Walsh Functions and Predicting Problem Complexity

Robert B. Heckendorn
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523 USA
heckendo@cs.colostate.edu

Abstract

Theorems are given establishing the epistatic bounds for problems that can be stated as mathematical expressions. Examples of the application of the theorems and techniques for controlling epistasis are presented.

1 Introduction

Many problems can be cast as the optimization of a mathematical model. In the preparation of the model for optimization by a genetic algorithm, the parameters are encoded into a bit string for processing by the algorithm resulting in a function whose domain is in bitspace and whose range is in the reals. More formally:

$$f_{ga} = f_{model}(f_{decode}) : \mathcal{B}^L \rightarrow \mathcal{R}$$

where $\mathcal{B}^L$ is an L dimensional bitspace.

Many features of f_{ga} tend to make it more difficult for a genetic algorithm to solve. Epistasis, linkage, hyperplane organization (deception) and interpartition conflict are some of these features. This paper presents several new measures of epistasis. Theorems are given establishing the bounds of epistasis for problems that can be stated as mathematical expressions[1]. How this insight might be used to reduce problem epistasis is discussed and empirical evidence to demonstrate the application of the theorems is presented.

2 Walsh Sums and Function Order

In order to measure the degree of interaction between bits in a function, $f : \mathcal{B}^L \rightarrow \mathcal{R}$ (where $\mathcal{B}^L$ is the L bit binary space), it is helpful to break the function down into a linear combination of the interactions. For an L bit function there are 2^L possible interactions. A Walsh polynomial [1, 3] is a classic and useful linear decomposition. Any function $f : \mathcal{B}^L \rightarrow \mathcal{R}$ can be broken down into the Walsh polynomial as follows:

$$f(x) = \sum_{i=0}^{2^L-1} w_i \psi_i(x)$$

where $\psi_i(x) : \mathcal{B}^L \times \mathcal{B}^L \mapsto \{1, -1\}$ is the i^{th} **Walsh function** of x and $w_i : \mathcal{B}^L \mapsto \mathcal{R}$ is the i^{th} **Walsh coefficient**. A Walsh function does a bit by bit parity check between two bit strings x and i. If they share an even number of `1` bits in the same position the function returns a 1 otherwise it returns a -1.

In order to apply Walsh functions to the problem of epistasis we need to group the Walsh coefficients into a more manageable measure. Let a **Walsh sum** be denoted by W_b where:

$$W_b = \sum_{i:bc(i)=b} |w_i|$$

and $bc(i)$ is the bit count of i. That is W_b is the sum of the **absolute value** of all of the Walsh coefficients for bit patterns i with exactly b bits set to 1. For example:

$$\begin{aligned}
W_0 &= |w_0| \\
W_1 &= |w_1| + |w_2| + |w_4| + |w_8| + \ldots + |w_{2^{(L-1)}}| \\
W_2 &= |w_3| + |w_5| + |w_6| + |w_9| + \ldots + |w_{3*2^{(L-2)}}| \\
&\vdots \\
W_L &= |w_{2^L-1}|
\end{aligned}$$

Notice that for an L bit function there are $L+1$ Walsh sums and that W_k is the sum of $\binom{L}{k}$ Walsh co-

[1] The proofs are available from the author.

efficients. Since the absolute value of the Walsh coefficients is used, any nonzero Walsh coefficient will force the corresponding Walsh sum to be nonzero. Therefore the n^{th} Walsh sum could be an effective measure of the magnitude of n-bit interactions.

Let the **order of a function**, denoted $\Omega(f)$, be defined as the largest i such that $W_i \neq 0$. In the special case where all W_i are 0, that is $f(x) = 0$, $\Omega(f) = 0$. Intuitively, the order of a function is the size of the largest set of interdependent bits. So $\Omega(f)$ is a measure of the maximum level of epistasis of f and W_i measures the magnitude of the i-bit interdependence.

Another useful linear decomposition of f is by Spectral functions. f can be decomposed into the sum of at most $L+1$ spectral functions S_i as

$$f(x) = \sum_{j=0}^{L} S_i(x) \quad \text{where} \quad S_b(x) = \sum_{i:bc(i)=b} w_i \psi_i(x)$$

Notice that the only possible nonzero Walsh sum for S_i is W_i and that $S_0 = w_0$. Spectral functions will enable us to more easily express ideas about functions with limited degrees of interaction.

3 Function Order and Models

If we know the mathematical expression for function f do we know anything about $\Omega(f)$? The answer is yes as seen in the next set of theorems:

Theorem 1 (Polynomial Complexity Theorem)

Let P_n be a polynomial with degree $n : n \geq 0$ such that $P_n : \mathcal{B}^L \rightarrow \mathcal{R}$ then

$$\Omega(P_n) \leq n$$

Extraction is often the basis of composing parameters for functions with domain $\mathcal{R}^n$ from strings in $\mathcal{B}^L$ that are processed by genetic algorithms. Let $x[n_1, n_2] : \mathcal{B}^L \times Integer \times Integer \rightarrow \mathcal{B}^L$ be the **extraction operator** which extracts bits in positions n_1 through n_2 ($n_2 \geq n_1$) from string x and placing them in the least significant bit portion of the string with zero fill. This operation can be performed by masking and shifting. The Extraction Theorem shows that Ω for a function is limited by the number of bits extracted for its parameter.

Theorem 2 (Extraction Theorem)

$$\Omega(f(x[n_1, n_2])) \leq \min(\Omega(f(x)), n_2 - n_1 + 1)$$

The next theorem will allow us to unite the previous two theorems and apply them to f_{ga}.

Theorem 3 (Polynomial Composition Theorem)

Let $P_n(x_0, x_1, ...x_{n-1})$ be a polynomial in $x_0, x_1, ...x_{n-1}$ that takes $\mathcal{R}^n \rightarrow \mathcal{R}$ such that

$$P_n(x_0, x_1, ...x_{n-1}) = \sum_{all\ terms} a_{k_0 k_1 ... k_{n-1}} x_0^{k_0} x_1^{k_1} ... x_{n-1}^{k_{n-1}}$$

with $k_i \geq 0$ and let $f_0, f_1, ...f_{n-1}$ be functions such that $f_i : \mathcal{B}^L \rightarrow \mathcal{R}$ then

$$\Omega(P_n(f_0(x), f_1(x), ...f_{n-1}(x)) \leq \max_{all\ terms} k_0 Omega(f_0(x)) + ... + k_{n-1}\Omega(f_{n-1}(x))$$

4 Predicting Complexity

By complexity I mean the epistatic component of problem difficulty. Although there are certainly easy problems with high epistasis there is a general correlation between epistasis and difficulty of solving a problem with a simple genetic algorithm. Empirical studies [2] show that functions with higher Ω tend to have higher levels of deception and, under a simple genetic algorithm, are less frequently solved to optimality and converge to less optimal answers. So even though high epistasis is not the complete measure of problem difficulty, it provides a mechanism for the creation of more difficult problems.

Barring recombination operators that understand about the structure of the decoding function, the genetic algorithm only sees f_{ga}. Therefore, the function that is really being solved is f_{ga} and it is the difficulty of this *composite* function that estimates the difficulty of solving the problem with a genetic algorithm.

The decoding function can be represented as a vector of functions $\vec{f} = (f_0, f_1, ...f_{n-1})$ where $f_i : \mathcal{B}^L \rightarrow \mathcal{R}$ and maps the bitstring to a real argument for the model function. Therefore, for a chromosome x in $\mathcal{B}^L$:

$$f_{model}(f_{decode}(x)) = P_n(f_0(x), f_1(x), ...f_{n-1}(x))$$

for model functions that are polynomials of n variables. In these cases the Polynomial Composition Theorem applies. This means given just the mathematical models and extraction functions we can derive an upper bound for the degree of bit interaction in f_{ga}. We may even be able to apply this understanding to control the level of complexity and thereby improve performance.

In the next section we will give some examples of the predictive power of the theorems.

5 Test Driving the Theorems

In the following examples we present tables of Walsh sums for W_0 through W_8. The first column is the order of the Walsh sum. The remaining columns are the Walsh sums for that order for various functions. All functions presented in this section are evaluated using a string length of 8.

Table 1 shows that the n^{th} power of a linear function has an Ω of n. The second column is the Walsh sum for $f(x) = x$. The third column is $f(x) = x^5$. For both functions $x \in [0, 2^L - 1]$. Notice how $\Omega(f)$ is limited to the maximum power of the polynomial by the Polynomial Complexity Theorem.

Order	x	x^5
0	127.5	1.811e+11
1	127.5	4.299e+11
2	0	3.457e+11
3	0	1.088e+11
4	0	1.229e+10
5	0	3.731e+08
6	0	0
7	0	0
8	0	0

Table 1: Walsh Sums for $f(x) = x$ and x^5

A very useful invariant of functions is found in Odd and Even orderness. The product of any two spectral functions S_i, S_j results in a function that is the sum of a series of all odd ordered spectral functions or all even ordered spectral functions up to and including order $i + j$. Such functions are called **odd ordered** or **even ordered** functions. The product of an odd ordered function with an even ordered function is an odd ordered function. The product of two odd or two even ordered functions is an even ordered function.

Since a linear function has an Ω of 1, it must be the sum of the two spectral functions S_0 and S_1. But S_0 is just the average of all of the function values. By adjusting the average of the linear function to zero we can get $S_0 = 0$ leaving the function equal to some S'_1.

In table 2 we have mapped the range $[0, 2^L - 1]$ to the range $[-(2^{(L-1)} - .5), (2^{(L-1)} - .5)]$. This technique is called **argument centering**. For the linear function x over this argument range, the average of x is now 0. Therefore, the linear function is now equal to a single nonzero spectral function S_1 and is an odd ordered function. Since odd powers of S_1 have only odd ordered Walsh sums, x^5 becomes the odd ordered function we see in the second column of table 2.

Order	x^5 Without Centering	x^5 With Centering
0	1.811e+11	0
1	4.299e+11	1.923e+10
2	3.457e+11	0
3	1.088e+11	1.409e+10
4	1.229e+10	0
5	3.731e+08	3.731e+08
6	0	0
7	0	0
8	0	0

Table 2: Walsh Sums for x^5

Keeping a f_{model} restricted to a polynomial may seem limiting but actually it is quite powerful since all continuously differentiable functions can be expressed as a Taylor series expansion. For example, the Taylor series expansion about 0 for cos is:

$$\cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + \ldots$$

The Walsh sums for $\cos(x)$ for $x \in [0, \pi/2]$ are presented in table 3. By centering the argument x so that $x \in [-\pi/2, \pi/2]$ the powers of x in cosine, which are all even, force the function to become even ordered in table 3. One may not always be in position to change the range of arguments to a function but there is often more flexibility than may seem at first.

Let's exercise the theorems on a more complex model. Suppose:

$$f_{model}(x, y, z) = x^2 + \cos(y)\sin(z)$$

Order	Without Centering	With Centering
0	0.6386	0.6366
1	0.512	0
2	0.1363	0.6459
3	0.01511	0
4	0.0007757	0.01548
5	1.837e-05	0
6	2.019e-07	1.634e-05
7	9.314e-10	0
8	1.444e-12	4.692e-10

Table 3: Walsh Sums for $\cos(x)$

and that the decoding functions for the chromosome string s are:

$$x = s[1,2], \quad y = s[3,5], \quad z = s[6,7]$$

where the extraction function centers the argument and scales the value to the range $[-\pi, \pi]$. Our analysis of $\Omega(f_{model})$ can proceed as follows:

1. f_{model} has 2 terms so Ω is the max of the Ω's of the individual terms.
2. $\Omega(x^2) = 2$
3. $\Omega(\cos(y))$ is limited by y having a width of 3 bits. Since y is centered and cos is an even ordered function under centered arguments only W_0 and W_2 will be nonzero. Therefore $\Omega(\cos(y)) = 2$. Note that if y included just one more bit $\Omega(\cos(y)) = 4$.
4. $\Omega(\sin(z))$ is limited by z having a width of 2 bits. A Similar argument to above shows only, W_1 will be nonzero. Therefore $\Omega(\sin(z)) = 1$. Again if z included just one more bit $\Omega(\sin(z)) = 3$.
5. So the computation of $\Omega(f_{ga})$ is as follows:

$$\begin{aligned}\Omega(f_{ga}) &\leq \max(\Omega(x^2), \Omega(cos(y)) + \Omega(sin(z)) \\ &= \max(2, 2+1) \\ &= 3\end{aligned}$$

If the arguments to cosine and sine each had just one more bit then $\Omega(f_{ga}) = 7$. This demonstrates that the choice of encoding length can be important and could be engineered to reduce interactions. Table 4 shows that our estimate is correct:

Order	Walsh Sum
0	5.483
1	0.1083
2	4.386
3	0.8885
4	3.331e-16
5	2.22e-16
6	1.11e-16
7	0
8	0

Table 4: $x[1,2]^2 + \cos(x[3,5])\sin(x[6,7])$

6 Conclusions

In this paper I have shown that in the case where a model function can be expressed as a polynomial and each parameter encoding can be expressed as a polynomial applied to an extraction of bits from a bit string, the degree and magnitude of bit interactions can be predicted. I have shown that this information could be used to design model and encoding functions with lower epistasis. I have shown that by use of Taylor series the meaning of polynomial can be expanded to some common infinitely differentiable functions. Finally, I developed some techniques such as argument centering and odd/even series truncation that could be used to reduce bit interactions.

References

[1] David E. Goldberg, "Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction", *Complex Systems, No. 3, 1989, pp129-152*

[2] R. B. Heckendorn, D. Whitley, S. Rana "Non-linearity, Hyperplane Ranking and the Simple Genetic Algorithms", *Foundations of Genetic Algorithms 4, 1997, Morgan Kaufmann Publishers*

[3] Colin Reeves and Christine Wright, "An Experimental Design Perspective on Genetic Algorithms", *Foundations of Genetic Algorithms 3, 1995, Morgan Kaufmann Publishers*