Technical Report

A primer on Bethke’s Walsh schema transform

William A. Greene
Computer Science Department
University of New Orleans
New Orleans, LA 70148
wagcs@uno.edu
1993
A primer on Bethke’s Walsh schema transform

William A. Greene
Computer Science Department
University of New Orleans
New Orleans, LA 70148
wagcs@uno.edu

Abstract: In genetic algorithms, Bethke’s Walsh schema transform asserts that a schema’s average fitness is a certain alternating sum of the Walsh coefficients of those Walsh partition elements that subsume the schema. We provide a primer on this result. For readers familiar with linear algebra and Walsh functions, our proof is short and direct.

For completeness we begin with definitions and results from linear algebra. A fine reference is [3]. The reader already familiar with vector spaces can skip to section 2.

1. Vector space review

A vector space over the reals $\mathbb{R}$ is a set $V$ of vectors, equipped with two operations:

(i) addition of vectors, $+: V \times V \rightarrow V$, under which $V$ is a commutative group with identity, denoted $\emptyset$, the zero vector.

(ii) scalar multiplication, $\cdot: \mathbb{R} \times V \rightarrow V$ (normally one writes $\lambda \hat{v}$ instead of $\lambda \cdot \hat{v}$).

Additionally it is assumed the operations interact according to the rules (for arbitrary $\lambda, \mu \in \mathbb{R}, \hat{v}, \hat{w} \in V$):

(i) $1 \hat{v} = \hat{v}$

(ii) $\lambda(\hat{v} + \hat{w}) = \lambda\hat{v} + \lambda\hat{w}$

(iii) $(\lambda + \mu) \hat{v} = \lambda\hat{v} + \mu\hat{v}$

(iv) $\lambda(\mu \hat{v}) = (\lambda\mu)\hat{v}$

Certain properties follow, including:

$0\hat{v} = \emptyset$, any $\hat{v} \in V$, where $0$ is the zero of $\mathbb{R}$.

$\lambda\emptyset = \emptyset$, any $\lambda \in \mathbb{R}$.

Subset $W \subseteq V$ is a (vector) subspace of $V$ if the addition of any two elements of $W$ is again an element of $W$, and any scalar multiple of any element of $W$ is again an element of $W$. Thus, $W$ is a vector space under the operations inherited from $V$.

If $\Gamma$ is a set of vectors, then a linear combination of elements of $\Gamma$ is a vector of the form $\Sigma_{\text{finite}} \lambda_i \hat{v}_i$, where $\lambda_i \in \mathbb{R}$ and $\hat{v}_i \in \Gamma$. The set of linear combinations of elements of $\Gamma$ is a subspace of $V$. A set $\Gamma$ of vectors spans a subspace $W$ if $W$ is the same as the set of all linear combinations of elements of $\Gamma$. A set $\Gamma$ of vectors is linearly independent if the only linear combination $\Sigma \lambda_i \hat{v}_i$ of vectors $\hat{v}_i \in \Gamma$ that results in the zero vector $\emptyset$ is the trivial one: each $\lambda_i$ is the real zero $0$. A basis for a vector space $V$ is a set of vectors which are linearly independent and also span $V$. It is a well-
known fact from linear algebra that any two bases for \( V \) must have the same cardinality (same number of elements); this common cardinality is the **dimension** of \( V \).

\( \mathbb{R} \) is a vector space over itself, with dimension 1; any single non-zero element of \( \mathbb{R} \) is a basis.

A function \( f : V_1 \to V_2 \) between vector spaces is **linear** if 
\[
f(\lambda \hat{v} + \hat{w}) = \lambda f(\hat{v}) + f(\hat{w})
\]
It follows that in general 
\[
f(\sum \lambda_i \hat{v}_i) = \sum \lambda_i f(\hat{v}_i)
\]

An **inner product** on a vector space \( V \) is a function of two arguments \( \langle \cdot | \cdot \rangle : V \times V \to \mathbb{R} \), satisfying

1. it is (separately) linear in its two arguments, that is, for any \( \hat{v}, \hat{w} \), \( \langle \hat{v} | \hat{w} \rangle : V \to \mathbb{R} \) is linear, and also \( \langle \hat{v} | \cdot \rangle : V \to \mathbb{R} \) is linear;
2. \( \langle \hat{v} | \hat{w} \rangle \geq 0 \), all \( \hat{v}, \hat{w} \);
3. \( \langle \hat{v} | \hat{v} \rangle \geq 0 \), all \( \hat{v} \), and equals 0 only if \( \hat{v} = \emptyset \).

Two vectors \( \hat{v}, \hat{w} \) are orthogonal if \( \langle \hat{v} | \hat{w} \rangle = 0 \). Define \( \| \hat{v} \| = \sqrt{\langle \hat{v} | \hat{v} \rangle} \). [Then it so happens that \( \| \cdot \| \) is what is termed a **norm** (which concerns the notion of size) on \( V \), and setting \( d(\hat{v}, \hat{w}) = \| \hat{v} - \hat{w} \| \) defines what is termed a **metric** (the notion of distance) on \( V \).] A **unit vector** is a vector \( \hat{v} \) such that \( \| \hat{v} \| = 1 \).

If \( \Gamma \) is a basis for \( V \), then since \( \Gamma \) spans \( V \), any vector \( \hat{w} \in V \) can be represented \( \hat{w} = \sum \lambda_i \hat{v}_i \) as some linear combination of vectors \( \hat{v}_i \in \Gamma \), and since \( \Gamma \)'s elements are linearly independent, there is only one such representation. If additionally the elements of \( \Gamma \) are pairwise orthogonal unit vectors, then the scalars \( \lambda_j \) can be easily calculated:
\[
\langle \hat{w} | \hat{v}_j \rangle = \langle \sum \lambda_i \hat{v}_i | \hat{v}_j \rangle = \sum \lambda_i \langle \hat{v}_i | \hat{v}_j \rangle = \lambda_j \langle \hat{v}_j | \hat{v}_j \rangle = \lambda_j
\]
Here the orthogonality of \( \Gamma \) made the sum simplify to one term.

### 2. Function spaces

Let \( X \) be a finite set. Then the set of all functions \( f : X \to \mathbb{R} \) is a vector space over \( \mathbb{R} \) under the point-wise operations:

\[
(f + g)(x) = f(x) + g(x)
\]
\[
(\lambda f)(x) = \lambda f(x), \text{ any } x \in X
\]

Letting \( |X| \) denote the cardinality of \( X \), then
\[
\langle f | g \rangle = \frac{1}{|X|} \sum_{x \in X} f(x) g(x)
\]
defines an inner product on this vector space. A basis for this vector space is provided by the set of functions \( \{ \delta_x \}_{x \in X} \) where \( \delta_x \) is defined \( \delta_x(y) = 1 \) if \( x=y \) but \( \delta_x(y) = 0 \) otherwise; \( \delta_x \) is termed point-mass at \( x \) (and sometimes termed a Kronecker delta function). It is easy to see that this family of functions forms an orthogonal basis (though not of unit vectors); the corresponding unique representation of a vector (which you recall is a function) is \( f = \sum_{x \in X} f(x) \delta_x \); i.e., the coefficient \( \lambda_x \) for \( \delta_x \) is \( f(x) \). In particular we note that the dimension of this vector space is \( |X| \).

### 3. Walsh schema transform

Now we consider the set \( \{0,1\}^l \) of length \( l \) bit strings, which we denote \( \Pi \) (for patterns). The set of functions \( f : \Pi \to \mathbb{R} \) is a vector space \( V \) over \( \mathbb{R} \), with dimension \( |\Pi| = 2^l \). We equip this vector space with the inner product

\[
\cal{R}
\]
In preview, we alert the reader that, as will be seen, a length $l$ bit pattern (an element of $\{0,1\}^l$) will wear two hats. Thought of as an element $x$ in the domain $\Pi$, it can be an argument to a Walsh (or any other) function, and can belong to the subset determined by a schema. Thought of as an index $j$ of a Walsh function, it collects together a family of schema, which family partitions $\Pi$.

For each length $l$ bit pattern $j$, define the $j$th Walsh function $\Psi_j : \Pi \to \mathbb{R}$ as follows: Let $x$ be an arbitrary element of the domain $\Pi$. If the bitwise-and $x \wedge j$ has an even number of 1’s, then $\Psi_j(x) = +1$, else $\Psi_j(x) = -1$. Put another way: the 1’s in bit pattern $j$ act as a mask on the bits of $x$; if the masked bits of $x$ have even parity then $\Psi_j(x) = +1$, else $\Psi_j(x) = -1$. The 1’s in $j$ are termed its **masking bits**.

**Claim 1:** The set of functions $\{\Psi_j\}_{j \in \Pi}$ are pairwise orthogonal.

**Proof:** If $j \neq k$ then they differ at some bit position $i$; then one of $j,k$ masks position $i$ but the other does not. Define the function $c : \Pi \to \Pi$ by $c(x) =$ the same bit pattern as $x$ except that one bit, the $j$th one, has been flipped. Note that $c(c(x)) = x$ and that mapping $c$ is one-to-one and onto. Compared to the values of $\Psi_j(x)$, $\Psi_k(x)$, exactly one of $\Psi_j(c(x))$, $\Psi_k(c(x))$ has changed sign. Therefore in the sum $\sum_{x \in \Pi} \Psi_j(x)\Psi_k(x)$, each addend $\Psi_j(x)\Psi_k(x)$ is cancelled out by the corresponding one $\Psi_j(c(x))\Psi_k(c(x))$ and so the sum is zero. Therefore $\langle \Psi_j \mid \Psi_k \rangle$ is zero. ❄

It is easily seen that each $\Psi_j$ is a unit vector. Since there are enough $\Psi_j$’s, namely $2^l$ of them, we conclude $\{\Psi_j\}_{j \in \Pi}$ form a basis (of orthogonal unit vectors) for vector space $V$. Any function $f : \Pi \to \mathbb{R}$ is representable $f = \sum_{j \in \Pi} \omega_j \Psi_j$ where the $j$th *(Walsh)* coefficient $\omega_j$ for $f$ must be $\omega_j = \langle f \mid \Psi_j \rangle$.

Now we consider **schemas**. A length $l$ schema is any element of $\{0,1,*\}^l$, where * is termed the **don’t-care** symbol. The components of a schema which are 0 or 1 (i.e., which are not the don’t-care symbol *) are said to be **fixed**.

If $j$ is a length $l$ bit pattern (such as used to index the Walsh functions), define its order $o(j)$ to be the number of 1’s in $j$ (the number of masking bits in $j$). Such a $j$ can be considered to correspond (or collect) a certain family of schema, namely: the schemas associated with $j$ are the ones whose fixed bits occur in exactly the same bit positions where $j$’s masking bits (1’s) occur. Plainly there are $2^{o(j)}$ schemas associated with $j$.

On the other hand, each schema $\sigma$ determines a subset $S_{\sigma} \subseteq \Pi$, namely $x \in S_{\sigma}$ if and only if $x$ is identical to $\sigma$ in all bit positions where $\sigma$ has fixed bits 0,1 (off the fixed bits of $\sigma$, we “don’t care” what $x$ is). If we define the order $o(\sigma)$ of $\sigma$ to be the number of fixed bits (the number of non-*’s) in $\sigma$, then the cardinality of $S_{\sigma}$ is $2^{l-o(\sigma)}$ (there are as many elements in $S_{\sigma}$ as there are ways of “fix”ing the *’s). We often abuse notation and identify schema $\sigma$ with its associated $\Pi$-subset $S_{\sigma}$.
Let j be a length l bit pattern. A little reflection shows that the schemas σ that are associated with j partition the domain Π. For this reason, j is sometimes termed a Walsh partition. Additionally we observe that Ψ_j is constant (with constant value +1 or −1) on the subset S_σ of any schema σ associated with j.

We say partition j subsumes schema σ if σ is a subset of one of the schemas associate with j. Equivalently, each masking bit of j masks a fixed bit of σ (each 1 in j masks a non-* in σ). Plainly there are 2^{o(σ)} j’s that subsume σ. The lower the order of σ, the fewer the j’s that subsume it.

Let σ be a schema, (let us identify σ with S_σ,) and define its characteristic function χ_σ by χ_σ(x) = 1 if x ∈ σ, but χ_σ(x) = 0 otherwise. Function χ_σ, like any other function, has a Walsh representation, χ_σ = Σ_j ω_j,σΨ_j, where ω_j,σ = ⟨χ_σ|Ψ_j⟩ = 1/2^1 Σ_{x∈Π} χ_σ(x)Ψ_j(x) = 1/2^1 Σ_{x∈σ} Ψ_j(x). If j subsumes σ then Ψ_j is constant over subset σ with a constant value that is either +1 or −1; we denote this constant value by Ψ_j(σ). In this case ω_j,σ = Ψ_j(σ) / 2^{o(σ)}.

Claim 2: If j subsumes σ then ω_j,σ = Ψ_j(σ) / 2^{o(σ)}, otherwise ω_j,σ = zero.

Proof: It only remains to prove the second assertion. If j does not subsume σ, some masking bit in j occurs in the same bit position, say i, as a * occurs in σ. Define c:Π → Π by c(x) = the same bit pattern as x, but after flipping the i'th bit in x. Note x ∈ σ if and only if c(x) ∈ σ, and that Ψ_j(x) and Ψ_j(c(x)) are of opposite sign. Therefore these two terms cancel in the sum Σ_{x∈σ} Ψ_j(x), and hence ω_j,σ = 0.

Now let F :Π → ℜ be any function, such as the fitness function of a genetic algorithm. The average value of F over (the subset of) schema σ is

\[ \frac{1}{|σ|} \sum_{x \in σ} F(x) = \frac{1}{|σ|} \sum_{x \in Π} F(x) χ_σ(x) = 2^{o(σ)}⟨F|χ_σ⟩ \]

\[ = 2^{o(σ)}⟨Σ_i Ω_i,σ F Ψ_j|Σ_k Ω_k,σ Ψ_k⟩ = 2^{o(σ)}Σ_j Ω_j,σ F Ψ_j(σ) \]

where the last equality holds because the Ψ_j’s are orthogonal unit vectors. Continuing, the last expression =

\[ 2^{o(σ)} Σ_{σ,j} Ω_j,σ F Ψ_j(σ) \]

Thus we have Bethke’s result:

The Walsh schema transform: The average value of F over schema σ equals

\[ 2^{o(σ)} Ω_j,σ F Ψ_j(σ) \]

where Ψ_j(σ) is the constant value ±1 that Ψ_j takes on over schema σ, and Ω_j is F’s jth Walsh coefficient.
References:
