Computing and Informatics, Vol. 24, 2005, 263–279

COMPUTING EPISTASIS OF TEMPLATE FUNCTIONS THROUGH WALSH TRANSFORMS*

María Teresa Iglesias, Concepción Vidal

Universidade da Coruña
Facultad de Informática
A Coruña, Spain
e-mail: totero@udc.es, conchi@dc.fi.udc.es

Alain Verschoren

University of Antwerp
Department of Mathematics and Computer Science
Antwerp, Belgium
e-mail: alain.verschoren@ua.ac.be

Manuscript received 21 May 2004; revised 2 May 2005

Abstract. Template functions have been introduced as a class of test functions, allowing to study the convergence behaviour of genetic algorithms. In this note, we show how to use Walsh transforms to calculate the normalized epistasis of these functions.

Keywords: Genetic algorithm, GA hardness, epistasis, Walsh transform, Fourier transform, template function

1 INTRODUCTION

It has long been accepted by the genetic algorithm community that there are many factors, including deception and high epistasis, which may make a fitness function acting on binary strings hard to optimize.

* Research supported by a research grant of Xunta de Galicia PGIDT03PXIA10502PR and the GOA "Generic Optimization" at the University of Antwerp.

The notion of epistasis measures, roughly speaking, the presence of links between separate bits in the codified version of the data to be optimized, whereas deception essentially deals with misleading data present in the underlying fitness landscape, leading the optimization process away from the global optimum.

Both deception and epistasis are by no means sufficient to predict GA-hardness (the difficulty of a fitness function to be optimized with a genetic algorithm). On the other hand, it has experimentally been shown in [9], for example, that there is a strong correlation between epistasis and GA-hardness for several classes of functions, which may be described by a limited number of control parameters. The Royal Road functions [4, 9] and the Template functions [7] studied in this note are of this type.

Template functions are controlled by just two parameters, the length of the strings we are dealing with and the length of the "template", whose presence in a chain increases the fitness value of the latter. These Template functions have been used in [7] as a laboratory to experiment with, while studying GA-hardness. Let us already point out here that increasing the length of the template also increases the epistasis value of the function, as well as its GA-hardness.

On the other hand, Walsh transforms have been introduced in the framework of genetic algorithms by Goldberg [6] and may be viewed as a binary analogue of ordinary Fourier transforms. The associated Walsh coefficients include in a very natural way the basic properties of the schemata and, in our context, they allow for a very efficient calculation of the normalized epistasis of Template functions.

In the first section of this note, we present the background needed for the formal study of the notion of epistasis and we introduce the Walsh coefficients of any fitness function f. We show how these coefficients may be used to calculate the normalized epistasis of f. In the second section, we introduce the Template functions we just referred to and we show how the use of Walsh transforms allows to easily calculate the normalized epistasis of these functions. In the last section, we show experimentally that the epistasis and GA-hardness of Template functions are strongly related providing extra evidence of the claims made before. Finally, in an appendix, we briefly sketch how the alternative approach in [7] leads to similar results, albeit in a much more complicated and technical way.

2 EPISTASIS AND WALSH TRANSFORMS

As pointed out in the introduction, epistasis essentially describes the dependency or independency of bits in strings to which a fitness function is applied. As in [10], one speaks of *minimal epistasis*, when every gene (bit) is independent of every other gene, and of *maximal epistasis*, when no proper subset of genes is independent of any other gene.

A quantitative approach to this concept is given in [3], where Davidor defines the epistasis of a string $s = s_{\ell-1} \dots s_0 \in \Omega = \{0,1\}^\ell$ with respect to a fitness function f as

$$\varepsilon_\ell(s) = f(s) - \frac{1}{2^{\ell-1}} \sum_{i=0}^{\ell-1} \sum_{t\in\Omega(s_i,i)} f(t) + \frac{\ell-1}{2^\ell} \sum_{t\in\Omega} f(t),$$

with $\Omega(s_i, i)$ consisting of all strings in Ω which have value s_i at position i. When no ambiguity arises, we will leave out the subscript ℓ. The *global epistasis* of f is defined to be

$$\varepsilon(f) = \sqrt{\sum_{s\in\Omega} \varepsilon^2(s)}.$$

It has been pointed out in [12, 14] that these definitions may be rewritten as follows. Define the vectors

$$\boldsymbol{e} = \begin{pmatrix} \varepsilon(0\dots00) \\ \varepsilon(0\dots01) \\ \vdots \\ \varepsilon(11\dots1) \end{pmatrix} \text{ resp. } \boldsymbol{f} = \begin{pmatrix} f(0\dots00) \\ f(0\dots01) \\ \vdots \\ f(11\dots1) \end{pmatrix} = \begin{pmatrix} f_0 \\ \vdots \\ f_{2^\ell-1} \end{pmatrix}$$

and for any $0 \le i, j < 2^\ell$, put

$$e_{ij} = \frac{1}{2^\ell}(\ell + 1 - 2d_{ij}),$$

where d_{ij} is the Hamming distance between i and j (the number of bits in which the binary representations of i and j differ). Putting $\boldsymbol{E}_\ell = (e_{ij}) \in M_{2^\ell}(\mathbb{Q})$, the set of 2^ℓ-dimensional square matrices with entries in $\mathbb{Q}$, it is then easy to see that

$$\boldsymbol{e} = \boldsymbol{f} - \boldsymbol{E}_\ell \boldsymbol{f}.$$

It thus follows that $\varepsilon(f) = \|\boldsymbol{e}\|$ and, since $\boldsymbol{E}_\ell$ has been proved in [12] to be symmetric and idempotent, we also have that

$$\varepsilon^2(f) = {}^t\boldsymbol{f}(\boldsymbol{I}_\ell - \boldsymbol{E}_\ell)\boldsymbol{f},$$

where $\boldsymbol{I}_\ell$ is the identity matrix of dimension 2^ℓ, and where we denote for any matrix $\boldsymbol{A}$ by ${}^t\boldsymbol{A}$ its transpose.

It is obvious that for any $\alpha > 0$ we have $\varepsilon(\alpha f) = \alpha\varepsilon(f)$, whereas intuitively αf and f should have the same epistasis. In order to remedy this, the authors of [12] introduce the notion of *normalized epistasis* of f. This is defined as

$$\varepsilon^*(f) = \varepsilon^2\left(\frac{f}{\|\boldsymbol{f}\|}\right) = \frac{\varepsilon^2(f)}{\|\boldsymbol{f}\|^2} = \frac{{}^t\boldsymbol{f}(\boldsymbol{I}_\ell - \boldsymbol{E}_\ell)\boldsymbol{f}}{{}^t\boldsymbol{f}\boldsymbol{f}},$$

which implies that $0 \le \varepsilon^*(f) \le 1$. Actually, one may show that $\varepsilon^*(f) = 0$ if, and only if, f has minimal epistasis, in the sense of [10]. On the other hand, the theoretical maximum $\varepsilon^*(f) = 1$ cannot be reached by (positive valued!) fitness functions f;

in fact, the maximal value $1 - \frac{1}{2^{\ell-1}}$ corresponds exactly to the so-called "Camel functions", i.e., functions f with $f_i = f_{2^\ell - i-1} \neq 0$ for some $0 \leq i < 2^\ell$ and $f_j = 0$ elsewhere.

In order to simplify the calculation of epistasis, it is usually easier to work with the matrix $\boldsymbol{G}_\ell = 2^\ell \boldsymbol{E}_\ell \in M_{2^\ell}(\mathbb{Z})$ with entries $g_{ij} = \ell + 1 - 2d_{ij}$ for all $0 \leq i, j \leq 2^\ell - 1$, and with the function

$$\gamma_\ell(f) \quad = \quad {}^t\boldsymbol{f}\boldsymbol{G}_\ell\boldsymbol{f}.$$

Clearly $0 \leq \gamma_\ell(f) \leq 2^\ell$ and

$$\varepsilon^*(f) \quad = \quad 1 - \frac{{}^t\boldsymbol{f}\boldsymbol{E}_\ell\boldsymbol{f}}{{}^t\boldsymbol{f}\boldsymbol{f}} = 1 - \frac{1}{2^\ell}\gamma_\ell\left(\frac{f}{\|\boldsymbol{f}\|}\right) = 1 - \frac{\gamma_\ell(f)}{2^\ell \|\boldsymbol{f}\|^2}. \tag{1}$$

We will need some results from [8]. As before, let $\Omega = \{0,1\}^\ell$ denote the space of binary strings of length ℓ. For any $t \in \Omega$, we define the Walsh function ψ_t on Ω by putting

$$\psi_t(s) = \prod_{i=0}^{\ell}(1-2s_i)^t = (-1)^{s \cdot t} = \prod_{i=0}^{\ell-1}(-1)^{s_i t_i},$$

where $s \cdot t = \sum_{i=0}^{\ell-1} s_i t_i$ is the scalar product of $s, t \in \Omega$. for any $s_{\ell-1} \ldots s_0 \in \Omega$. It follows that ψ_t counts, for any string s, the number of ones situated at loci of s where t has also value one. The result is 1 or -1, depending on whether this number is even or odd. The Walsh functions may be represented by the matrix:

$$\boldsymbol{V}_\ell = (\psi_t(s))_{s,t\in\Omega} \in M_{2^\ell}(\mathbb{Z}).$$

For small values of ℓ, we have $\boldsymbol{V}_0 = (1)$ and:

$$\boldsymbol{V}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ resp. } \boldsymbol{V}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Moreover, it is easy to see that $\boldsymbol{V}_\ell$ may inductively be constructed by

$$\boldsymbol{V}_\ell = \begin{pmatrix} \boldsymbol{V}_{\ell-1} & \boldsymbol{V}_{\ell-1} \\ \boldsymbol{V}_{\ell-1} & -\boldsymbol{V}_{\ell-1} \end{pmatrix}.$$

From this, one easily deduces that $\boldsymbol{V}_\ell^2 = 2^\ell \boldsymbol{I}_\ell$. Sometimes it is easier to work with the matrix $\boldsymbol{W}_\ell = 2^{-\ell/2}\boldsymbol{V}_\ell$. It has the property that $\boldsymbol{W}_\ell^2 = \boldsymbol{I}_\ell$ and it satisfies the recursion relation

$$\boldsymbol{W}_\ell = 2^{-1/2}\begin{pmatrix} \boldsymbol{W}_{\ell-1} & \boldsymbol{W}_{\ell-1} \\ \boldsymbol{W}_{\ell-1} & -\boldsymbol{W}_{\ell-1} \end{pmatrix}.$$

The Walsh functions form a basis for the vector space of real valued functions on Ω. In fact, if we represent any such function f by the associated vector $\boldsymbol{f} \in \mathbb{R}^{2^\ell}$, then we define the Walsh transform w of f by $\boldsymbol{w} = \boldsymbol{W}_\ell \boldsymbol{f}$. The components $w_i = w_i(f)$ of $\boldsymbol{w}$ are called the *Walsh coefficients* of f and are (up to a factor $2^{-\ell/2}$) the coordinates of f with respect to the basis $\{\psi_t; t \in \Omega\}$. The Walsh coefficients of f, of course, easily permit to recover f, since it follows from $\boldsymbol{W}_\ell^2 = \boldsymbol{I}_\ell$ that

$$\boldsymbol{f} = \boldsymbol{W}_\ell(\boldsymbol{W}_\ell \, \boldsymbol{f}) = \boldsymbol{W}_\ell \, \boldsymbol{w}.$$

In order to calculate the normalized epistasis of f in terms of Walsh coefficients, let us define the diagonal matrix $\boldsymbol{D}_\ell$, whose only non-zero diagonal entries d_{ii} have value 1 and are situated at $i = 0$ and $i = 2^j$, for $0 \leq j \leq \ell - 1$. The proof of the following result can be found in [8].

Lemma 1. With notations as before, we have:

$$\boldsymbol{W}_\ell \, \boldsymbol{E}_\ell \, \boldsymbol{W}_\ell = \boldsymbol{D}_\ell.$$

Taking into account the above result and also (1), it can be easily proved that:

Proposition 1. If $w_0, \ldots, w_{2^\ell - 1}$ are the Walsh coefficients of the fitness function f, then the normalized epistasis $\varepsilon^*(f)$ of f is given by

$$\varepsilon^*(f) = 1 - \frac{w_0^2 + \sum_{i=0}^{\ell-1} w_{2^i}^2}{\sum_{j=0}^{2^\ell - 1} w_j^2}.$$

3 COMPUTING EPISTASIS THROUGH WALSH TRANFORMS

In this section, we show how the use of Walsh transforms permits an easy calculation of the epistasis of the Template functions. We invite the reader to compare it with the set-up in [7], which will briefly be sketched in an Appendix.

The "Template functions" we are about to consider calculate the fitness of a string of length ℓ, by sliding a fixed string t of length $n \leq \ell$ (the "template") over it. Each time an occurrence of t in s is found, a fixed amount a is added to the fitness of s. For convenience's sake, we will assume throughout that $a = 1$ and that t is the length n string $1^n = 11\ldots11$. So, the Template functions we will study in this note depend only on the parameters ℓ and n and will be denoted by T_ℓ^n. For example,

$$T_\ell^2(1^\ell) = T_\ell^2(11\ldots11) = \ell - 1.$$

It seems reasonable to expect that increasing the length n of the template will also increase the epistasis of T_ℓ^n, in view of the strong linkage between the different loci.

The first step of this consists in evaluating $\|\boldsymbol{T}_\ell^n\|$ where $\boldsymbol{T}_\ell^n \in \mathbb{R}^{2^\ell}$ denotes the vector corresponding to T_ℓ^n, i.e.,

$$\boldsymbol{T}_\ell^n = \begin{pmatrix} T_\ell^n(00\ldots0) \\ \vdots \\ T_\ell^n(11\ldots1) \end{pmatrix}.$$

To simplify this calculation, let us first note that, if $\ell < n$, then $\boldsymbol{T}_\ell^n \in \mathbb{R}^{2^\ell}$ is the zero vector whereas, for any $\ell \geq n$, we have

$$\boldsymbol{T}_\ell^n = \begin{pmatrix} \boldsymbol{T}_{\ell-1}^n \\ \boldsymbol{T}_{\ell-1}^n + \boldsymbol{D}_{\ell-1}^n \end{pmatrix}$$

with

$$\boldsymbol{D}_{\ell-1}^n = \begin{pmatrix} \mathbf{0}_{\ell-2} \\ \vdots \\ \mathbf{0}_{\ell-n} \\ \boldsymbol{e}_{\ell-n} \end{pmatrix} \in \mathbb{R}^{2^{\ell-1}}.$$

Here, for any positive integer m, we denote by $\mathbf{0}_m$ the zero vector in $\mathbb{R}^{2^m}$ and by $\boldsymbol{e}_m$ the vector in $\mathbb{R}^{2^m}$ all of whose entries have value 1.

We need to know more about the structure of $\boldsymbol{T}_\ell^n$. An easy induction argument yields:

Lemma 2. For any $1 \leq i \leq n$, we have

$$\boldsymbol{T}_{n+i}^n = \begin{pmatrix} \boldsymbol{T}_{n+i-1}^n \\ \boldsymbol{T}_{n+i-2}^n \\ \vdots \\ \boldsymbol{T}_n^n \\ \widetilde{\mathbf{0}}_n^i \\ \boldsymbol{e}_{i-1} \\ 2\boldsymbol{e}_{i-2} \\ \vdots \\ i\boldsymbol{e}_0 \\ i+1 \end{pmatrix},$$

where $\widetilde{\mathbf{0}}_n^i = {}^t(0,\ldots,0) \in \mathbb{R}^{2^n-2^i}$.

From this, one easily deduces:

Lemma 3. For any $0 \leq i \leq n$,

$$\left\|\boldsymbol{T}_{n+i}^n\right\|^2 = 2^i(3i-1)+2.$$

Proposition 2. For any pair of integers $\ell \geq n$,

$$\|\boldsymbol{T}_\ell^n\|^2 = \begin{cases} 2^{\ell-n}(3(\ell-n)-1)+2 & \text{if } n \leq \ell \leq 2n \\ 2^{\ell-n}(3(\ell-n)-1)+2^{\ell-2n}(2+(\ell-2n)(\ell-2n-1)) & \text{if } \ell \geq 2n. \end{cases}$$

Proof. The first case ($n \leq \ell \leq 2n$) is just the previous lemma. In the second case ($\ell \geq 2n$), we again apply induction on ℓ and use that ${}^t\boldsymbol{T}_{n+i}^n \boldsymbol{D}_{n+i}^n = 2^{i+1}-1$ and the fact that the trace of ${}^t\boldsymbol{T}_\ell^n$ is given by $Tr({}^t\boldsymbol{T}_\ell^n) = 2^{\ell-n}(\ell-n+1)$, for any $\ell \geq n$. $\square$

In order to apply Proposition 1 and taking into account the previous result, it only remains to calculate

$$w_0^2 + \sum_{i=0}^{\ell-1} w_{2^i}^2.$$

First of all, let us note that it will be easier to work with

$$\boldsymbol{v}_\ell^n = \boldsymbol{V}_\ell \boldsymbol{T}_\ell^n \text{ and } v_{\ell,j}^n = (\boldsymbol{v}_\ell^n)_j, \text{ for } j = 0, \ldots, 2^\ell - 1.$$

so it is clear that $\boldsymbol{W}_\ell^n = 2^{-\ell/2}\boldsymbol{v}_\ell^n$. Let us first consider the case $\ell = n$. To simplify, let us write $\boldsymbol{v}_\ell = \boldsymbol{v}_\ell^\ell$. We have

$$\boldsymbol{v}_\ell = \boldsymbol{V}_\ell \boldsymbol{T}_\ell^\ell = \boldsymbol{V}_\ell \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} (-1)^{u_\ell(0)} \\ \vdots \\ (-1)^{u_\ell(2^\ell-1)} \end{pmatrix},$$

where $u_\ell(i)$ denotes the number of ones in the binary representation of i. So $(\boldsymbol{v}_\ell)_0 = 1$ and $(\boldsymbol{v}_\ell)_{2^i} = -1$, for all $i = 0, \ldots, \ell-1$. As $||\boldsymbol{T}_\ell^\ell||^2 = 1$, we find that

$$\varepsilon^*(\boldsymbol{T}_\ell^\ell) = 1 - \frac{1+\ell}{2^\ell}.$$

More generally, let us now assume $\ell \geq n+1$. Note that

$$\boldsymbol{T}_\ell^n = \begin{pmatrix} \boldsymbol{T}_{\ell-1}^n \\ \boldsymbol{T}_{\ell-1}^n + \boldsymbol{D}_{\ell-1}^n \end{pmatrix} = \begin{pmatrix} \boldsymbol{T}_{\ell-1}^n \\ \boldsymbol{T}_{\ell-1}^n \end{pmatrix} + \boldsymbol{D}_\ell^{n+1},$$

with

$$\boldsymbol{D}_\ell^{n+1} = \begin{pmatrix} \boldsymbol{0}_{\ell-1} \\ \boldsymbol{D}_{\ell-1}^n \end{pmatrix}.$$

Using this, we obtain

$$\begin{aligned}
\boldsymbol{v}_\ell^n &= \boldsymbol{V}_\ell \left(\begin{pmatrix} \boldsymbol{T}_{\ell-1}^n \\ \boldsymbol{T}_{\ell-1}^n \end{pmatrix} + \boldsymbol{D}_\ell^{n+1} \right) \\
&= \begin{pmatrix} \boldsymbol{V}_{\ell-1} & \boldsymbol{V}_{\ell-1} \\ \boldsymbol{V}_{\ell-1} & -\boldsymbol{V}_{\ell-1} \end{pmatrix} \left(\begin{pmatrix} \boldsymbol{T}_{\ell-1}^n \\ \boldsymbol{T}_{\ell-1}^n \end{pmatrix} + \boldsymbol{D}_\ell^{n+1} \right) \\
&= 2 \begin{pmatrix} \boldsymbol{v}_{\ell-1}^n \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + \boldsymbol{d}_\ell^{n+1}.
\end{aligned}$$

where $\boldsymbol{d}_\ell^n = \boldsymbol{V}_\ell \boldsymbol{D}_\ell^n$.

So,

$$\begin{aligned}
\boldsymbol{v}_\ell^n &= 2 \begin{pmatrix} 2 \begin{pmatrix} \boldsymbol{v}_{\ell-2}^n \\ \boldsymbol{0}_{\ell-2} \end{pmatrix} + \boldsymbol{d}_{\ell-1}^{n+1} \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + \boldsymbol{d}_\ell^{n+1} \\
&= 4 \begin{pmatrix} \boldsymbol{v}_{\ell-2}^n \\ \boldsymbol{0}_{\ell-2} \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + 2 \begin{pmatrix} \boldsymbol{d}_{\ell-1}^{n+1} \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + \boldsymbol{d}_\ell^{n+1} \\
&= 8 \begin{pmatrix} \boldsymbol{v}_{\ell-3}^n \\ \boldsymbol{0}_{\ell-3} \\ \boldsymbol{0}_{\ell-2} \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + 4 \begin{pmatrix} \boldsymbol{d}_{\ell-2}^{n+1} \\ \boldsymbol{0}_{\ell-2} \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + 2 \begin{pmatrix} \boldsymbol{d}_{\ell-1}^{n+1} \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + \boldsymbol{d}_\ell^{n+1} \\
&= \cdots \\
&= 2^{\ell-n} \begin{pmatrix} \boldsymbol{v}_n^n \\ \boldsymbol{0}_n \\ \vdots \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + 2^{\ell-n-1} \begin{pmatrix} \boldsymbol{d}_{n+1}^{n+1} \\ \boldsymbol{0}_{n+1} \\ \vdots \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + \cdots + 2 \begin{pmatrix} \boldsymbol{d}_{\ell-1}^{n+1} \\ \boldsymbol{0}_{\ell-1} \end{pmatrix} + \boldsymbol{d}_\ell^{n+1}.
\end{aligned}$$

We may write the above formula in a more elegant way, by using the so-called *Kronecker* or *tensor product*. For any two matrices $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times m}$ and $\boldsymbol{B} = (b_{ij}) \in \mathbb{R}^{p\times q}$, this is defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = (a_{ij}\boldsymbol{B}_{ij}) = \begin{pmatrix} a_{11}\boldsymbol{B} & \dots & a_{1m}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\boldsymbol{B} & \cdots & a_{nm}\boldsymbol{B} \end{pmatrix} \in \mathbb{R}^{np\times mq}.$$

To calculate $\boldsymbol{d}_i^{n+1}$ for $i = n+1, \ldots, \ell$, first note that

$$\begin{aligned}
\boldsymbol{d}_\ell^n &= \boldsymbol{V}_\ell \boldsymbol{D}_\ell^n = \begin{pmatrix} \boldsymbol{V}_{\ell-1} & \boldsymbol{V}_{\ell-1} \\ \boldsymbol{V}_{\ell-1} & -\boldsymbol{V}_{\ell-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{0}_{\ell-1} \\ \boldsymbol{0}_{\ell-2} \\ \vdots \\ \boldsymbol{0}_{\ell-n+1} \\ \boldsymbol{e}_{\ell-n+1} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{V}_{\ell-1}\boldsymbol{D}_{\ell-1}^{n-1} \\ -\boldsymbol{V}_{\ell-1}\boldsymbol{D}_{\ell-1}^{n-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{d}_{\ell-1}^{n-1} \\ -\boldsymbol{d}_{\ell-1}^{n-1} \end{pmatrix} \\
&= \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \boldsymbol{d}_{\ell-1}^{n-1} = \boldsymbol{v}_1 \otimes \boldsymbol{d}_{\ell-1}^{n-1}.
\end{aligned}$$

Similarly, if we write $\boldsymbol{v}_i = \boldsymbol{v}_1^{\otimes i}$, for any i we have

$$\boldsymbol{d}_\ell^n = \boldsymbol{v}_i \otimes \boldsymbol{d}_{\ell-i}^{n-i}.$$

Taking $i = n-1$, we can write

$$\begin{aligned}
\boldsymbol{d}_\ell^n &= \boldsymbol{v}_{n-1} \otimes \boldsymbol{d}_{\ell-n+1}^1 = \boldsymbol{v}_{n-1} \otimes \boldsymbol{V}_{\ell-n+1} \begin{pmatrix} \boldsymbol{0}_{\ell-n} \\ \boldsymbol{e}_{\ell-n} \end{pmatrix} \\
&= \boldsymbol{v}_{n-1} \otimes \begin{pmatrix} \boldsymbol{V}_{\ell-n}\boldsymbol{e}_{\ell-n} \\ -\boldsymbol{V}_{\ell-n}\boldsymbol{e}_{\ell-n} \end{pmatrix} = \boldsymbol{V}_{n-1} \otimes \boldsymbol{v}_1 \otimes \boldsymbol{V}_{\ell-n}\boldsymbol{e}_{\ell-n} \\
&= \boldsymbol{v}_n \otimes 2^{\ell-n}\boldsymbol{h}_{\ell-n},
\end{aligned}$$

where

$$\boldsymbol{h}_k = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{2^k},$$

for all $k \in \mathbb{N}$. In a similar way,

$$\boldsymbol{d}_i^{n+1} = 2^{i-n}\boldsymbol{v}_n \otimes \boldsymbol{h}_{i-n},$$

for $i = n+1, \ldots, \ell$. We thus obtain

$$
\begin{aligned}
\boldsymbol{v}_\ell^n &= 2^{\ell-n} \begin{pmatrix} \boldsymbol{v}_n \\ \mathbf{0}_n \\ \vdots \\ \mathbf{0}_{\ell-1} \end{pmatrix} + \sum_{i=n+1}^{\ell} 2^{\ell-i} \begin{pmatrix} \boldsymbol{d}_i^{n+1} \\ \mathbf{0}_i \\ \vdots \\ \mathbf{0}_{\ell-1} \end{pmatrix} \\
&= 2^{\ell-n} \left(\begin{pmatrix} \boldsymbol{v}_n \\ \mathbf{0}_n \\ \vdots \\ \mathbf{0}_{\ell-1} \end{pmatrix} + \sum_{i=n+1}^{\ell} \begin{pmatrix} \boldsymbol{v}_n \otimes \boldsymbol{h}_{i-n} \\ \mathbf{0}_i \\ \vdots \\ \mathbf{0}_{\ell-1} \end{pmatrix} \right) \\
&= 2^{\ell-n} \left(\boldsymbol{h}_{\ell-n} \otimes \boldsymbol{v}_n + \sum_{i=n+1}^{\ell} \boldsymbol{h}_{\ell-i} \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{i-n} \right) \\
&= 2^{\ell-n} \sum_{i=0}^{\ell-n} \boldsymbol{h}_1^{\otimes i} \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_1^{\otimes \ell-n-i}.
\end{aligned}
$$

Note that $\boldsymbol{h}_i = \boldsymbol{h}_1^{\otimes i}$ with $\boldsymbol{h}_1^{\otimes 0} = 1$. As we have mentioned before, we are only interested in the value of $v_{\ell,0}^n$ and $v_{\ell,2^i}^n$ for $i = 0, \ldots, \ell-1$. For the first one, it is clear that

$$v_{\ell,0}^n = 2^{\ell-n}(\ell - n + 1)$$

since $(\boldsymbol{h}_j \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{\ell-n-j})_0 = 1$ for all j.

In order to deduce a general formula for the second case, let us first consider two examples, one for the case $n < \ell \leq 2n$ (we take $\ell = 6$ and $n = 4$) and another one for the case $\ell \geq 2n$ (we take $\ell = 6$ and $n = 2$). First, note that

$$\boldsymbol{v}_6^4 = 2^2 \sum_{j=0}^{2} \boldsymbol{h}_j \otimes \boldsymbol{v}_4 \otimes \boldsymbol{h}_{2-j} = -4\{\boldsymbol{h}_0 \otimes \boldsymbol{v}_4 \otimes \boldsymbol{h}_2 + \boldsymbol{h}_1 \otimes \boldsymbol{v}_4 \otimes \boldsymbol{h}_1 + \boldsymbol{h}_2 \otimes \boldsymbol{v}_4 \otimes \boldsymbol{h}_0\}.$$

Since $\boldsymbol{h}_0 \otimes \boldsymbol{v}_4 \otimes \boldsymbol{h}_2 = {}^t(\boldsymbol{a}, -\boldsymbol{a}, -\boldsymbol{a}, \boldsymbol{a})$ with $\boldsymbol{a} = {}^t(\boldsymbol{h}_2, -\boldsymbol{h}_2, -\boldsymbol{h}_2, \boldsymbol{h}_2)$, $\boldsymbol{h}_1 \otimes \boldsymbol{v}_4 \otimes \boldsymbol{h}_1 = {}^t(\boldsymbol{b}, -\boldsymbol{b}, -\boldsymbol{b}, \boldsymbol{b}, \mathbf{0}_5)$ with $\boldsymbol{b} = {}^t(\boldsymbol{h}_1, -\boldsymbol{h}_1, -\boldsymbol{h}_1, \boldsymbol{h}_1)$ and $\boldsymbol{h}_2 \otimes \boldsymbol{v}_4 \otimes \boldsymbol{h}_0 = {}^t(\boldsymbol{v}_4, \mathbf{0}_4, \mathbf{0}_4, \mathbf{0}_4)$, it should be clear that

$$
\begin{aligned}
v_{6,1}^4 = v_{6,32}^4 &= -2^2 \\
v_{6,2}^4 = v_{6,16}^4 &= -2^2 2 \\
v_{6,4}^4 = v_{6,8}^4 &= -2^2 3.
\end{aligned}
$$

We need to distinguish three cases.

1. If $0 \leq i < \ell - n$, the non-zero summands are $\boldsymbol{h}_j \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{\ell-n-j}$ with $j = \ell - n - i, \ldots, \ell - n$.
2. If $\ell - n \leq i < n$, the non-zero summands are $\boldsymbol{h}_j \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{\ell-n-j}$ with $j = 0, \ldots, \ell - n$.

3. If $n \leq i \leq \ell-1$, the non-zero summands are $\boldsymbol{h}_j \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{\ell-n-j}$ with $n+\ell-n-j > i$, so $j = 0, \ldots, \ell - i - 1$.

In the second case, we have, in a similar way,

$$\boldsymbol{v}_6^2 = 2^4 \sum_{j=0}^{4} \boldsymbol{h}_j \otimes \boldsymbol{v}_2 \otimes \boldsymbol{h}_{4-j} = 16\{\boldsymbol{h}_0 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{h}_4 + \boldsymbol{h}_1 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{h}_3 + \boldsymbol{h}_2 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{h}_2 + \boldsymbol{h}_3 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{h}_1 + \boldsymbol{h}_4 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{h}_0\},$$

so,

$$v_{6,1}^2 = v_{6,32}^2 = -2^4$$
$$v_{6,2}^2 = v_{6,4}^2 = v_{6,8}^2 = v_{6,16}^2 = -2^4 2.$$

Again, we can distinguish three cases:

1. If $0 \leq i \leq n-1$, the non-zero summands are $\boldsymbol{h}_j \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{\ell-n-j}$ with $j = \ell - n - i, \ldots, \ell - n$.
2. If $n \leq i \leq \ell - n$, the non-zero summands are $\boldsymbol{h}_j \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{\ell-n-j}$ with $j = \ell - i - n, \ldots, \ell - i - 1$.
3. If $\ell-n \leq i \leq \ell-1$, the non-zero summands are $\boldsymbol{h}_j \otimes \boldsymbol{v}_n \otimes \boldsymbol{h}_{\ell-n-j}$ with $n+\ell-n-j > i$, so $j = 0, \ldots, \ell - i - 1$.

The general case works similarly, so we obtain:

1. if $n < \ell \leq 2n$ and $i = 0, \ldots, \ell - 1$

$$v_{\ell,2^i}^n = \begin{cases} -2^{\ell-n}(i+1) & \text{if } 0 \leq i < \ell - n \\ -2^{\ell-n}(\ell - n + 1) & \text{if } \ell - n \leq i < n \\ -2^{\ell-n}(\ell - i) & \text{if } n \leq i \leq \ell - 1 \end{cases}$$

2. if $\ell \geq 2n$ and $i = 0, \ldots, \ell - 1$

$$v_{\ell,2^i}^n = \begin{cases} -2^{\ell-n}(i+1) & \text{if } 0 \leq i \leq n - 1 \\ -2^{\ell-n} n & \text{if } n \leq i \leq \ell - n \\ -2^{\ell-n}(\ell - i) & \text{if } \ell - n \leq i < \ell \end{cases}$$

In the first case, we have

$$(v_{\ell,0}^n)^2 + \sum_{i=0}^{\ell-1} (v_{\ell,2^i}^n)^2 = 4^{\ell-n}\{(\ell - n + 1)^2 + 2\sum_{i=0}^{\ell-n} i^2 + (2n - \ell)(\ell - n + 1)^2\} =$$
$$= 4^{\ell-n}\{(\ell - n + 1)^2(2n - \ell + 1) + \tfrac{1}{3}(\ell - n)(\ell - n + 1)(2\ell - 2n + 1)\}$$
$$= 4^{\ell-n}\{1 + n(\ell - n - 1)^2 + \tfrac{\ell-n}{3}(4 - (\ell - n)^2)\}.$$

Similarly, in the second case,

$$(v^n_{\ell,0})^2 + \sum_{i=0}^{\ell-1} (v^n_{\ell,2^i})^2 = 4^{\ell-n}\{(\ell-n+1)^2 + 2\sum_{i=0}^{n} i^2 + (\ell-2n)n^2\}$$
$$= 4^{\ell-n}\{(\ell-n+1)^2 + \tfrac{1}{3}n(n+1)(2n+1) + (\ell-2n)n^2\}$$
$$= 4^{\ell-n}\{(\ell-2n)(n^2+\ell+2) + \tfrac{n}{3}(2n^2+7) + 2n^2+1\}.$$

Finally, the combination of all of the previous results with Proposition 1 yields

Proposition 3. For any pair of integers $\ell \geq n$, we have

$$\gamma(T^n_\ell) = \begin{cases} 4^{\ell-n}\left(1 + n(\ell-n+1)^2 + \frac{\ell-n}{3}(4-(\ell-n)^2)\right) & \text{if } n \leq \ell \leq 2n \\ 4^{\ell-n}\left((\ell-2n)(n^2+\ell+2) + \frac{n}{3}(2n^2+7) + 2n^2+1\right) & \text{if } \ell \geq 2n. \end{cases}$$

Proof. Note that

$$\gamma(T^n_\ell) = {}^t\boldsymbol{T}^n_\ell \boldsymbol{G}_\ell \boldsymbol{T}^n_\ell = 2^\ell\, {}^t\boldsymbol{W}^n_\ell\, {}^t\boldsymbol{W}_\ell \boldsymbol{E}_\ell \boldsymbol{W}_\ell \boldsymbol{w}^n_\ell = {}^t\boldsymbol{v}^n_\ell \boldsymbol{D}_\ell \boldsymbol{v}^n_\ell = (v^n_{\ell,0})^2 + \sum_{i=0}^{\ell-1}(v^n_{\ell,2^i})^2.$$

□

Theorem 1. The epistasis of the Template function T^n_ℓ is:

$$\varepsilon^*(T^n_\ell) = \begin{cases} 1 - \dfrac{1 + n(\ell-n+1)^2 + \frac{(\ell-n)}{3}(4-(\ell-n)^2)}{2^n\left(3(\ell-n) - 1 + 2^{n-\ell+1}\right)} & \text{if } n \leq \ell \leq 2n \\[2ex] 1 - \dfrac{(\ell-2n)(n^2+\ell+2) + \frac{n}{3}(2n^2+7) + 2n^2+1}{2^n\left(3(\ell-n)-1\right) + (\ell-2n)^2 + 2(n+1) - \ell} & \text{if } \ell \geq 2n. \end{cases}$$

In the case $n = \ell$, we recover for large values of ℓ, the *high* epistatic value

$$\varepsilon^*(T^\ell_\ell) = 1 - \frac{1+\ell}{2^\ell}.$$

because T^ℓ_ℓ is just the Dirac function at 1^ℓ. On the other hand, for general values of ℓ, the minimal epistatic value is easily seen to be reached by the case $n = 1$,

$$\varepsilon^*(T^1_\ell) = 1 - \frac{(\ell-2)(\ell+3)+6}{2(3(\ell-1)-1) + (\ell-2)^2 + 4 - \ell} = 0.$$

Indeed, in this case, T^1_ℓ just counts the number of $1's$ in a string and $T^1_\ell = \sum_{i=0}^{\ell-1} g_i$, where $g_i(s) = \delta_{1,s_i}$ is the Kronecker function with value 1 when $s_i = 1$ and zero elsewhere.

4 SOME EXPERIMENTAL RESULTS

In this final section, we show, with some explicit runs, that, for Template functions, the epistasis is a nice indicator of GA-hardness. As a measure of this GA-hardness, we use the average N of the number of generations needed to reach a population, half of whose members are equal to the maximum $1^\ell = 1 \ldots 1$.

We used an SGA (simple genetic algorithm [1]) over a population of size 100, with binary tournament as selection operator, single point crossover with probability 0.7 and mutation with probability 0.02.

