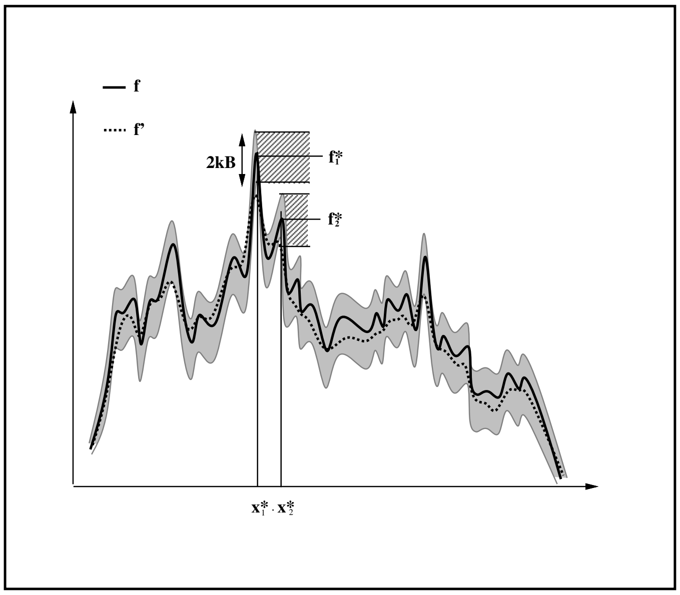

Figure 17: Use of the bound B, in a practical implementation of a GA with sharing.

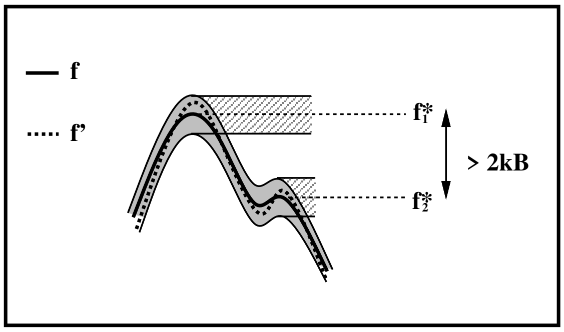

Figure 18: Favorable case : no deceptivity.

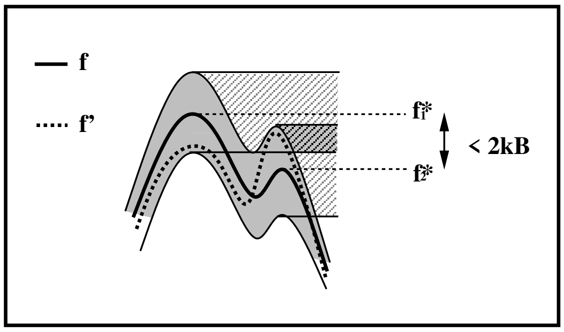

Figure 19: Unfavorable case : possible deceptivity.

a correct sampling: the relation $|f_1^* - f_2^*| > k\frac{1}{2^{lh}}$ is used to find a lower bound for l, while the relation $|f_1^* - f_2^*| > 2 * k * B(p_m, p_c, l, h)$ provides conditions on p_m, p_c and l.

To summarize, the *a posteriori* validation test may be written as :

$$\boxed{|f_1^* - f_2^*| > k * max(2 * B(p_m, p_c, l, h), \frac{1}{2^{lh}})}$$

A practical method to adjust p_c, p_m and l may be derived from the observations made in section 5.2. Let us define :

- Δ as the maximal value of B which insures non-deceptivity :

$$\Delta = \frac{|f_1^* - f_2^*|}{2k}$$

.

- B_2 and B_∞ as the two limit values of B considered as a function of l :

$$B_2 = B(p_m, p_c, 2, h) = p_c 2^{-2h-1} + p_m(2^{-h} + 2^{1-2h}) \quad (7)$$

$$B_\infty = \lim_{l\to\infty} B(p_m, p_c, l, h) = p_m \frac{2^{-h}}{(2^{-h}-1)^2} \quad (8)$$

- l_{max} as the value of l where B reaches its maximum $B_{max} = B(p_m, p_c, l_{max}, h)$,
- l_0 as the minimal sampling rate that provides a correct localization :

$$l_0 = \frac{\log_2(2\Delta)}{h}$$

We wish to find parameters l, p_m, p_c such that :

$$l \geq l_0 \quad \text{and} \quad B \leq \Delta$$

The first test to perform is to check whether $\Delta > B_2$ or $\Delta > B_\infty$. If not, then decrease p_m and/or p_c until at least one of the two inequalities is verified.

Once this is done, four cases may arise (figures 20 to 22 are drawn for particular values, similar figures may be drawn in other cases), due to the shape of the curve $B(p_m, p_c, \bullet, h)$, and according to the values of l_0, l_{max} and Δ. Each configuration defines a "permitted area" in which l has to be searched :

- the "permitted area" is $[l_0, l_1]$ with $l_1 < l_{max}$ (figure 20),
- the "permitted area" is $[l_0, l_1] \bigcup [l_2, +\infty[$ with $l_1 < l_{max} < l_2$ (figure 21),
- the "permitted area" is $[l_2, +\infty[$ with $l_{max} < l_2$ (figure 22),
- the "permitted area" is $[l_0, +\infty[$.

The first, second and fourth case are favorable, since it will be possible to find a reasonably low value of l. In the third case, arbitrary large values of l may occur, leading to computationally intractable settings. These different configurations may be modified by changing p_m and p_c. However, when decreasing p_m, and especially p_c, it becomes necessary to increase the population size and generations number of the GA, in order to maintain what we can intuitively call the "search potentiality".

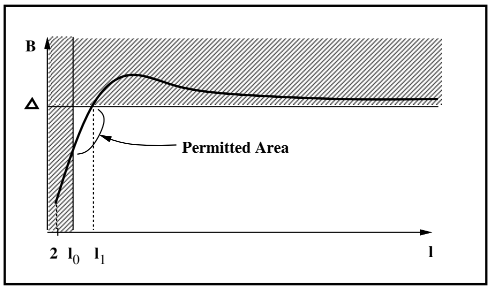

Figure 20: First case: permitted area before l_{max}.

