Walsh and Haar Functions in Genetic Algorithms

Sami Khuri
Department of Mathematics and Computer Science
San José State University
One Wahington Square
San José, CA 95192-0103
khuri@sjsumcs.sjsu.edu.
Fax: (408)924-5080.

In the Proceedings of the 1994 ACM Symposium on Applied Computing
Phoenix, Arizona, March 6–8, 1994; pages 201-205.

Abstract

Theoretical analysis of fitness functions in genetic algorithms has included the use of Walsh functions [14]. They form a convenient basis for the expansion of fitness functions [3]. These orthogonal, rectangular functions have also been used to compute the average fitness values of schemata [5]. This work explores the use of Haar functions [7] for the same purposes. While 2^ℓ non-zero terms are required for the expansion of a given function as a linear combination of Walsh functions, at most $\ell + 1$ non-zero terms are required with the Haar expansion, where ℓ is the size of each binary string in the solution space. Similarly, Haar coefficients require less computation than their Walsh counterparts. The total number of terms required for the expansion of the fitness function at a given point using Haar is of order 2^ℓ, substantially less than Walsh's $2^{2\ell}$. A comparison of Haar functions and Walsh functions with respect to fitness averages shows that the use of Haar functions will reduce computation time. Furthermore, the advantage of Haar over Walsh functions remains large (of order $\ell 2^\ell$) when fast transforms are used.

Introduction

Traditionally, Fourier series and transforms have been used to represent large classes of functions by superpositioning sine and cosine functions. More recently, other classes of complete, orthogonal functions are being used for the same purpose [8]. These new functions are rectangular and are easier to define and use with digital logic. More specifically, Walsh functions [14] have been used to calculate the average fitness value of a schema, to decide whether a certain function is hard or easy for a genetic algorithm, and to design deceptive functions for the genetic algorithm [3, 4, 5]. As a basis, these rectangular functions, which take values ± 1, are more practical than the traditional trigonometric basis [3].

In the next section, Walsh functions as a basis for fitness functions of genetic algorithms are briefly reviewed. In the third section, the representation of fitness functions as linear combinations of Haar functions is investigated. Then Walsh and Haar functions as bases for fitness functions are compared. Haar functions can be processed more efficiently than Walsh functions: if ℓ denotes the size of each binary string in the solution space, at most 2^ℓ non-zero terms are required for the expansion of a given function as a linear combination of Walsh functions, while at most $\ell + 1$ non-zero terms are required with Haar expansion. Similarly, Haar coefficients require less computation than their Walsh counterparts. The total number of terms required for the computation of the expansion of the fitness function at a given point using Haar is of order 2^ℓ, which is for large ℓ substantially less than Walsh's $2^{2\ell}$ and the advantage of Haar over Walsh functions is of order $\ell 2^\ell$ when fast transforms are used.

Walsh Functions

Walsh functions form an ordered set of rectangular waveforms taking one of two amplitude values: $+1$ and -1. They form a complete orthogonal set of functions and can thus be used as a basis. Consequently, any fitness function defined on binary strings of length ℓ can be represented as a linear combination of discrete Walsh functions. They were first introduced by Bethke [9, 3] and later presented by Goldberg [4]. It is more convenient to express a binary string $x = x_\ell x_{\ell-1} \ldots x_2 x_1$ as a string $y = y_\ell y_{\ell-1} \ldots y_2 y_1$ where $y_i \in \{+1, -1\}$ for all $i = 1, \ldots, \ell$. This is achieved by defining an auxilliary function,$auxw$, which maps binary strings of length ℓ into strings whose components are ± 1. More precisely,

$$\begin{aligned} & auxw(x_\ell x_{\ell-1} \ldots x_2 x_1) \\ = \quad & y_\ell y_{\ell-1} \ldots y_2 y_1, \text{ where } y_i = 1 - 2x_i \text{ for } i = 1, 2, \ldots, \ell. \end{aligned}$$

We note that a string of binary digits $s_\ell s_{\ell-1}\ldots s_1$ can be thought of as being the binary representation of a decimal number s where $s=\Sigma_{i=1}^{\ell} s_i 2^{i-1}$.

