Complex Systems 6 (1992) 125–135

Walsh Functions, Schema Variance, and Deception

Scott E. Page
Department of Managerial Economics and Decision Sciences,
J. L. Kellogg Graduate School of Management,
Evanston, IL 60208, USA

David W. Richardson
Department of Economics, Northwestern University,
Evanston, IL 60208, USA

Abstract. In this paper we show how the Walsh functions can be used to compute schema variance and relate schema variance to deception. We also calculate operator-adjusted fitness for Walsh functions.

1. Introduction

The Walsh functions are an alternative basis for binary strings. They can be used to compute schema average, approximate functions, and create functions of varying complexity. In this paper we show how the Walsh functions can also be used to compute schema variance and explore the impact of variance on selection and deception. We follow Goldberg [3, 4], who both clarified and extended Bethke's dissertation [1], which introduced the Walsh functions to the study of genetic algorithms.

The extant theory of genetic algorithm performance relies heavily on the schema theorem, which places a lower bound on schema growth as a function of relative fitness, defining length and crossover and mutation rates. Recently the schema theorem has been criticized along a variety of fronts (see [5], among others). In particular, the schema theorem's snapshot approach may not fully capture the subtleties of genetic selection. In subsequent generations, a genetic algorithm's population will be biased toward high-variance schemata. Given Weitzman's [9] result—that when maximizing the present discounted value of search from many sources, the weight of a source's distribution in the good tail is of primary importance—this bias may be beneficial. For sources drawn from the same family of distributions, tail weight will be positively correlated with variance. Including schema variance in the theory of genetic algorithm performance may lead to greater insight. The extent to which schema variance proves to be an effective tool depends partially on the ease with which variance can be calculated.

© 1992 Complex Systems Publications, Inc.

In the next section of the paper we restate much of what Goldberg [3] restated from Bethke [1]: the definition of the Walsh functions, their use as an alternative basis, and the formula for schema average. In section 3 we examine the schema theorem and operator-adjusted Walsh coefficients, arriving at a different result than Goldberg [3]. In sections 4 and 5 we present the main results of the paper, an explicit formula for schema variance as a function of the w_i and a relationship between variance and deception. The conclusion elaborates on the importance of schema variance.

2. Genetic algorithms and Walsh functions

Many applications of genetic algorithms (GAs) encode the domain as binary strings. This encoding creates exploitable characteristics. One of the more important is the easy classification of a large number of subsets called schemata. Strings of length N have 3^N schemata. Comparing functional values over schemata (often referred to as hyperplanes, or when they are large subsets, building blocks) provides a means for evaluation and explanation of the performance of GAs and other search algorithms.

The huge number of schemata—the ratio of schemata to strings is $(3/2)^N$—can render the use of schemata values useless as an analytical tool. This criticism disappears provided two conditions are met. First, the set of informative schemata, the schemata with positive values, must be relatively small. Second, a method must exist for finding the informative schemata. The Walsh functions (WFs) not only determine schemata values, but also can be used to create small numbers of informative schemata (see [8]).

Formally, the domain can be encoded as binary strings, the schemata as ternary strings, and the WFs as binary strings:

a string,	$\mathbf{s} = x_1 \ldots x_N$	$x_i \in \{0,1\}$	(d1)
a Walsh Function,	$\boldsymbol{\phi} = y_1 \ldots y_N$	$y_i \in \{0,1\}$	(d2)
a schema,	$\mathbf{h} = z_1 \ldots z_M$	$z_i \in \{0,1,*\}$	(d3)

Let $S = 2^N$, the total number of strings and WFs. The strings and the WFs can be runumbered from 0 to $S-1$ with the standard map from base 2 to base 10. The objective function maps the set of strings into the positive real numbers:

$$f : \{0,1\}^N \rightarrow R^+ \qquad \text{(d4)}$$

Having assigned values to each string, the same must be done for each WF. Each WF can be viewed as a map from the strings into the set $\{1,-1\}$ (see [7] for an alternative treatment),

$$\begin{aligned} \boldsymbol{\phi}(\mathrm{s}) &= 1 \quad \text{if} \quad \boldsymbol{\phi}\cdot\mathrm{s} = 2k \qquad\; k \in \{0,1,2,\ldots\} \\ &= -1 \quad \text{if} \quad \boldsymbol{\phi}\cdot\mathrm{s} = 2k+1 \quad k \in \{0,1,2,\ldots\} \end{aligned} \qquad \text{(d5)}$$

where $\boldsymbol{\phi}\cdot\mathbf{s}$ is the standard dot product.