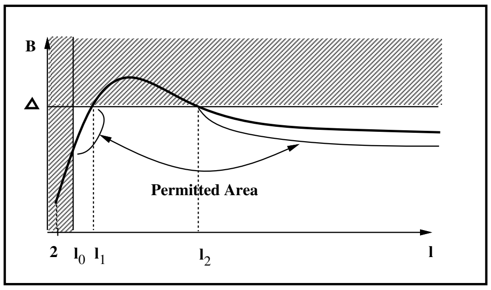

Figure 21: Second case: permitted area before and after l_{max}.

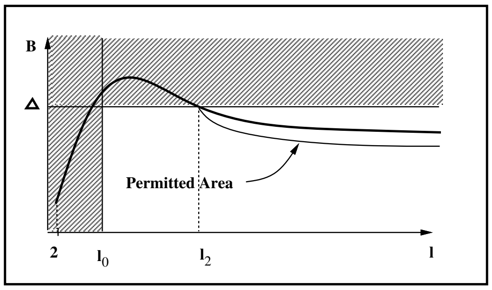

Figure 22: Third case: permitted area after l_{max}.

The *a posteriori* validation procedure reads:

1. estimate k and h from the data,
2. choose a first set of p_c, p_m and l parameters,
3. run the GA in order to detect f_1^* and f_2^*,
4. if the relation $|f_1^* - f_2^*| > k\frac{1}{2^{lh}}$ is not verified, increase l and re-run the program until it is verified,
5. if the relation $|f_1^* - f_2^*| > 2 * k * B(p_m, p_c, l, h)$ is not verified, decrease p_c and p_m (to a certain extent), and/or increase l and re-run the program until it is verified.

7 Experimental results

In order to verify experimentally the ideas presented above, we have performed tests on Weierstrass functions (see equation 2). On figure 23 and 24 are shown Weierstrass functions with $h = 0.95$ and 0.3 respectively, which correspond to box dimensions of 1.05 and 1.7. The range of x is [0,1], b is taken equal to 5, and we look for the maxima of the functions.

1. **Minimization of $W_{5,1.95}$ $(h = 0.95)$:**

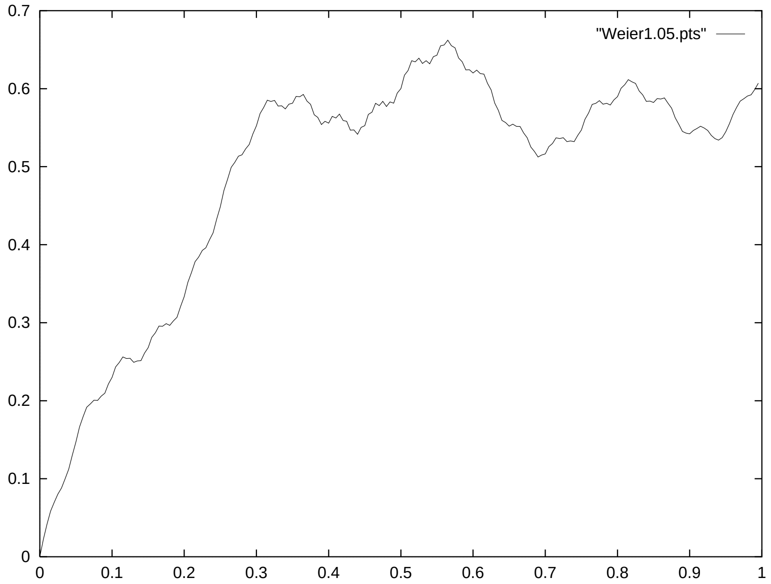

Figure 23: Weierstrass function of dimension 1.05.

- **Accuracy test :** according to section 3, the test for accuracy reads :

$$|f_1^* - f_2^*| > k\epsilon^{0.95} = k2^{-0.95l} \tag{9}$$

We start with an 8-bit sampling, and run the GA during 50 generations, with $p_m = 0.0025$, $p_c = 0.5$, and a population size of 50. The detected two first maxima are :

$$\begin{array}{ll} x_1 = 0.566406 & f_1^* = 0.661077 \\ x_2 = 0.523438 & f_2^* = 0.638224 \end{array}$$

The on-line estimation of k gives $k = 3.114$, and we compute :

$$|f_1^* - f_2^*| = 0.022852 > 3.114 * 2^{-7.6} = 0.01604$$

Since the test is verified, we expect good localization. In fact, as one can verify it on figure 23, the G.A. has found the right maximum.

- **Deceptivity test :** according to section 6, we would also like to verify :

$$|f_1^* - f_2^*| > 2 * k * B \tag{10}$$

In our case, we get :

$$|f_1^* - f_2^*| = 0.022852 < 2kB = 0.620156$$

The function may then be deceptive.

As $B_2 = 0.123208$ and $B_\infty = 0.001294$, it is possible to find a value l for which the function is not deceptive for the GA (we are in the case corresponding of figure 22). However, with the above parameters, this value is so large that it does not allow to perform the computations. We thus decrease p_m and p_c in order to get a more reasonable value for l.
As a second test, we choose $l = 40$, $p_m = 0.001$, $p_c = 0.1$, a population size of 100, and we run the GA during 50 generations.
This time we get :

$$x_1 = 0.565846.. \qquad f_1^* = 0.662315..$$
$$x_2 = 0.525944.. \qquad f_2^* = 0.638076..$$

and :

$$|f_1^* - f_2^*| = 0.024239 > k2^{-0.95*40} = 1.13 * 10^{-11}$$
$$|f_1^* - f_2^*| = 0.024239 > 2kB = 0.016778$$

with these parameters, we thus have both an accurate sampling and a non deceptive function. Indeed, it may be verified that the GA has given the right maximum.

2. **Minimization of $W_{5,1.7}$ $(h = 0.3)$:**

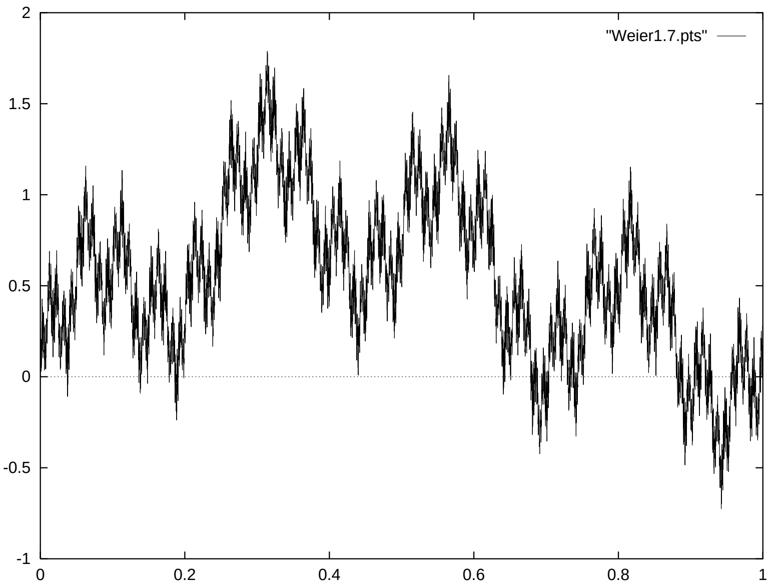

Figure 24: Weierstrass function of dimension 1.7.

The on-line estimation of k provides $k = 3.154301$.

The GA is run during 50 generations, with an 8 bits sampling, $p_m = 0.001$, $p_c = 0.7$, and a population size of 100, yielding :

$$x_1 = 0.324219 \qquad f(x_m) = 1.696581$$
$$x_2 = 0.312500 \qquad f(x_{m'}) = 1.679072$$

- **Accuracy test :**

  $$|f_1^* - f_2^*| = 0.01751 < 3.154301 * 2^{-0.3*8} = 0.59616$$

  We thus have to try again with a finer resolution.
  With a 30 bits sampling, all the other parameters being kept the same, we obtain :

  $$x_1 = 0.314137.. \qquad f(x_m) = 1.824930..$$
  $$x_2 = 0.314061.. \qquad f(x_{m'}) = 1.782212..$$

  and

  $$|f_1^* - f_2^*| = 0.042718 > 3.154301 * 2^{-0.3*30} = 0.0061603$$

  The sampling is correct, and the G.A. has given a good localization.

- **Deceptivity test** (30-bits sampling):
  The relation reads:

  $$|f_1^* - f_2^*| = 0.042718 < 2kB = 2.404567$$

  The function may be deceptive.

In this case, we have not been able to find an affordable parameters setting for which the non-deceptivity criterion is met. However, it can be verified that the right optimum has been correctly located even though the *a posteriori* non-deceptivity criterion is not verified.

The examples we have displayed above are in some sense "worst-case" examples, because the functions are everywhere irregular, i.e. nowhere differentiable with infinitely many local maxima around each point, which makes it difficult for the algorithm to correctly locate the two first maxima of the functions. When the accuracy increases, the GA discovers more and more maxima, which get closer and closer. This explains why, in some configurations, it is impossible to find an *a posteriori* non-deceptive parameters setting, as in the second case we have presented here. For other functions, the task may be much easier for the GA, and reasonable non-deceptive parameters settings may exist. We stress again that the non-deceptivity criterion we have derived is a sufficient condition: as we have seen, the GA may perform well even though it has been declared "potentially deceptive" by the test.

To summarize, while the accuracy test seems to give precise indications, the deceptivity test has to be used more carefully, only as a "confidence" measure on the results of the GA.

8 Conclusion

We have shown that in the framework of Hölder functions, it is possible to derive criteria to *a posteriori* validate the parameters setting of a discrete optimization method. An accuracy condition has been derived, which is valid for any search method.

The use of Haar decomposition instead of Walsh decomposition yields some interesting results for the particular case of optimization with GA. These theoretical results quantify the intuitive guess that the more irregular the function looks like, the more deceptive it is likely to be.

The explicit formula obtained in section 5.2 provides a relation between:

- an intrinsic parameter of the function to be optimized: its Hölder exponent h,
- the parameters of the GA: l, p_m and p_c.

A simple analysis of this formula has allowed to shed new light on previous results obtained by other theoretical approaches of GA. Moreover, we have derived a simple sufficient non-deceptivity criterion. We thus have, in the particular case of Hölder functions, a mean to *a posteriori* validate the parameters setting.

Future work should be done in the following directions:

1. *Generalization to local Hölder characterization*: such an analysis would provide a variable-size strip around the function, yielding more precise results, at the expense of more complex computations.
2. *Use of other analyses than the deceptivity to quantify the efficiency of a GA*: Goldberg's deceptivity analysis is based on the schema theorem which models the action of genetic operators in a "negative" way, i.e. only the destruction of schemata by genetic operators is considered (this is the reason why we have an inequality in the schema theorem). More recent approaches, as Price 's theorem [3], or Markov-based modelling [5], seem to be of interest in the framework of Hölder functions.

References

[1] E. Aarts and P. Van Laarhoven. Simulated annealing : a pedestrian review of the theory and some applications. *AI Series* F30, NATO.

[2] J. Albert, F. Ferri, J. Domingo, and M. Vincens. An Approach to Natural Scene Segmentation by Means of Genetic Algoritms with Fuzzy Data. In *Pattern Recognition and Image Analysis*, pages 97–113, 1992. Selected papers of the 4th Spanish Symposium (Sept 90), Perez de la Blanca Ed.

[3] L. Altenberg. The Schema Theorem and Price's Theorem. In *Foundation of Genetic Algorithms 3*, 1994.

[4] P. Andrey and P Tarroux. Unsupervised image segmentation using a distributed genetic algorithm. *Pattern Recognition*, 1993.

[5] R. Cerf. Asymptotic Convergence of Genetic Algorithms. *PhD thesis, Université Montpellier II*, 1994.

[6] Y. Davidor. Genetic Algorithms and Robotics. A heuristic Strategy for Optimization. World Scientific, 1990. *World Scientific Series in Robotics and Automated Systems - vol 1.*

[7] T. E. Davis and J. 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.

[8] K. J. Falconer. Fractal Geometry : Mathematical Foundation and Applications. *John Wiley & Sons, 1990.*

[9] D. A. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison-Wesley, January 1989.*

[10] D. E. Goldberg. Genetic Algorithms and Walsh functions : Part I, a gentle introduction. *TCGA Report No. 88006*, University of Alabama, Tuscaloosa, US, 1988.

[11] D. E. Goldberg. Genetic Algorithms and Walsh functions : Part II, deception and its analysis. *TCGA Report No. 89001*, University of Alabama, Tuscaloosa, US, 1989.

[12] 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.

[13] 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, Hillsdale, New Jersey, 1987. Lawrence Erlbaum Associates.

[14] A. Hill and C. J. Taylor. Model-Based Image Interpretation using Genetic Algorithms. *Image and Vision Computing*, 10(5):295–301, June 1992.

[15] J. H. Holland. Adaptation in Natural and Artificial System. *Ann Arbor, University of Michigan Press*, 1975.

[16] A. Homaifar and X. Qi. Analysis of Genetic Algorithms Deception by Hadamard Transform. In *International symposium machine learning and neuronal networks*, pages 75–78, October 1990. org. by IASTED.

[17] S. Khuri. Walsh and Haar functions in Genetic Algorithms. 1993. *San Jose State University.*

[18] J. R. Koza. Genetic Programming. *MIT Press*, 1992.

[19] E. Lutton and P. Martinez. A genetic algorithm for the detection of 2d geometric primitives in images. In *12-ICPR*, 1994. Jerusalem, Israel, 9-13 October.

[20] B.B. Mandelbrot and J.W. Van Ness. Fractional Brownian Motion, fractional Gaussian noises and applications. *SIAM Review 10*, 4:422–437, 1968.

[21] D. J. Nettleton and R. Garigliano. Evolutionary algorithms and a fractal inverse problem. *Biosystems*, (33):221–231, 1994. Technical note.

[22] G. Roth and M. D. Levine. Geometric Primitive Extraction Using a Genetic Algorithm. In *IEEE Computer Society Conference on CV and PR*, pages 640–644, 1992.

[23] P. Trompette, J. L. Marcelin, and C. Schmelding. Optimal damping of viscoelastic constrained beams or plates by use of a genetic algotithm. In *IUTAM*, 1993. Zakopane Pologne.

[24] S. Truvé. Using a Genetic Algorithm to solve Constraint Satisfation Problems Generated by an Image Interpreter. In *Theory and Applications of Image Analysis : 7th Scandinavian Conference on Image Analysis*, pages 378–386, August 1991. Aalborg (DK).

A Expression of the Haar functions in the Walsh basis

For every real function f, defined on $[0..2^l-1]$, let:

$$\forall x \in [0..2^l-1] \qquad f(x) = \sum_{j=0}^{2^l-1} \omega_j \Psi_j(x)$$

with $\Psi_j(x) = \prod_{t=0}^{l-1}(-1)^{x_t j_t}$, x_t and j_t being the values of the t^{th} bit of the binary decomposition of x and j. We will sometimes write $\Psi_j^l(x)$ to emphasize the dependance on l.

The Walsh coefficients are given by:

$$\omega_j = \frac{1}{2^l}\sum_{x=0}^{2^l-1} f(x)\Psi_j(x)$$

Let m_t and k_t be the t^{th} bit of the binary decomposition of m and k : $m = \sum_{t=0}^{l-1} m_t 2^t$. and $k = \sum_{t=0}^{l-1} k_t 2^t$

Proposition 4 *Every function H_j can be decomposed in the Walsh basis as follows:*

$$\boxed{H_j(x) = \tfrac{1}{2^q}(\textstyle\sum_{k=0}^{2^q-1}(-1)^{\sum_{t=0}^{l-1} m_t k_t}\Psi_{2^{l-q-1}+k2^{l-q}}(x))} \tag{11}$$

with $j = 2^q + m, \qquad q \in [0..l-1]$ *and* $m \in [0..2^q-1]$.

Proof:

Let T_j be the righthand side term of (11):

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

with $j = 2^q + m$ and $m \in [0..2^q-1]$.
We have to prove:

$$T_j(x) = H_j(x) = \begin{cases} 1 & \text{if } (2m)2^{l-q-1} \le x < (2m+1)2^{l-q-1} \\ -1 & \text{if } (2m+1)2^{l-q-1} \le x < (2m+2)2^{l-q-1} \\ 0 & \text{else} \end{cases}$$

Define: $j' = 2^{l-q-1} + k2^{l-q}$. We have:

$$\Psi_{2^{l-q-1}+k2^{l-q}}(x) = \Psi_{j'}(x) = (-1)^{\sum_{t=0}^{l-1} x_t j'_t}$$

with $j' = \sum_{t=0}^{l-1} j'_t 2^t = 2^{l-q-1} + k2^{l-q} = 2^{l-q-1} + \sum_{t=0}^{q-1} k_t 2^{t+(l-q)}$
and:

$$\begin{cases} j'_t = 0 & \text{if } t \in [0..l-q-2] \\ j'_{l-q-1} = 1 & \\ j'_t = k_{t-(l-q)} & \text{if } t \in [l-q..l-1] \end{cases}$$

Thus:

$$\Psi_{j'}(x) = (-1)^{x_{l-q-1}+\sum_{t=l-q}^{l-1} k_{t-(l-q)} x_t}$$

Replacing in the formula giving T_j :

$$T_j(x) = \frac{1}{2^q} \sum_{k=0}^{2^q-1} (-1)^{\sum_{t=l-q}^{l-1} m_t k_t + x_{l-q-1} + \sum_{t=0}^{l-1} k_{t-(l-q)} x_t}$$

$$T_j(x) = \frac{(-1)^{x_{l-q-1}}}{2^q} \sum_{k=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} k_t (m_t + x_{t+(l-q)})}$$

We consider two cases:

1 $\underline{x \in [m2^{l-q}..(m+1)2^{l-q}[}$

We may write: $x = m2^{l-q} + \alpha$, $\alpha \in [0..2^{l-q}[$ with $\alpha = \sum_{t=0}^{l-q-1} \alpha_t 2^t$.
Thus:

$$x \;=\; \sum_{t=0}^{q-1} m_t 2^{l-q+t} + \sum_{t=0}^{l-q-1} \alpha_t 2^t \;=\; \sum_{t=l-q}^{l-1} m_{t-(l-q)} 2^t + \sum_{t=0}^{l-q-1} \alpha_t 2^t$$

$$\forall\ t \in [l-q..l-1] \ , \ x_t = m_{t-(l-q)}$$

$$T_j(x) = \frac{(-1)^{x_{l-q-1}}}{2^q} \sum_{k=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} k_t (2m_t)}$$

Since $\sum_{t=0}^{q-1} k_t(2m_t)$ is even, the expression $(-1)^{\sum_{t=0}^{q-1} k_t(2m_t)}$ equals 1, and:

$$T_j(x) = (-1)^{x_{l-q-1}}$$

Thus:

1. if $x \in [(2m)2^{l-q-1}..(2m+1)2^{l-q-1}[$ $\quad T_j(x) = 1$ because $x_{l-q-1} = 0$

2. if $x \in [(2m+1)2^{l-q-1}..(2m+2)2^{l-q-1}[$ $\quad T_j(x) = -1$ because $x_{l-q-1} = 1$

2 $\underline{x \notin [m2^{l-q}..(m+1)2^{l-q}[}$

This is equivalent to:

$$\exists t \in [l-q..l-1] \quad \text{such that} \quad x_t \neq m_{t-(l-q)}$$

$$\Longleftrightarrow \exists t \in [0,..-1] \quad \text{such that} \quad x_{t+(l-q)} \neq m_t$$

using

$$\forall t, \quad m_t \in \{0,1\} \quad \text{and} \quad x_t \in \{0,1\}$$

we get:

$$m_t \neq x_{t+(l-q)} \Longrightarrow m_t + x_{t+(l-q)} = 1$$

Let us define T_1 as:

$$T_1 = \{t \in [0..q-1] \ \ /m_t + x_{t+(l-q)} = 1\}$$

We know that $T_1 \neq \emptyset$, thus:

$$T_j(x) = \frac{(-1)^{x_{l-q-1}}}{2^q}[\sum_{k=0}^{2^q-1}(-1)^{\sum_{t\in T_1} k_t}]$$

Let $b \in [0..2^l-1]$ be such that:

$$\begin{cases} b_t = 1 & t \in T_1 \\ b_t = 0 & t \notin T_1 \end{cases}$$

Then:

$$T_j(x) = \frac{(-1)^{x_{l-q-1}}}{2^q}[\sum_{k=0}^{2^q-1}(-1)^{\sum_{t=0}^{q-1} b_t k_t}]$$

The term $(-1)^{\sum_{t=0}^{q-1} b_t k_t}$ is equal to $\Psi_b^q(k)$. We can thus write:

$$T_j(x) = \frac{(-1)^{x_{l-q-1}}}{2^q}[\sum_{k=0}^{2^q-1} \Psi_b^q(k)]$$

The term $\sum_{k=0}^{2^q-1} \Psi_b^q(k)$ is zero if b is not 0, which is the case here, because there is at least one bit of b that equals 1.
We finally obtain:

$$x \notin [m2^{l-q}..(m+2)2^{l-q}] \quad \Rightarrow \quad T_j(x) = 0.$$

□

B Expression of the Walsh functions in the Haar basis

For $i = 2^q + m$, $q \in [0..l-1]$ and $m \in [0..2^q-1]$, we have:

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

.

The coefficients of transformation matrix between Walsh and Haar bases are thus:

$$\begin{cases} m_{ij} = 0 & \text{if} \quad j \neq 2^{l-q-1} + k2^{l-q} \\ m_{ij} = \frac{1}{2^q}(-1)^{\sum_{t=0}^{q-1} m_t k_t} & \text{else} \end{cases}$$

$$H_i(x) = \sum_{j=0}^{2^{l-1}} m_{ij}\Psi_j(x)$$

As the two bases are othogonal bases, which are non-orthonormal, the inverse formula is:

$$\Psi_i(x) = \sum_{j=0}^{2^{l-1}} m_{ji}\frac{||\Psi_i||^2}{||H_j||^2}H_j(x)$$

with:

$$||\Psi_i||^2 = \sum_{x=0}^{x=2^l-1} [\Psi_i(x)]^2 = 2^l$$

$$||H_i||^2 = \sum_{x=0}^{x=2^l-1} [H_i(x)]^2 = \sum_{x=m2^{l-q}}^{x=(m+1)2^{l-q}} 1 = 2^{l-q}, \quad \text{because} \quad i = 2^q + m$$

Thus

$$\Psi_i(x) = \sum_{j=0}^{2^{l-1}} 2^q m_{ji} H_j(x)$$

For every integer $i \in [1..2^l - 1]$, whose binary decomposition is $i = \sum_{t=0}^{l-1} i_t 2^t$, there exists a unique couple $(q, k), k \in [0..2^q - 1]$ such that:

$$i = 2^{l-q-1} + k2^{l-q}$$

In the expression of Ψ_i, the only m_{ji} coefficients which are non zero correspond to the $j = 2^q + m$ such that:

$$\exists k \in [0..2^q - 1] \text{ such that } i = 2^{l-q-1} + k2^{l-q}$$

For $q > 0$:

$$i = \sum_{t=0}^{l-1} i_t 2^t = 2^{l-q-1} + \sum_{t=0}^{q-1} k_t 2^{t+(l-q)} = 2^{l-q-1} + \sum_{t=l-q}^{l-1} k_{t-(l-q)} 2^t$$

- if $t \in [0..l - q - 1[\quad i_t = 0$
- if $t = l - q - 1 \quad i_t = 1$
- if $t \in [l - q..l - 1] \quad i_t = k_{t-(l-q)}$

$l - q - 1$ is the first non-zero bit of i.
Thus:

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

Remark: this relation also holds for $q = 0$ (in this case $m = 0$), with the convention $\sum_{t=0}^{t=-1} \star = 0$.
Finally:

$$\boxed{\Psi_i(x) = \sum_{m=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} k_t m_t} H_{2^q+m}(x) \quad \text{with} \quad i = 2^{l-q-1}(1 + 2k), \quad k \in [0..2^q - 1], \quad \text{and} \quad q \in [0..l - 1]}$$

C Expression of the Haar coefficients as a function of the Walsh coefficients and conversely

For any function f defined on $[0..2^l - 1]$, write:

$$f(x) = \sum_{i=0}^{2^l-1} \omega_i \Psi_i(x) = \sum_{j=0}^{2^l-1} h_j H_j(x)$$

with $\quad h_j = \frac{1}{2^{l-q}} \sum_{x=0}^{2^l-1} f(x) H_j(x) \quad$ and $\quad \omega_i = \frac{1}{2^l} \sum_{x=0}^{2^l-1} f(x) \Psi_i(x)$.

Thus:

$$h_j = \frac{1}{2^{l-q}} \sum_{x=0}^{2^l-1} (\sum_{k=0}^{2^l-1} \omega_k \Psi_k(x)) H_j(x)$$

Using the expression of H_j in the Walsh basis:

$$h_j = \frac{1}{2^{l-q}} \sum_{x=0}^{2^l-1} (\sum_{k=0}^{2^l-1} \omega_k \Psi_k(x)) (\frac{1}{2^q} \sum_{v=0}^{2^q-1} (-1)^{\sum_{t=0}^{l-1} m_t v_t} \Psi_{2^{l-q-1}+v2^{l-q}}(x))$$

$$h_j = \frac{1}{2^{l-q}} \frac{1}{2^q} \sum_{v=0}^{2^q-1} (-1)^{\sum_{t=0}^{l-1} m_t v_t} (\sum_{k=0}^{2^l-1} \omega_k \sum_{x=0}^{2^l-1} \Psi_k(x) \Psi_{2^{l-q-1}+v2^{l-q}}(x))$$

The Ψ_j form an othogonal basis:

$$\sum_{x=0}^{2^l-1} \Psi_i(x) \Psi_j(x) = \begin{cases} 2^l & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

We obtain:

$$\boxed{h_j = \sum_{v=0}^{2^q-1} (-1)^{\sum_{t=0}^{l-1} m_t v_t} \omega_{2^{l-q-1}+v2^{l-q}} \qquad \text{with} \quad j = 2^q + m}$$

We now move to the Walsh coefficients:

$$\omega_i = \frac{1}{2^l} \sum_{x=0}^{2^l-1} f(x) \Psi_i(x) \quad \text{with} \quad i = 2^{l-q-1}(1+2k)$$

$$\omega_i = \frac{1}{2^l} \sum_{x=0}^{2^l-1} (\sum_{v=0}^{2^l-1} h_v H_v(x)) \Psi_i(x)$$

$$\omega_i = \frac{1}{2^l} \sum_{x=0}^{2^l-1} (\sum_{v=0}^{2^l-1} h_v H_v(x)) (\sum_{m=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} m_t k_t} H_{2^q+m}(x))$$

$$\omega_i = \frac{1}{2^l} (\sum_{m=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} m_t k_t} \sum_{v=0}^{2^l-1} h_v \sum_{x=0}^{2^l-1} H_v(x) H_{2^q+m}(x))$$

$$\boxed{\omega_i = \frac{1}{2^q} \sum_{m=0}^{2^q-1} h_{2^q+m} (-1)^{\sum_{t=0}^{q-1} m_t k_t} \qquad \text{with} \quad i = 2^{l-q-1}(1+2k)}$$

D Computation of the Haar adjusted coefficients

Let :

$$f(x) = \sum_{i=0}^{2^l-1} \omega_i \Psi_i(x) = \sum_{j=0}^{2^l-1} h_j H_j(x)$$

and :

$$f'(x) = \sum_{i=0}^{2^l-1} \omega'_i \Psi_i(x) = \sum_{j=0}^{2^l-1} h'_j H_j(x)$$

We write :

$i = 2^{l-q-1}(1+2k)$, $q \in [0..l-1]$, $k \in [0..2^q-1]$
$j = 2^q + m$, $q \in [0..l-1]$, $m \in [0..2^q-1]$.

Then :

$$f'(x) \quad = \quad \omega'_0 + \sum_{q=0}^{l-1} \sum_{k=0}^{2^q-1} \omega'_i \sum_{m=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} m_t k_t} H_{2^q+m}(x) \quad = \quad h'_0 + \sum_{q=0}^{l-1} \sum_{m=0}^{2^q-1} h'_{2^q+m} H_{2^q+m}(x)$$

In the following, the subscript t indicates the t^{th} bit of the binary decomposition, i.e. $k = \sum_{t=0}^{t=l-1} k_t 2^t$, $m = \sum_{t=0}^{t=l-1} m_t 2^t$, etc ...

$$f'(x) = \omega'_0 + \sum_{q=0}^{l-1} \sum_{m=0}^{2^q-1} [\sum_{k=0}^{2^q-1} \omega'_{2^{l-q-1}(1+2k)} (-1)^{\sum_{t=0}^{q-1} m_t k_t}] H_{2^q+m}(x) = h'_0 + \sum_{q=0}^{l-1} \sum_{m=0}^{2^q-1} h'_{2^q+m} H_{2^q+m}(x)$$

$$h'_{2^q+m} = \sum_{k=0}^{2^q-1} \omega'_{2^{l-q-1}(1+2k)} (-1)^{\sum_{t=0}^{q-1} m_t k_t}$$

(we have : $\omega'_0 = \omega_0 = h_0 = h'_0$)
Using the expression of the ω'_i (equation (5)) :

for $q = 0$:

$$\omega'_{2^{l-1}} = \omega_{2^{l-1}}(1 - 2p_m) = h'_1$$

thus :

$$h'_1 = h_1(1 - 2p_m)$$

for $q > 0$:

$$h'_{2^q+m} = \sum_{k=0}^{2^q-1} \omega_{2^{l-q-1}(1+2k)} (-1)^{\sum_{t=0}^{q-1} m_t k_t} [1 - p_c \frac{\delta(2^{l-q-1}(1+2k))}{l-1} - 2p_m \mathcal{O}(2^{l-q-1}(1+2k))]$$

Then :

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

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

- It is obvious that: $\delta(2^{l-q-1}(1+2k)) = \delta^*(k)+1$
  $\delta^*(k)$ being the position of the last non-zero bit of k.
  For $k=0$, we have $\delta(2^{l-q-1})=0$. We thus define: $\delta^*(0)=-1$
- We also have: $\mathcal{O}(2^{l-q-1}(1+2k)) = 1+\mathcal{O}(k)$

h'_{2^q+m} thus becomes:

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

and finally:

$$\begin{aligned} h'_{2^q+m} &= h_{2^q+m}(1-\frac{p_c}{l-1}-2p_m) \\ &-\frac{p_c}{2^q(l-1)}\sum_{m'=0}^{2^q-1} h_{2^q+m'}\sum_{k=0}^{2^q-1}\delta^*(k)(-1)^{\sum_{t=0}^{q-1}(m_t+m'_t)k_t} \\ &-\frac{2p_m}{2^q}\sum_{m'=0}^{2^q-1} h_{2^q+m'}\sum_{k=0}^{2^q-1}\mathcal{O}(k)(-1)^{\sum_{t=0}^{q-1}(m_t+m'_t)k_t}. \end{aligned}$$

Let us define $\Delta(m,m')$ and $\mathcal{O}(m,m')$ as:

$$\begin{aligned} \Delta(m,m') &= \sum_{k=0}^{2^q-1}\delta^*(k)(-1)^{\sum_{t=0}^{q-1}(m'_t+m_t)k_t} \\ &= -1+\sum_{k=1}^{2^q-1}\delta^*(k)(-1)^{\sum_{t=0}^{\delta^*(k)}(m'_t+m_t)k_t} \end{aligned}$$

and

$$\begin{aligned} \mathcal{O}(m,m') &= \sum_{k=0}^{2^q-1}\mathcal{O}(k)(-1)^{\sum_{t=0}^{q-1}(m'_t+m_t)k_t} \\ &= \sum_{k=0}^{2^q-1}[\sum_{t=0}^{q-1}k_t](-1)^{\sum_{t=0}^{q-1}(m'_t+m_t)k_t} \end{aligned}$$

Write $k=2^d+b$, or :

$$k = 2^d + \sum_{t=0}^{d-1} b_t 2^t,$$

$$\text{thus} \qquad \delta^*(k) = d$$

◇ **Computation of $\Delta(m,m')$:**

$$\Delta(m,m') = -1+\sum_{d=0}^{q-1}\sum_{b=0}^{2^d-1} d(-1)^{\sum_{t=0}^{q-1}(m'_t+m_t)k_t}$$

$$\Delta(m,m') = -1+\sum_{d=0}^{q-1} d(-1)^{m'_d+m_d}\sum_{b=0}^{2^d-1}(-1)^{\sum_{t=0}^{q-1}(m'_t+m_t)b_t}$$

$\sum_{b=0}^{2^d-1}(-1)^{\sum_{t=0}^{q-1}(m'_t+m_t)b_t}$ corresponds to:

$$\sum_{b=0}^{2^d-1} \Psi_m^d(b)\Psi_{m'}^d(b) = \begin{cases} 0 & \text{if the first } d \text{ bits of } m \text{ and } m' \text{ are the same} \\ 2^d & \text{else} \end{cases}$$

where Ψ_m^d is the restriction of the m^{th} Walsh function on d bits, i.e. to the set $[0..2^d-1]$. Thus:

1. if $\forall\ t \in [0..q-1]\quad m_t \neq m'_t$ then $\Delta(m,m') = -1$
2. if $m = m'$ then $\Delta(m,m') = -1 + \sum_{d=0}^{q-1} d2^d = 1 + q2^q - 2^{q+1}$
3. Let us define u such that $\forall t \in [0..u-1]$, $m_t = m'_t$, and $m_u \neq m'_u$ (i.e. $m_u + m'_u = 1$). Then:

$$\Delta(m,m') = -1 + \sum_{d=0}^{q-1} d(-1)^{m_d+m'_d} \sum_{b=0}^{2^d-1} [\Psi_m^d(b)\Psi_{m'}^d(b)]$$

$$\begin{aligned} \Delta(m,m') \quad = \quad & -1 + \sum_{d=0}^{u-1} d(-1)^{m'_d+m_d}[\sum_{b=0}^{2^d-1} \Psi_m^d(b)\Psi_{m'}^d(b)] \\ & +u(-1)^{m'_u+m_u}[\sum_{b=0}^{2^u-1} \Psi_m^u(b)\Psi_{m'}^u(b)] \\ & + \sum_{d=u+1}^{q-1} d(-1)^{m'_d+m_d}[\sum_{b=0}^{2^d-1} \Psi_m^d(b)\Psi_{m'}^d(b)] \end{aligned}$$

$$\Delta(m,m') = -1 + \sum_{d=0}^{u-1} d2^d - u2^u = 1 - 2^{u+1}$$

Finally: denoting u the integer such that $\forall t \in [0..u-1]$, $m_t = m'_t$ and $m_u \neq m'_u$, the three cases above are summarized as:

$$\boxed{\begin{array}{lcl} \text{if } u \in [0..q-1] & \longrightarrow & \Delta(m,m') = 1 - 2^{u+1} \\ \text{if } u = q \text{ (i.e. } m = m') & \longrightarrow & \Delta(m,m') = 1 + q2^q - 2^{q+1} \end{array}}$$

◇ **Computation of $\mathcal{O}(m,m')$:**

1. If $m = m'$: $\quad (-1)^{\sum_{t=0}^{q-1}(m'_t+m_t)k_t} = 1$, and:

$$\mathcal{O}(m,m) = \sum_{k=0}^{2^q-1} \mathcal{O}(k)$$

Set $s = \mathcal{O}(k)$, with $k \in [0..2^q-1]$. We obtain:

$$\mathcal{O}(m,m) = \sum_{s=0}^{q} C_q^s s = q2^{q-1}$$

2. If $\forall t \in [0..q-1] \quad m_t \neq m'_t$:

$$\mathcal{O}(m, m') = \sum_{k=0}^{2^q-1} \mathcal{O}(k)(-1)^{\mathcal{O}(k)} = \sum_{s=0}^{q} C_q^s (-1)^s = 0$$

3. In the general case, we have:

$$\mathcal{O}(m, m') = \sum_{t=0}^{q-1} (-1)^{m_t + m'_t} \prod_{v=0, v \neq t}^{q-1} (1 + (-1)^{m_v + m'_v}) \tag{12}$$

Proof: For $q = 1$, this equality is obvious, and we prove the formula by induction: suppose it is true for q, then:

$$\forall m, m' \in [0..2^{q+1}] \qquad \mathcal{O}^{q+1}(m, m') = \sum_{k=0}^{2^{q+1}-1} \sum_{t=0}^{q} k_t (-1)^{\sum_{t=0}^{q} (m_t + m'_t) k_t}$$

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

In the term $\sum_{k=2^q}^{2^{q+1}-1} \sum_{t=0}^{q} k_t (-1)^{\sum_{t=0}^{q} (m_t + m'_t) k_t}$, let us write $k = 2^q + f$ with $f \in [0..2^q - 1]$, i.e. $k_t = f_t \ \ \forall t \in [0..q-1]$.

$$\begin{aligned} \mathcal{O}^{q+1}(m, m') &= \mathcal{O}^q(m, m') \\ &+ (-1)^{m_q + m'_q} \sum_{f=0}^{2^q-1} (1 + \sum_{t=0}^{q-1} f_t) (-1)^{\sum_{t=0}^{q-1} (m_t + m'_t) f_t} \end{aligned}$$

$$\begin{aligned} \mathcal{O}^{q+1}(m, m') &= \mathcal{O}^q(m, m') + (-1)^{m_q + m'_q} \mathcal{O}^q(m, m') \\ &+ (-1)^{m_q + m'_q} \sum_{f=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} (m_t + m'_t) f_t} \end{aligned}$$

We have to prove that:

$$\sum_{f=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} (m_t + m'_t) f_t} = \prod_{v=0}^{q-1} (1 + (-1)^{m_v + m'_v})$$

This is obviously true for $q = 1$ and $q = 2$. Define $S_q = \sum_{f=0}^{2^q-1} (-1)^{\sum_{t=0}^{q-1} (m_t + m'_t) f_t}$. Then

$$\begin{aligned} S_{q+1} &= \sum_{f=0}^{2^{q+1}-1} (-1)^{\sum_{t=0}^{q} (m_t + m'_t) f_t} \\ &= \sum_{f=0}^{2^q-1} (-1)^{\sum_{t=0}^{q} (m_t + m'_t) f_t} + \sum_{f=2^q}^{2^{q+1}-1} (-1)^{\sum_{t=0}^{q} (m_t + m'_t) f_t} \\ &= S_q + (-1)^{m_q + m'_q} S_q = (1 + (-1)^{m_q + m'_q}) S_q \end{aligned}$$

We thus obtain for $\mathcal{O}^{q+1}(m,m')$:

$$\begin{aligned}\mathcal{O}^{q+1}(m,m') &= \mathcal{O}^{q}(m,m')(1+(-1)^{m_q+m'_q}) + (-1)^{m_q+m'_q}\prod_{v=0}^{q-1}(1+(-1)^{m_v+m'_v}) \\ &= \sum_{t=0}^{q-1}(-1)^{m_t+m'_t}\prod_{v=0,v\neq t}^{q}(1+(-1)^{m_v+m'_v}) + (-1)^{m_q+m'_q}\prod_{v=0,v\neq t}^{q-1}(1+(-1)^{m_v+m'_v})\end{aligned}$$

□

Now :

- If $m=m'$: $\forall t \ \ (-1)^{m_t+m'_t}=1$ then : $\mathcal{O}(m,m')=\sum_{t=0}^{q-1}\prod_{v=0,v\neq t}^{q-1}2=q2^{q-1}$
- Let u be the number of bits where m and m' differ : if $u=1$, then :
  - $\exists t_0$ such that $m_{t_0}+m'_{t_0}=1$ and then $(1+(-1)^{m_{t_0}+m'_{t_0}})=0$,
  - $\forall t\neq t_0 \quad m_t+m'_t=0$ or 2 and then all the terms $\prod_{v=0,v\neq t}^{q-1}(1+(-1)^{m_v+m'_v})=0$

  Thus

  $$\mathcal{O}(m,m')=(-1)^{m_{t_0}+m'_{t_0}}\prod_{v=0,v\neq t_0}^{q-1}(1+(-1)^{m_v+m'_v})=-2^{q-1}$$

  If $u>1$, let T_u be the subset of $[0..q-1]$ such that $t\in T_u$ iff $m_t+m'_t=1$. Then :

  $$\begin{aligned}\text{if} t\in T_u, m_t+m'_t=1, &\qquad \text{and}[(1+(-1)^{m_t+m'_t})]=0\\ \text{if} t\notin T_u, m_t+m'_t=0 \text{ or } 2, &\qquad \text{and}[(1+(-1)^{m_t+m'_t})]=2\end{aligned}$$

  Thus $\mathcal{O}(m,m')=0$

Finally :

if m and m' differ by more than 1 bit,	$\mathcal{O}(m,m')=0$,
if m and m' differ by 1 bit,	$\mathcal{O}(m,m')=-2^{q-1}$,
if $m=m'$,	$\mathcal{O}(m,m)=q2^{q-1}$

Recall that h'_{2^q+m} can be written as :

$$\begin{aligned}h'_{2^q+m} &= h_{2^q+m}(1-\frac{p_c}{l-1}-2p_m)\\ &\quad -\frac{p_c}{2^q(l-1)}\sum_{m'=0}^{2^q-1}h_{2^q+m'}\Delta(m,m')\\ &\quad -\frac{2p_m}{2^q}\sum_{m'=0}^{2^q-1}h_{2^q+m'}\mathcal{O}(m,m')\end{aligned}$$

◇ the $\mathcal{O}(m, m')$ term yields:

$$\sum_{m'=0}^{2^q-1} h_{2^q+m'}\mathcal{O}(m,m') = q2^{q-1}h_{2^q+m} \quad - \quad 2^{q-1} \sum_{m'/\exists u, |m'-m|=2^u} h_{2^q+m'}$$

since m and m' differ only by 1 bit: $\exists u, \; m' = m + (1-2m_u)2^u$

Thus:

$$\sum_{m'=0}^{2^q-1} h_{2^q+m'}\mathcal{O}(m,m') = q2^{q-1}h_{2^q+m} - 2^{q-1}\sum_{t=0}^{q-1} h_{2^q+m+(1-2m_t)2^t}$$

◇ the $\Delta(m, m')$ term yields:

$$\sum_{m'=0}^{2^q-1} h_{2^q+m'}\Delta(m,m') = [1+(q-2)2^q]h_{2^q+m} + \sum_{m'=0, m'\neq m}^{2^q-1} h_{2^q+m'}\Delta(m,m')$$

for $m' \neq m$, $\exists u / \forall t \in [0..u-1] \quad m_t = m'_t$ and $m_u \neq m'_u$ (i.e. $m'_u = 1 - m_u$).

We can thus write:

$$m' = \sum_{t=0}^{u-1} m_t 2^t + (1-m_u)2^u + \sum_{t=u+1}^{q-1} m'_t 2^t \qquad u \in [0..q-1]$$

And:

$$\sum_{m'=0}^{2^q-1} h_{2^q+m'}\Delta(m,m') = [1+(q-2)2^q]h_{2^q+m} + \sum_{u=0}^{q-1}(1-2^{u+1}) \sum_{r=0}^{2^{q-u-2}} h_{2^q+\sum_{t=0}^{u-1} m_t 2^t + (1-m_u)2^u + r2^{u+1}}$$

Finally, h'_{2^q+m} can be written as:

$$\boxed{\begin{array}{lll} h'_{2^q+m} & = & h_{2^q+m}[1 - \frac{p_c}{l-1}(1 + \frac{1+(q-2)2^q}{2^q}) - 2p_m(1+\frac{q}{2})] \\ & - & \frac{p_c}{2^q(l-1)} \sum_{u=0}^{q-1}(1-2^{u+1}) \sum_{r=0}^{2^{q-u-2}} h_{2^q+\sum_{t=0}^{u-1} m_t 2^t + (1-m_u)2^u + r2^{u+1}} \\ & - & p_m \sum_{t=0}^{q-1} h_{2^q+m+(1-2m_t)2^t} \end{array}}$$

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, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
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)
ISSN 0249-6399