Thus a Walsh function (or monomial [4]) of index j over a binary string x of length ℓ is given by $\Psi_j(x)=\Pi_{i=1}^{\ell} y_i^{j_i}$ where $x_\ell x_{\ell-1}\ldots x_1$ and $j_\ell j_{\ell-1}\ldots j_1$ are the binary representations of x and j, and $y_i=1-2x_i$. The set $\{\Psi_j(x) : j=0,1,2,\ldots,2^\ell-1\}$ forms a basis for the fitness functions defined on $[0,2^\ell)$. That is:

$$f(x)=\Sigma_{j=0}^{2^\ell-1} w_j \Psi_j(x), \tag{1}$$

where the w_j 's are the Walsh coefficients given by:

$$w_j=\frac{1}{2^\ell}\Sigma_{x=0}^{2^\ell-1} f(x)\Psi_j(x). \tag{2}$$

$\Psi_j(x)$ is defined for discrete values of $x\in[0,2^\ell)$. Since $\Psi_j(x)\neq 0$ for each $j\in[0,2^\ell)$, every Walsh coefficient w_j depends on the function value over every integer in $[0,2^\ell)$. For the same reason, unless $f(x)$ is orthogonal to a Walsh function $\Psi_j(x)$, the expansion of $f(x)$ as a linear combination of Walsh functions has 2^ℓ non-zero terms. We now explore the Haar functions and compare them with the Walsh functions.

Haar Functions

The set of Haar functions also forms a complete set of orthogonal rectangular basis functions. These functions, as proposed by the Hungarian mathematician Haar [7], take values of 1, 0 and -1, multipled by powers of $\sqrt{2}$. The interval they are defined on is usually normalized to $[0,1)$. Kremer's definition of rank-ordered Haar functions, $H_j(t)$, for instance, gives an orthonormal basis [11]. Since $t\in[0,1)$, the integers x in $[0,2^l)$ have to be mapped "in the usual way (i.e., $t=2^{-l}x$)" to the unit interval $[0,1)$ before applying them as a basis for fitness functions.

In the following variation, the Haar functions are defined on $[0,2^\ell)$, but they are unnormalized, taking values of 0 and ± 1.

$$\begin{aligned}
H_0(x) &= 1 \quad \text{for } 0\le x<2^\ell\\
H_1(x) &= \begin{cases} 1 & \text{for } 0\le x<2^{\ell-1}\\ -1 & \text{for } 2^{\ell-1}\le x<2^\ell\end{cases}\\
H_2(x) &= \begin{cases} 1 & \text{for } 0\le x<2^{\ell-2}\\ -1 & \text{for } 2^{\ell-2}\le x<2^{\ell-1}\\ 0 & \text{elsewhere in } [0,2^\ell)\end{cases}\\
&\vdots
\end{aligned}$$

$$H_{2^q+m}(x) = \begin{cases} 1 & \text{for } (2m)2^{\ell-q-1} \leq x < (2m+1)2^{\ell-q-1} \\ -1 & \text{for } (2m+1)2^{\ell-q-1} \leq x < (2m+2)2^{\ell-q-1} \\ 0 & \text{elsewhere in } [0, 2^\ell) \end{cases} \quad (3)$$

$$\vdots$$

$$H_{2^\ell-1}(x) = \begin{cases} 1 & \text{for } 2(2^{\ell-1}-1) \leq x < 2^\ell - 1 \\ -1 & \text{for } 2^\ell - 1 \leq x < 2^\ell \\ 0 & \text{elsewhere in } [0, 2^\ell) \end{cases}$$

For every value of $q = 0, 1, \ldots, \ell - 1$, we have $m = 0, 1, \ldots, 2^q - 1$.

Table 1 shows the set of eight Haar function for $\ell = 3$. The Haar function, $H_{2^q+m}(x)$, has degree q and order m. Functions with the same degree are translations of each other.

The set $\{H_j(x) \ : \ j = 0, 1, 2, \ldots, 2^\ell - 1\}$ forms a basis for the fitness functions defined on the integers in $[0, 2^\ell)$. That is:

$$f(x) = \Sigma_{j=0}^{2^\ell-1} h_j \ H_j(x), \quad (4)$$

where the h_j's, for $j = 2^q + m$, are the Haar coefficients given by:

$$h_j = \frac{1}{2^{\ell-q}} \Sigma_{x=0}^{2^\ell-1} f(x) \ H_j(x). \quad (5)$$

x	$H_0(x)$	$H_1(x)$	$H_2(x)$	$H_3(x)$	$H_4(x)$	$H_5(x)$	$H_6(x)$	$H_7(x)$
000	1	1	1	0	1	0	0	0
001	1	1	1	0	-1	0	0	0
010	1	1	-1	0	0	1	0	0
011	1	1	-1	0	0	-1	0	0
100	1	-1	0	1	0	0	1	0
101	1	-1	0	1	0	0	-1	0
110	1	-1	0	-1	0	0	0	1
111	1	-1	0	-1	0	0	0	-1
index j	0	1	2	3	4	5	6	7
degree q	undefined	0	1	1	2	2	2	2
order m	undefined	0	0	1	0	1	2	3

Table 1. Haar functions $H_{2^q+m}(x)$ for $\ell = 3$.

As Equation 3 and Table 1 indicate, the higher the degree q, the smaller the subinterval with non-zero values for $H_j(x)$. Consequently, each Haar coefficient depends only on the local behavior of $f(x)$.

More precisely, from its definition (see Equation 3), we have that $H_{2^q+m}(x) \neq 0$ only for $m2^{\ell-q} \leq x < (m+1)2^{\ell-q}$. Every degree q partitions the interval $[0, 2^\ell)$ into pairwise disjoint subintervals:
$[0, 2^{\ell-q})$, $[2^{\ell-q}, (2)2^{\ell-q})$, $[(2)2^{\ell-q}, (3)2^{\ell-q})$, ..., $[(2^q-1)(2^{\ell-q}), 2^q(2^{\ell-q}))$, each of width $2^{\ell-q}$ and such that $H_{2^q+m}(x) = 0$ on all but one of the subintervals.

The search space contains 2^ℓ points and each subinterval will have $2^{\ell-q}$ points x such that $H_{2^q+m}(x) \neq 0$. Thus by the definition of h_{2^q+m} (see 5), there are at most $2^{\ell-q}$ non-zero terms in the computation. The following results are equivalent to Beauchamp's [2] concerning the linear combination of the Haar coefficients h_{2^q+m} where $m < 2^q$.

Result 1 Every Haar coefficient of degree q has at most $2^{\ell-q}$ non-zero terms. Each term corresponds to a point in an interval of the form $[(i)2^{\ell-q}, (i+1)2^{\ell-q})$. Consequently, the linear combination of each Haar coefficient h_j, where $j = 2^q + m$, has at most $2^{\ell-q}$ non-zero terms. In addition, h_0 has at most 2^ℓ non-zero terms. □

A similar result holds for the computation of $f(x)$ in Equation 4. In the linear combination, for a given x, only a few terms have non-zero values. Since $H_0(x) \neq 0$ and $H_1(x) \neq 0$ for all $x \in [0, 2^\ell)$, $H_0(x)$ and $H_1(x)$ appear in the right-hand side of Equation 5 for any given x. We have already seen that degree $q > 0$ partitions $[0, 2^\ell)$ into $2^{\ell-q}$ pairwise disjoint subintervals:
$[0, 2^{\ell-q})$, $[2^{\ell-q}, (2)2^{\ell-q})$, $[(2)2^{\ell-q}, (3)2^{\ell-q})$, ..., $[(2^q-1)(2^{\ell-q}), 2^q(2^{\ell-q}))$, each of width $2^{\ell-q}$ and such that $H_{2^q+m}(x) = 0$ except on the subinterval $[(m)2^{\ell-q}, (m+1)2^{\ell-q})$ for $m = 0, 1, \ldots, 2^q - 1$. Hence, for a given $x \in [0, 2^\ell)$, and a given q, $H_{2^q+m}(x)$ is non-zero for $m = i$, and zero for all other values of m. Thus, each degree q contributes at most one non-zero Haar function in the right-hand side of Equation 4, which can be rewritten as:

$$f(x) = h_0 H_0(x) + h_1 H_1(x) + \Sigma_{q=1}^{\ell-1} \Sigma_{m=0}^{2^q-1} h_{2^q+m} H_{2^q+m}(x) \tag{6}$$

For each degree q, $\Sigma_{m=0}^{2^q-1} h_{2^q+m} H_{2^q+m}(x)$ has at most one non-zero term. From Equation 6, the total number of non-zero terms is at most $2 + (\ell - 1) = \ell + 1$. We have shown the following result [10]:

Result 2 For any fixed value $x \in [0, 2^\ell)$, $f(x)$ has at most $\ell + 1$ non-zero terms. □

According to Result 1, every Haar coefficient of degree q, $q > 1$, has at most $2^{\ell-q}$ non-zero terms in its computation (Equation 5). Since however, Walsh functions are never zero, each Walsh coefficient can be written as a linear combination of at most 2^ℓ non-zero terms (see Equation 2). According to Result 2, for any fixed value x, $f(x) = \Sigma_{j=0}^{2^\ell-1} h_j\ H_j(x)$, has at most $\ell+1$ terms. Again, since Walsh functions are never zero, at most 2^ℓ non-zero terms are required for the Walsh expansion (see Equation 1).

These results are illustrated by considering a very simple example from Goldberg's textbook [6].

Example 1 We seek to maximize the elementary function $f(x) = x^2$ where x is an integer and $0 \leq x \leq 7$. Here $\ell = 3$, and the Walsh and Haar coefficients are computed and tabulated in Table 2.

j	0	1	2	3	4	5	6	7
Walsh coefficient w_j	17.5	−3.5	−7	1	−14	2	4	0
Haar coefficient h_j	17.5	−14	−3	−11	−0.5	−2.5	−4.5	−6.5

Table 2. Walsh and Haar coefficients for $f(x) = x^2$ with $\ell = 3$.

Thus, $f(x)$ can be written as the following linear combination of the Walsh monomials:

$$f(x) = 17.5\Psi_0(x) - 3.5\Psi_1(x) - 7\Psi_2(x) + \Psi_3(x) - 14\Psi_4(x) + 2\Psi_5(x) + 4\Psi_6(x).$$

Likewise, $f(x)$ can be expressed as a linear combination of Haar functions:

$f(x) = 17.5H_0(x) - 14H_1(x) - 3H_2(x) - 11H_3(x) - 0.5H_4(x) - 2.5H_5(x) - 4.5H_6(x) - 6.5H_7(x)$.

Note that the computation of w_6, for instance, has eight terms (see Equation 2) requiring the values of $f(x)$ at all points in the interval $[0, 8)$.

$$w_6 = \frac{1}{8}[f(0) + f(1) - f(2) - f(3) - f(4) - f(5) + f(6) + f(7)] = 4.$$

On the other hand (see Result 1 where $q = 2$ since $6 = 2^2 + 2$), h_6 requires the values of the function at the two points $x = 4$ and $x = 5$ only, since $H_6(x) = 0$ for all other values of $x \in [0, 8)$.

$$h_6 = \frac{1}{2^{3-2}}[f(4)\ H_6(4) + f(5)\ H_6(5)] = \frac{1}{2}[16(1) + 25(-1)] = -\frac{9}{2}.$$

Similarly, $f(5)$ will require the computation of eight Walsh terms (see Equation 1) instead of just four Haar terms (see Result 2).

$$\begin{aligned} f(5) &= \Sigma_{j=0}^{7} w_j\ \Psi_j(5) \\ &= 17.5(1) - 3.5(-1) - 7(1) + 1(-1) - 14(-1) + 2(1) + 4(-1) + 0(1) \\ &= 25. \end{aligned}$$

Since $H_2(5) = H_4(5) = H_5(5) = H_7(5) = 0$ (see Table 1), we have

$$\begin{aligned} f(5) &= h_0\ H_0(5) + h_1\ H_1(5) + h_3\ H_3(5) + h_6\ H_6(5) \\ &= h_0 - h_1 + h_3 - h_6 = 25. \end{aligned}$$

$\square$

Note that the total number of terms required for the computation of the expansion of $f(5)$ in the case of Walsh is 64 (8×8), and for Haar, 22 (8 each for h_0 and h_1; 4 for h_3 and 2 terms for h_5). Thus, the computation of 42 more terms is required with Walsh than with Haar. In practical cases, ℓ is substantially larger than 3. For instance, for $\ell = 20$ there are about $2^{40} \approx 10^{12}$ more terms using Walsh expansion.

No comparison between Walsh and Haar would be complete without considering fitness averages of schemata [4]. A comparison between the maximum number of non-zero terms, and the total number of terms for the computation of all 81 schemata of length $\ell = 4$ is tabulated in Figure 3. A fixed position is represented with "d" while "*" stands for a "don't care".

	Non-zero terms		Total number of non-zero terms	
Schema	Walsh	Haar	Walsh	Haar
****	1	1	16	16
***d	4	20	64	64
**d*	4	10	64	64
*d**	4	6	64	64
d***	4	4	64	64
**dd	16	36	256	160
*d*d	16	28	256	160
d**d	16	24	256	160
dd	16	20	256	160
d*d*	16	16	256	160
dd**	16	12	256	160
*ddd	64	56	1024	352
d*dd	64	48	1024	352
dd*d	64	40	1024	352
ddd*	64	32	1024	352
dddd	128	80	2048	736
Total	469	433	7952	3376

Table 3. Computing schemata for $\ell = 4$ with Walsh and Haar functions.

Consider, for example, the row in Table 3 corresponding to d**d. It represents four schemata, $E = \{0{*}{*}0, 0{*}{*}1, 1{*}{*}0, 1{*}{*}1\}$. The average fitness of each one can be expressed as a linear combination of at most four non-zero Walsh coefficients. For instance,

$$f(1{*}{*}0) = w_0 + w_1 - w_8 - w_9.$$

Since E has four schemata, the maximum number of non-zero terms for all schemata represented by d**d is $4 \times 4 = 16$ and tabulated in the second column of Table 3. Moreover, each single Walsh coefficient requires 16 terms for its computation (see Equation 2). Thus the total number of terms required in the computation of the expansion of $f(1{*}{*}0)$ is 4×16; and that of all schemata represented by d**d, 4×64 reported in the third column. On the other hand, the average fitness of each schema in E can be expressed as a linear combination of at most six non-zero Haar coefficients. For instance,

$$f(1{*}{*}0) = h_0 - h_1 + \frac{1}{4}(h_{12} + h_{13} + h_{14} + h_{15}). \tag{7}$$

The third column's entry has therefore the value 4×6. It might thus appear easier to use Walsh functions for this fitness average. Nevertheless, according to Result 1, only two terms in Equation are required for the computation of each of h_{12}, h_{13}, h_{14}, and h_{15}, while 16 are needed for h_0 and 16 for h_1. Likewise, it can be shown that 40 terms are required in the computation of the expansion of the other three schemata in E, bringing the total to 6×40 as reported in the last column of Table 3. As can be seen in the last row of Table 3, a substantial savings can be achieved by using Haar instead of Walsh functions.

With respect to fast transforms, by taking advantage of the many repetitive computations performed with orthogonal transformations, they can be implemented with on the order of at most $\ell 2^\ell$ computations in the case of fast Walsh transforms [13] and on the order of 2^ℓ for the fast Haar transforms [12]. With these implementations, modeled after the fast Fourier transform, the difference between the total number of terms required for the computation of the expansions of Walsh and of Haar still remains exponential in ℓ (of order $\ell 2^\ell$). Because of the interdependent nature of fast transforms, the computation of a single term is built upon the values of previous levels: many more levels for Walsh than Haar. Thus, many more computations (of the order $\ell 2^\ell$) are required for the computation of a single Walsh term. Fast transforms are represented by layered flowcharts where an intermediate result at a certain stage is obtained by adding (or subtracting) two intermediate results from the previous layer. Thus, when dealing with fast transforms, it is more appropriate to count the number of operations (additions or subtractions) which is equivalent to counting the number of terms [12]. It can be shown from [12] and [13] that exactly $\ell 2^\ell - 2^{\ell+1} + 2$ more operations are needed with Walsh than with Haar functions. For instance, for $\ell = 20$, one needs to perform $18,875,002$ more operations when the Walsh fast transform is used instead of the Haar transform. We conclude this section by noting that the Haar transforms "are the fastest linear transformations presently available" [2, page 80].

Conclusion

This work highlights the computational advantages that Haar functions have over Walsh monomials. The former can thus be used as practical transforms for discrete objective functions in optimization problems. More precisely, the total number of terms required for the computation of the expansion of the fitness function $f(x)$ for a given x using Haar is of order 2^ℓ which is substantially less than Walsh's $2^{2\ell}$. Similarly, we have seen that while w_j depends on the behaviour of $f(x)$ at all 2^ℓ points, h_j depends only on the local behaviour of $f(x)$ at a few points which are "close together", and furthermore, the advantage of Haar over Walsh remains very large (of order $\ell 2^\ell$) if fast transforms are used. One more advantage Haar functions have over Walsh is evident when they are used to approximate continuous functions. Walsh expansions might diverge at some points, whereas Haar expansions always converge [1, page 62].

Acknowledgments

The author would like to thank Aïda Batarekh, Dr. Frederick Stern, Thomas Bäck, Jörg Heitkötter, and one of the reviewers for their comments and very useful suggestions.

References

[1] Alexits, G., (1961), *Convergence Problems of Orthogonal Series*, Pergammon, New York and London.

[2] Beauchamp K. G., (1984), *Applications of Walsh and Related Functions*, Academic Press.

[3] Bethke , A. D., (1980), *Genetic Algorithms as Function Optimizers*, Doctoral dissertation, University of Michigan.

[4] Goldberg, D. E., (1989), *Genetic Agorithms and Walsh Functions: Part I, A Gentle Introduction*, Complex Systems 3:129-152.

[5] Goldberg, D.E., (1989), *Genetic Algorithms and Walsh Functions Part II: Deception and its Analysis*, Complex Systems 3, pp. 153-171.

[6] Goldberg, D.E., (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

[7] Haar, A., (1910), *Zür Theorie der Orthogonalen Funktionensysteme*, Math. Annalen 69:331-371.

[8] Harmuth, H. F., (1968), *A Generalized Concept of Frequency and Some Applications*, IEEE Transactions on Information Theory, IT-14:375-382.

[9] Holland, J.H., (1975), *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor.

[10] Karpovsky, M.G., (1985), *Spectral Techniques and Fault Detection*, Academic Press.

[11] Kremer, H., (1973), *On Theory of Fast Walsh Transform Algorithms*, Colloquium on the Theory and Applications of Walsh and Other Non-Sinusoidal Functions, Hatfield, U.K.

[12] Roeser, P. R., and Jernigan, M. E., (1982), *Fast Haar Transform Algorithms*, IEEE Transactions on Computers C-31 no 2:175-177.

[13] Shanks, J. L., (1969), *Computation of the Fast Walsh-Fourier Transform*, IEEE Transactions on Computers C-18:457-459.

[14] Walsh, J.L., (1923), *A Closed Set of Orthogonal Functions*, Ann. J. Math. 55:5-24.