The following lemmas will be important in later claims.

Lemma 0.A. *For all $i > 0$, the set $\{\mathbf{s} : {}^i\boldsymbol{\phi}(\mathbf{s}) = 1\}$ has cardinality $S/2$.*

Proof. Let $\mathbf{h}_1 = 1$ and $\mathbf{h}_j = *$ for $j > 1$, and $\mathbf{h}^{\mathbf{c}}_1 = 0$ and $\mathbf{h}^{\mathbf{c}}_j = *$ for $j > 1$. The following four sets of strings partition the space:

$$\begin{aligned}
\{\mathbf{s} : \mathbf{s} \in \mathbf{h} \text{ and } \boldsymbol{\phi}(\mathbf{s}) &= 1\} = A_1 \\
\{\mathbf{s} : \mathbf{s} \in \mathbf{h} \text{ and } \boldsymbol{\phi}(\mathbf{s}) &= -1\} = A_2 \\
\{\mathbf{s} : \mathbf{s} \in \mathbf{h}^{\mathbf{c}} \text{ and } \boldsymbol{\phi}(\mathbf{s}) &= 1\} = A_3 \\
\{\mathbf{s} : \mathbf{s} \in \mathbf{h}^{\mathbf{c}} \text{ and } \boldsymbol{\phi}(\mathbf{s}) &= -1\} = A_4
\end{aligned}$$

From the definition of $\boldsymbol{\phi}(\cdot)$, $|A_1| = |A_4|$ and $|A_2| = |A_3|$, which completes the proof. ∎

Lemma 0.B. *For any two unique WFs $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$, the set $\{\mathbf{s} : \boldsymbol{\phi}(\mathbf{s}) = \boldsymbol{\phi}'(\mathbf{s})\}$ has cardinality $S/2$.*

Proof. Uniqueness implies that there exists a bit where the WFs differ. Wolog assume that $\boldsymbol{\phi}_1 = 1$ and $\boldsymbol{\phi}'_1 = 0$. Define the sets A_i similar to above, that is, let

$$\{\mathbf{s} : \mathbf{s} \in \mathbf{h} \text{ and } \boldsymbol{\phi}(\mathbf{s}) = \boldsymbol{\phi}'(\mathbf{s})\} = A_1$$

and so on. Again the cardinalities of A_1 and A_4 are equal, which completes the proof. ∎

The coefficient of ${}^i\boldsymbol{\phi}$ given the function f will be referred to as the Walsh coefficient and denoted by $w_i(f)$:

$$w_i(f) = \left[\sum_{\mathbf{s}=0}^{S-1} f(\mathbf{s}) \cdot {}^i\boldsymbol{\phi}(\mathbf{s})\right] \Big/ S \tag{d6}$$

The WFs form a basis over R^S [7]. Let the vector $\mathbf{f} \in R^S$ represent the function values on the S strings and $\mathbf{w}$ denote the vector of w_i's for a given $\mathbf{f}$, the vector of Walsh coefficients. The Walsh matrix $\mathbf{M}$ can be created from (d6) where $\mathbf{w} = \mathbf{f} \cdot \mathbf{M}$. Equation (d7) gives the string values as a function of the w_i's, and can be used to create $\mathbf{M}^{-1}$:

$$f(\mathbf{s}) = \sum_{i=0}^{N-1} w_i \cdot {}^i\boldsymbol{\phi}(\mathbf{s}) \tag{d7}$$

To compute schema average using the WFs, each schema must be mapped to a string using the vector-valued function $\boldsymbol{\tau}$. The WFs are then applied to the string $\boldsymbol{\tau}(\mathbf{h})$:

$$\begin{aligned}
&\boldsymbol{\tau} : \{0,1,*\}^N \to \{0,1\}^N \text{ according to the following rule:} \\
&\quad \boldsymbol{\tau}_i(1) = 1 \\
&\quad \boldsymbol{\tau}_i(0) = \boldsymbol{\tau}_i(*) = 0
\end{aligned} \tag{d8}$$

$$\begin{aligned}
&\boldsymbol{\phi}_i(\cdot) : \{0,1,*\}^N \to \{-1,1\} \text{ according to the following rule:} \\
&\quad {}^i\boldsymbol{\phi}(\mathbf{h}) = 1 \quad \text{iff} \quad {}^i\boldsymbol{\phi} \cdot \boldsymbol{\tau}(\mathbf{h}) = 2k \qquad\quad k \in \{0,1,2,\ldots\} \\
&\quad {}^i\boldsymbol{\phi}(\mathbf{h}) = -1 \quad \text{iff} \quad {}^i\boldsymbol{\phi} \cdot \boldsymbol{\tau}(\mathbf{h}) = 2k+1 \quad k \in \{0,1,2,\ldots\}
\end{aligned} \tag{d9}$$

The following definition is needed to formulate Claim 1, which computes schema average as a function of the Walsh coefficients:

$$d(\mathbf{h}) = \{i : {}^i\phi_j = 0 \text{ if } \mathbf{h}_j = *\} \tag{d10}$$

Claim 1. *The average value of a schema* $\mathbf{h}$, *namely* $\mathbf{h}^\mu$, *is given by*

$$\mathbf{h}^\mu = \sum_{i \in d(\mathbf{h})} w_i \cdot {}^i\phi(\mathbf{h})$$

Proof. Suppose $i \in d(\mathbf{h})$. If ${}^j\mathbf{h}_j = *$ then ${}^i\phi_j = 0$. This implies that, for any $\mathbf{s} \in \mathbf{h}$, ${}^i\phi(\mathbf{s}) = {}^i\phi(\mathbf{h})$, so the term ${}^i\phi(\mathbf{h})$ appears in the schema average.

Suppose that i is not a member of $d(\mathbf{h})$. In Lemma 0.B let $\phi' = {}^0\phi$ and $\phi = {}^j\phi$. Half of the strings in $\mathbf{h}$ satisfy ${}^j\phi(\mathbf{s}) = 1$ and half satisfy ${}^j\phi(\mathbf{s}) = -1$. In computing the average value, w_j will be added and subtracted an equal number of times. ∎

As a corollary to Claim 1, the average value of the trivial schema $\{*\}^N$, namely w_0, equals the average string value in the population. Claim 1 hints at an advantage of the Walsh basis as opposed to the string values. If most of the Walsh coefficients are zero, then the Walsh basis allows for easier calculation of schemata values. If all of the w_i have nonzero values, then the Walsh basis offers no comparative advantage for computing averages. Goldberg [3, 4] and Tanese [8] show that many interesting functions can be created by assigning positive values to a relatively small number of Walsh coefficients.

3. Operator-adjusted Walsh coefficients

The schema theorem gives a lower bound for the number of strings in generation $t+1$ that lie in schema $\mathbf{h}$, $n_{t+1}(\mathbf{h})$, as a function of the number in generation t, $n_t(\mathbf{h})$; the crossover rate, p_c; the maximum space between two defined bits, $\delta(\mathbf{h})$; the average fitness of strings in the current population that lie in the hyperplane, $f(\mathbf{h})$; the probability of mutation, p_m; the number of defined bits (the order), $o(\mathbf{h})$; and the average value of all strings in the population, f_{ave}.

The Schema Theorem.

$$n_{t+1}(\mathbf{h}) \geq n_t \cdot (f(\mathbf{h})/f_{\text{ave}}) \cdot \left[1 - \left(\frac{p_c \cdot \delta(\mathbf{h})}{N-1}\right) - p_m \cdot o(\mathbf{h})\right]$$

Proof. See Holland [6]. ∎

The schema theorem provides insight into and motivation for the performance of GAs. It formalizes the idea that good hyperplanes reproduce themselves as a consequence of good strings reproducing themselves. This "survival of the fittest" feature represents both a strength and a weakness. Advocates of genetic algorithms point to selection occurring on many more schemata than there are strings in the population, a phenomenon referred to as *implicit parallelism.* Critics point to the ease with which GAs can be deceived.

In an intriguing line of inquiry Goldberg has applied the idea of the schema theorem to the Walsh coefficients to arrive at operator-adjusted Walsh coefficients. Broadly speaking, Goldberg's idea is that if the w_i's are adjusted to take into account the effects of crossover and mutation, then the string values can be recomputed using the new w_i's. The adjusted string values approximate the value a GA assigns to a string given that it might destroy the schemata that create the string's value. Letting $\delta(\boldsymbol{\phi})$ equal the maximum distance between 1's in ${}^i\boldsymbol{\phi}$, and $o({}^i\boldsymbol{\phi})$ equal the number of ${}^i\boldsymbol{\phi}_j$ that equal 1, Goldberg defines the operator-adjusted Walsh coefficients as

$$
{}^{\mathrm{adj}}w_i = w_i \cdot \left[1 - \left(\frac{p_c \cdot \delta({}^i\boldsymbol{\phi})}{N-1}\right) - 2p_m \cdot o({}^i\boldsymbol{\phi})\right] \tag{d11}
$$

We will make a slight correction in Goldberg's estimate. First, borrowing from Tanese [8], we will say that a string $\mathbf{s}$ and a WF are *in parity* if ${}^i\boldsymbol{\phi}(\mathbf{s}) = 1$ and *out of parity* if ${}^i\boldsymbol{\phi}(\mathbf{s}) = -1$. Recall from (d8) that, when computing string values using the WFs, a Walsh coefficient is added if the WF and the string are in parity and subtracted if not. For ${}^i\boldsymbol{\phi}$, let p_{cr} be the probability that an arbitrary string switches parity after the crossover, and p_{mu} be the probability that an arbitrary string switches parity after mutation. The probability that the string retains the same parity after both operators is given by

$$
\begin{aligned}
P^{\mathrm{par}} &= 1 - p_{\mathrm{cr}} \cdot (1 - p_{\mathrm{mu}}) - (1 - p_{\mathrm{cr}}) \cdot p_{\mathrm{mu}} \\
&= 1 - p_{\mathrm{cr}} - p_{\mathrm{mu}} + 2 \cdot p_{\mathrm{cr}} \cdot p_{\mathrm{mu}}
\end{aligned} \tag{d12}
$$

Following the crossover and mutation operators, the expected contribution from ${}^i\boldsymbol{\phi}$ to a string initially in parity with it equals

$$
{}^E w_i = w_i \cdot [P^{\mathrm{par}} - (1 - P^{\mathrm{par}})] = w_i \cdot [2 \cdot P^{\mathrm{par}} - 1] \tag{d13}
$$

Computing the expected contribution from ${}^i\boldsymbol{\phi}$ reduces to calculating p_{cr} and p_{mu}. The following lemma will prove useful for calculating the former.

Lemma 2.A. ${}^i\boldsymbol{\phi}(\mathbf{s}) = {}^i\boldsymbol{\phi}(\mathbf{s}')$ *iff* ${}^i\boldsymbol{\phi} \cdot (\mathbf{s} - \mathbf{s}') = 2 \cdot k$ *for* $k \in \{0, \pm 1, \ldots\}$

Proof. Suppose ${}^i\boldsymbol{\phi}(\mathbf{s}) = {}^i\boldsymbol{\phi}(\mathbf{s}')$. Wolog assume ${}^i\boldsymbol{\phi}(\mathbf{s}) = 1$. From the definition, ${}^i\boldsymbol{\phi}(\mathbf{s}) = 2 \cdot k_1$ and ${}^i\boldsymbol{\phi}(\mathbf{s}') = 2 \cdot k_2$. Subtracting yields the result. The other direction is proved similarly. ■

Let p be the proportion of ith bits that equal 1 in the GA population. If two strings cross on the ith bit, the probability that their ith bits disagree approximately equals $p^2 + (1 - p)^2$, which is bounded above by 1/2. By Lemma 2.A parity changes if and only if the string switches bit values on an odd number of defined bits of the schema. Therefore, the following holds:

$$
p_{\mathrm{cr}} \le (1/2) \cdot p_c \cdot \left[\frac{\delta({}^i\boldsymbol{\phi})}{(N-1)}\right] \tag{d14}
$$

The probability that a string remains in parity with ${}^i\phi$ after mutation can be approximated and bounded by (d15) provided that the mutation rate is small:

$$p_{\text{mu}} \leq p_m \cdot o({}^i\phi) \tag{d15}$$

If both p_{cr} and p_{mu} are less than 1/2, then taking derivatives of (d12) shows that overestimating p_{cr} and p_{mu} underestimates P^{par}. Finally, (d13) shows that ${}^E w_i$ will be underestimated. We can now prove the following claim.

Claim 2. *An upper bound on the expected contribution from ${}^i\phi$ to a string initially in parity with it is given by*

$${}^E w_i \leq w_i \cdot \left[1 - \left(\frac{p_c \cdot \delta({}^i\phi)}{N-1}\right)\right] \cdot [1 - 2 \cdot p_m \cdot o({}^i\phi)]$$

Proof. Follows from above. ