B2.6.2 Transform methods: Haar analysis, Haar transforms

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

Abstract

As mentioned in section B2.6.1, theoretical analysis of fitness functions in genetic algorithms has included the use of Walsh functions since they form a convenient basis for the expansion of fitness functions. They have been used to compute the average fitness values of schemata, to decide whether a certain function is hard or easy for a genetic algorithm, and to design deceptive functions for the genetic algorithm. This work explores the use of Haar functions for the same purposes, and highlights the computational advantages that they have over Walsh functions.

Introduction

Walsh functions, which take values ± 1, form a complete orthogonal set of functions and can be used as a basis, 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 (Bethke 1980, Goldberg 1989a, 1989b). Any fitness function defined on binary strings of length ℓ can be represented as a linear combination of discrete Walsh functions.

If $\Psi_j(x)$ denotes a Walsh function, then $\{\Psi_j(x) \;:\; j = 0, 1, 2, \ldots, 2^\ell - 1\}$ forms a basis for the fitness functions defined on $[0, 2^\ell)$, then

$$f(x) = \Sigma_{j=0}^{2^\ell - 1} w_j \Psi_j(x), \qquad \text{(B2.6.1)}$$

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

$$w_j = \frac{1}{2^\ell} \Sigma_{x=0}^{2^\ell - 1} f(x) \Psi_j(x). \qquad \text{(B2.6.2)}$$

This section explores the use of another orthogonal, rectangular function: Haar functions (Haar 1910), that can be used as a convenient basis for the expansion of fitness functions. 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.

Haar Functions

The set of Haar functions also forms a complete set of orthogonal rectangular basis functions. These functions were proposed by the Hungarian mathematician Haar (1910), approximately 13 years before Walsh's work. Haar functions take values of 1, 0 and -1, multiplied by powers of $\sqrt{2}$. The interval Haar functions are defined on is usually normalized to $[0, 1)$ [e.g. Kremer 1973].

One could also use the unnormalized Haar functions, taking values of 0 and ± 1, [e.g. (Khuri 1994)]. In the following definition, the Haar functions are defined on $[0, 2^\ell)$, and are normalized,

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

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

$$H_3(x) = \begin{cases} 0 & \text{for } 0 \le x < 2^{\ell-1} \\ \sqrt{2} & \text{for } (2)2^{\ell-2} \le x < (3)2^{\ell-2} \\ -\sqrt{2} & \text{for } (3)2^{\ell-2} \le x < (4)2^{\ell-2} \end{cases}$$

$$\vdots$$

$$H_{2^q+m}(x) = \begin{cases} (\sqrt{2})^q & \text{for } (2m)2^{\ell-q-1} \le x < (2m+1)2^{\ell-q-1} \\ -(\sqrt{2})^q & \text{for } (2m+1)2^{\ell-q-1} \le x < (2m+2)2^{\ell-q-1} \\ 0 & \text{elsewhere in } [0, 2^\ell) \end{cases} \quad \text{(B2.6.3)}$$

$$\vdots$$

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

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 functions 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:

$$f(x) = \Sigma_{j=0}^{2^\ell-1} h_j \, H_j(x), \quad \text{(B2.6.4)}$$

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

$$h_j = \frac{1}{2^\ell} \Sigma_{x=0}^{2^\ell-1} f(x) \, H_j(x). \quad \text{(B2.6.5)}$$

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	$\sqrt{2}$	0	2	0	0	0
001	1	1	$\sqrt{2}$	0	-2	0	0	0
010	1	1	-$\sqrt{2}$	0	0	2	0	0
011	1	1	-$\sqrt{2}$	0	0	-2	0	0
100	1	-1	0	$\sqrt{2}$	0	0	2	0
101	1	-1	0	$\sqrt{2}$	0	0	-2	0
110	1	-1	0	-$\sqrt{2}$	0	0	0	2
111	1	-1	0	-$\sqrt{2}$	0	0	0	-2
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 B2.6.3 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 B2.6.3), we have that $H_{2^q+m}(x) \neq 0$ only for $m2^{\ell-q} \le x < (m+1)2^{\ell-q}$. 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 B2.6.5), there are at most $2^{\ell-q}$ non-zero terms in the computation. The following results are equivalent to Beauchamp's concerning the linear combination of the Haar coefficients h_{2^q+m} where $m < 2^q$ (Beauchamp 1984).

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 B2.6.4. 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 B2.6.5 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 B2.6.4, which can be rewritten as:

$$f(x) = h_0 H_0(x) + h_1 H_1(x) + \Sigma_{q=1}^{\ell-1} \Sigma_{m=0}^{2^q-1} h_{2^q+m} H_{2^q+m}(x) \tag{B2.6.6}$$

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 B2.6.6, the total number of non-zero terms is at most $2 + (\ell - 1) = \ell + 1$. We have shown the following result (Karpovsky 1985):

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 B2.6.5). 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 B2.6.2). 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 B2.6.1). These results are illustrated by considering a simple problem instance of the integer knapsack problem.

Example

The integer knapsack problem consists of a knapsack of capacity M, and of ℓ objects with associated weights $w_1, \ldots, w_\ell$, and profits $p_1, \ldots, p_\ell$. The knapsack is to be filled with some of the objects without exceeding M, and such as to maximize the sum of the profits of the selected objects. In other words:

Maximize $\Sigma_{i \in S} p_i$ where $S \subseteq \{1, 2, \ldots, \ell\}$ and $\Sigma_{i \in S} w_i \leq M$.

The solution space can be encoded as 2^ℓ binary strings $x_\ell x_{\ell-1} \ldots x_1$ where $x_i = 1$ means that object i is placed in the knapsack and $x_i = 0$ means that object i is not selected. Each string $\vec{x} = x_\ell \ldots x_1$ where $x_i \in \{0, 1\}$, and $1 \leq i \leq \ell$, has a profit associated with it: $P(\vec{x}) = \Sigma_{i=1}^{\ell} x_i\, p_i$. Some strings represent infeasible solutions. The weight of the solution is given by $W(\vec{x}) = \Sigma_{i=1}^{\ell} x_i w_i$. Infeasible strings are those whose weight $W(\vec{x})$ is greater than M.

The fitness $f(\vec{x})$ of strings is defined as follows: $f(\vec{x}) = P(\vec{x}) - \mathit{penalty}(\vec{x})$ where $\mathit{penalty}(\vec{x}) = 0$ if $\vec{x}$ is feasible. The penalty is a function of the weight and profit of the string, and of the knapsack capacity (Khuri and Batarekh 1990).

We now consider the following problem instance:

Objects:	4	3	2	1	
Weights:	10	5	8	9	and $M = 23$
Profits:	15	8	10	6	

Table 2 gives the fitnesses of the strings $\vec{x} = x_4 x_3 x_2 x_1$ after a penalty has been applied to the infeasible strings (strings 1011, 1101, and 1111). The Walsh and Haar coefficients are given in Table 3.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(x)$	0	6	10	16	8	14	18	24	15	21	25	12	23	10	33	3

Table 2. Fitness values for problem instance.

j	0	1	2	3	4	5	6	7
w_j	$\frac{238}{16}$	$\frac{26}{16}$	$\frac{-44}{16}$	$\frac{-36}{16}$	$\frac{-28}{16}$	$\frac{-36}{16}$	$\frac{2}{16}$	$\frac{-2}{16}$
h_j	$\frac{238}{16}$	$\frac{-46}{16}$	$\frac{-32}{16}\sqrt{2}$	$\frac{4}{16}\sqrt{2}$	$\frac{-40}{16}$	$\frac{-40}{16}$	$\frac{-2}{16}$	$\frac{-6}{16}$

j	8	9	10	11	12	13	14	15
w_j	$\frac{-46}{16}$	$\frac{-74}{16}$	$\frac{-36}{16}$	$\frac{36}{16}$	$\frac{-36}{16}$	$\frac{36}{16}$	$\frac{-2}{16}$	$\frac{2}{16}$
h_j	$\frac{-12}{16}\sqrt{2}$	$\frac{-12}{16}\sqrt{2}$	$\frac{-12}{16}\sqrt{2}$	$\frac{-12}{16}\sqrt{2}$	$\frac{-12}{16}\sqrt{2}$	$\frac{26}{16}\sqrt{2}$	$\frac{26}{16}\sqrt{2}$	$\frac{60}{16}\sqrt{2}$

Table 3. Walsh and Haar coefficients for the problem instance.

Note that the computation of w_{13}, for instance, has 16 terms (see Equation B2.6.2) requiring the values of $f(x)$ at all points in the interval $[0, 16)$.

$$\begin{aligned} w_{13} &= \frac{1}{16}[0 - 6 + 10 - 16 - 8 + 14 - 18 + 24 \\ &\quad -15 + 21 - 25 + 12 + 23 - 10 + 3 - 3] \\ &= \frac{36}{16}. \end{aligned}$$

On the other hand (see Result 1 where $q = 3$ and $m = 4$ since $13 = 2^3 + 4$), h_{13} requires the values of $f(x)$ at the two points: $x = 1010$ and $x = 1011$ only, since $H_{13} = 0$ for all other values of $x \in [0, 16)$.

$$\begin{aligned} h_{13} &= \frac{1}{16}[2\sqrt{2}(25) - 2\sqrt{2}(12)] \\ &= \frac{26}{16}\sqrt{2} \end{aligned}$$

Similarly, $f(11)$ (i.e., $x = 1011$) will require the computation of sixteen Walsh terms (see Equation B2.6.1) instead of just five Haar terms (see Result 2).

$$\begin{aligned} f(1011) &= \Sigma_{j=0}^{15} w_j \, \Psi_j(1011) \\ &= w_0 - w_1 - w_2 + w_3 + w_4 - w_5 - w_6 + w_7 - w_8 + w_9 + w_{10} \\ &\quad -w_{11} - w_{12} + w_{13} + w_{14} - w_{15} \\ &= \frac{1}{16}[238 - 26 + 44 - 36 - 28 + 36 - 2 - 2 \\ &\quad +46 - 74 - 36 - 36 + 36 + 36 - 2 - 2] \\ &= 12 \\ f(1011) &= \Sigma_{j=0}^{15} h_j \, H_j(1011) \\ &= h_0 - h_1 + 2\sqrt{2}h_3 - 2h_6 - 2\sqrt{2}h_{13} \\ &= \frac{1}{16}[238 + 46 + 8 + 4 - 104] \\ &= 12 \end{aligned}$$

Note that the total number of terms required for the computation of the expansion of $f(1011)$ in the case of Walsh is 256 (16×16), and for Haar, 46 (16 each for h_0 and h_1; 8 for h_3, 4 for h_6, and 2 terms for h_{13}). Thus, the computation of 210 more terms is required with Walsh than with Haar. In practical cases, ℓ is substantially larger than 4. 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 (Goldberg 1989a).

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 Table 4. 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 4. Computing schemata for $\ell = 4$ with Walsh and Haar functions. Reproduced from (Khuri 1994).

Consider, for example, the row in Table 4 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 4. Moreover, each single Walsh coefficient requires 16 terms for its computation (see Equation B2.6.2). 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,

$$f(1{*}{*}0) = h_0 - h_1 + \frac{1}{4}(h_{12} + h_{13} + h_{14} + h_{15}). \tag{B2.6.7}$$

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 B2.6.2 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 4. As can be seen in the last row of Table 4, a substantial savings can be achieved by using Haar instead of Walsh functions.

With respect to fast transforms, they can be implemented with on the order of at most $\ell 2^\ell$ computations in the case of fast Walsh transforms and on the order of 2^ℓ for the fast Haar transforms (Roeser and Jernigan 1982). With these implementations, 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 $\ell 2^\ell$). The computation of a single term in fast transforms 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 (Roeser and Jernigan 1982). It can be shown 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" (Beauchamp 1984).

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 (Alexits 1961).

References

Alexits G 1961 *Convergence Problems of Orthogonal Series* (New York and London: Pergammon)

Beauchamp K G 1984 *Applications of Walsh and Related Functions* (Academic Press)

Bethke A D 1980 *Genetic Algorithms as Function Optimizers* Doctoral dissertation University of Michigan

Goldberg D E 1989a Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction *Complex Systems* **3** 129–52

Goldberg D E 1989b Genetic Algorithms and Walsh Functions Part II: Deception and its Analysis *Complex Systems* **3** 153–71

Haar A 1910 Zur Theorie der Orthogonalen Funktionensysteme *Math. Annalen* **69** 331–71

Karpovsky M G 1985 *Spectral Techniques and Fault Detection* (Academic Press)

Khuri S and Batarekh A 1990 Heuristics for the Integer Knapsack Problem *Proc. of the Tenth Int. Computer Science Conf (Santiago, Chile)* 161–72

Khuri S 1994 Walsh and Haar Functions in Genetic Algorithms *Proc. of the 1994 ACM Symposium of Applied Computing* (New York: ACM Press) 201–5

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.)*

Roeser P R and Jernigan M E 1982 Fast Haar Transform Algorithms *IEEE Transactions on Computers* **31** 175–7

Walsh J L 1923 A Closed Set of Orthogonal Functions *Ann. J. Math* **55** 5–24