In the first case, we consider strings with length $\ell = 16$ and we calculate both epistasis and GA-hardness for all template functions T_ℓ^n, with $2 \leq n \leq 15$. The results are given in Table 1. As expected, the epistasis strongly correlates with GA-hardness.

Note: We did not include the values $n = 1$ and $n = 16$. Actually, for $n = 1$, the problem is linear (one just counts the number of 1's in a string). We found $N = 16.64$, which is slightly higher than the corresponding value for T_{16}^2. This is essentially due to the non-disruptive character of single point crossover in this case.

On the other hand, for $n = 16$, the value of N is unstable (and high), as T_{16}^{16} is just the Dirac function at $1^{(16)}$ and the optimum has to be found through random search, since the associated fitness landscape does not contain any information allowing for directed search.

$\ell = 16$		
n	$\varepsilon^*(T_{16}^n)$	N
2	0.05034	16.28
3	0.20707	16.70
4	0.42233	17.73
5	0.61742	18.51
6	0.76096	20.03
7	0.85654	22.44
8	0.91698	26.47
9	0.95394	32.33
10	0.97552	43.35
11	0.98743	69.49
12	0.99375	149.12
13	0.99698	417.33
14	0.99859	1755.69
15	0.99937	11197.98

Table 1

We also fixed the size of the template (n) and calculated epistasis for different values of $\ell \geq n$. There again appears a clear link between the (decreasing) values of epistasis and GA-hardness. In Table 2 we show the results for $n = 3$ and $3 \leq \ell \leq 7$.

$n = 3$		
ℓ	Epistasis	Number of runs
3	0.5	53.5
4	0.41667	32.6
5	0.36364	26.8
6	0.33333	13.3
7	0.31111	9.7

Table 2

5 CONCLUSION

In this note, we introduced some general machinery relating the epistasis of a fitness function to its Walsh coefficients. We then applied this to explicitly calculate the epistasis of Template functions. Experimental evidence indicates that the GA-hardness of individual Template functions is strongly related to their epistasis.

6 APPENDIX

The main purpose of this section is to sketch the alternative (tedious) calculation of the normalized epistasis of the Template functions given in [7]. As we have already calculated $||\boldsymbol{T}_\ell^n||$, let us reconsider $\gamma(T_\ell^n) = {}^t\boldsymbol{T}_\ell^n \boldsymbol{G}_\ell \boldsymbol{T}_\ell^n$. Note that, for all $\ell \geq n$,

$$\begin{aligned}
\gamma(T_\ell^n) &= {}^t\boldsymbol{T}_\ell^n \boldsymbol{G}_\ell \boldsymbol{T}_\ell^n \\
&= \left({}^t\boldsymbol{T}_{\ell-1}^n, {}^t\boldsymbol{T}_{\ell-1}^n + {}^t\boldsymbol{D}_{\ell-1}^n\right) \begin{pmatrix} \boldsymbol{G}_{\ell-1} + \mathbf{U}_{\ell-1} & \boldsymbol{G}_{\ell-1} - \mathbf{U}_{\ell-1} \\ \boldsymbol{G}_{\ell-1} - \mathbf{U}_{\ell-1} & \boldsymbol{G}_{\ell-1} + \mathbf{U}_{\ell-1} \end{pmatrix} \\
&\quad \begin{pmatrix} \boldsymbol{T}_{\ell-1}^n \\ \boldsymbol{T}_{\ell-1}^n + \boldsymbol{D}_{\ell-1}^n \end{pmatrix} \\
&= 4{}^t\boldsymbol{T}_{\ell-1}^n \boldsymbol{G}_{\ell-1} \boldsymbol{T}_{\ell-1}^n + 4{}^t\boldsymbol{T}_{\ell-1}^n \boldsymbol{G}_{\ell-1} \boldsymbol{D}_{\ell-1}^n + {}^t\boldsymbol{D}_{\ell-1}^n \boldsymbol{G}_{\ell-1} \boldsymbol{D}_{\ell-1}^n \\
&\quad + {}^t\boldsymbol{D}_{\ell-1}^n \mathbf{U}_{\ell-1} \boldsymbol{D}_{\ell-1}^n,
\end{aligned}$$

where the last terms are:

$${}^t\boldsymbol{D}_{\ell-1}^n \mathbf{U}_{\ell-1} \boldsymbol{D}_{\ell-1}^n = 2^{\ell-n} \|\boldsymbol{e}_{\ell-n}\|^2 = 4^{\ell-n}$$

and

$${}^t\boldsymbol{D}_{\ell-1}^n \boldsymbol{G}_{\ell-1} \boldsymbol{D}_{\ell-1}^n = \sum_{i,j=0}^{2^{\ell-n}-1} g_{ij} + (n-1) \cdot 4^{\ell-n} = n \cdot 4^{\ell-n}.$$

In order to determine ${}^t\boldsymbol{T}_{\ell-1}^n \boldsymbol{G}_{\ell-1} \boldsymbol{D}_{\ell-1}^n$, let us denote by

$$\beta_k = ({}^t\mathbf{0}_{\ell-k-1}, \ldots, {}^t\mathbf{0}_{\ell-n}, {}^t\boldsymbol{e}_{\ell-n}),$$

for any $1 \leq k \leq n$. Using a recursive argument, it can be proved that

$$\beta_k = 2\beta_{k+1} + (n-k+1)2^k 4^{\ell-n-k}.$$

Using this and the symmetry of $\boldsymbol{G}_\ell$, we obtain:

$${}^t\boldsymbol{T}^n_{\ell-1}\boldsymbol{G}_{\ell-1}\boldsymbol{D}^n_{\ell-1} = \beta_1 = 2\beta_2 + 4^{\ell-n-1}2n = \cdots = 2^i\beta_{i+1} + 4^{\ell-n-1}i(2n+1-i).$$

In particular, if $i = n-2$, then

$$\begin{aligned} {}^t\boldsymbol{T}^n_{\ell-1}\boldsymbol{G}_{\ell-1}\boldsymbol{D}^n_{\ell-1} &= 2^{n-2}\beta_{n-1} + 4^{\ell-n-1}(n-2)(n+3) \\ &= 2^{n-2}\left({}^t\boldsymbol{0}_{\ell-n}, {}^t\boldsymbol{e}_{\ell-n}\right)\boldsymbol{G}_{\ell-n+1}\boldsymbol{T}^n_{\ell-n+1} + 4^{\ell-n-1}(n-2)(n+3) \\ &= 2^{n-2}\left(2{}^t\boldsymbol{e}_{\ell-n}\boldsymbol{G}_{\ell-n}\boldsymbol{T}^n_{\ell-n} + {}^t\boldsymbol{e}_{\ell-n}\boldsymbol{G}_{\ell-n}\boldsymbol{D}^n_{\ell-n} \right. \\ &\quad \left. + {}^t\boldsymbol{e}_{\ell-n}\mathrm{U}_{\ell-n}\boldsymbol{D}^n_{\ell-n}\right) + 4^{\ell-n-1}(n-2)(n+3). \end{aligned}$$

As ${}^t\boldsymbol{e}_{\ell-n}\boldsymbol{G}_{\ell-n} = {}^t\boldsymbol{e}_{\ell-n}\mathrm{U}_{\ell-n} = 2^{\ell-n}\,{}^t\boldsymbol{e}_{\ell-n}$ and ${}^t\boldsymbol{T}^n_{n+i}\boldsymbol{D}^n_{n+i} = 2^{i+1}-1$, we have

$$\begin{aligned} {}^t\boldsymbol{T}^n_{\ell-1}\boldsymbol{G}_{\ell-1}\boldsymbol{D}^n_{\ell-1} &= 2^{\ell-1}\left(Tr(\boldsymbol{T}^n_{\ell-n}) + Tr(\boldsymbol{D}^n_{\ell-n})\right) + 4^{\ell-n-1}(n-2)(n+3) \\ &= 2^{\ell-1}\left(2^{\ell-2n}(\ell-2n+1) + 2^{\ell-2n+1}\right) + 4^{\ell-n-1}(n-2)(n+3) \\ &= 4^{\ell-n-1}(n^2-3n+2\ell). \end{aligned}$$

Combining all of these facts with another recursive argument, we find for $\ell \geq 2n$,

$$\begin{aligned} \gamma(T^n_\ell) = {}^t\boldsymbol{T}^n_\ell\boldsymbol{G}_\ell\boldsymbol{T}^n_\ell &= 4{}^t\boldsymbol{T}^n_{\ell-1}\boldsymbol{G}_{\ell-1}\boldsymbol{T}^n_{\ell-1} + 4^{\ell-n}(n^2-3n+2\ell) + n4^{\ell-n} + 4^{\ell-n} \\ &= 4^{\ell-n}\left((\ell-2n)(n^2+\ell+2) + \tfrac{n}{3}(2n^2+7) + 2n^2+1\right), \end{aligned}$$

which coincides with the results in 3.

One proves in a similar way that

$$\gamma(T^n_{n+i}) = 4^i\left(n(i+1)^2 + 1 + \frac{i}{3}(4-i^2)\right),$$

when $0 \leq i \leq n$, which again coincides with the corresponding case in 3.

REFERENCES

[1] Bäck, T.—Hoffmeister, F.: Extended Selection Mechanisms in Genetic Algorithms. In: Proceedings of the Fourth International Conference on Genetic Algorithms, ed. R. K. Belew and L. B. Booker, Morgan Kaufmann Publishers, San Mateo, 1991.

[2] Corcoran, A. L.—Wainwright, R. L.: LibGA: A User-Friendly Workbench for Order-Based Genetic Algorithm Research. Proceedings of the 1993 ACM/SIGAPP Symposium on Applied Computing, 1993.

[3] Davidor, Y.: Epistasis Variance: A Viewpoint on GA-hardness. In: Foundations of Genetic Algorithms 1, ed. G. J. E. Rawlins, Morgan Kaufmann Publishers, San Mateo, 1991.

[4] Forrest, S.—Mitchell, M.: Relative Building-Block Fitness and the Building-Block Hypothesis. In: Foundations of Genetic Algorithms 2, ed. L. D. Withley, Morgan Kaufmann Publishers, San Mateo, 1993.

[5] Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[6] Goldberg, D. E.: Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction. Complex Systems 3, pp. 129–152, 1989.

[7] Iglesias, M. T.—Verschoren, A.—Vidal, C.: Template Functions and Their Epistasis. Proceedings of the International Conference On Modelling and Simulation MS-2000, Las Palmas de Gran Canaria, pp. 539–546, 2000.

[8] Naudts, B.—Suys, D.—Verschoren, A.: Epistasis, Deceptivity and Walsh Transforms. Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS '98) Volume 1: Fuzzy Logic/Genetic Algorithms, pp. 210–216, ICSC Academic Press, 1998.

[9] Naudts, B.—Suys, D.—Verschoren, A.: Generalized Royal Road Functions. Bull. Soc. Math. Belg., Simon Stevin, 6, pp. 147–154, 1999.

[10] Rawlins, G. J. E.: Foundations of Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, 1991.

[11] Stickberger, M. M.: Genetics. Collier-McMillan Ltd., London, 1968.

[12] Suys, D.—Verschoren, A.: Extreme Epistasis. International Conference on Intelligent Technologies in Human-Related Sciences (ITHURS '96) Vol. II, pp. 251–258, León, 1996.

[13] Van Hove, H.: Representational Issues in Genetic Algorithms. Ph. D. thesis, University of Antwerp 1995.

[14] Van Hove, H.—Verschoren, A.: On Epistasis. Computers and AI, Vol. 14, pp. 271–277, 1994.

[15] Van Hove, H.—Verschoren, A.: What Is Epistasis? Bull. Soc. Math. Belg., 5, pp. 69–77, 1998.

María Teresa Iglesias is full professor at the Universidade da Coruña. After she finished her studies in mathematics, she started research in differential geometry with a grant of the Spanish Ministry of Education and Science. Currently she works in artificial intelligence. She is the co-author of several papers in differential geometry and computer science and also a book on generic optimization.

Concepción Vidal is full professor at the Department of Computer Science of the University of A Coruña. She received the Ph. D. degree in mathematics from the University of Santiago de Compostela in 1993 with a work on ring theory and noncommutative algebraic geometry. After receiving her habilitation at the Faculty of Computer Science of the University of A Coruña, she started her research in artificial intelligence. She is the author or co-author of more than 10 papers and books on pure mathematics and computer science.

Alain Verschoren was born in 1954 and is full professor at the University of Antwerp. He is the author of some 200 papers and 40 books or conference proceedings, mainly on ring theory, noncommutative algebraic geometry and artificial intelligence. He was the advisor of more than 20 Ph. D.'s in these topics and gave numerous talks at universities and conferences.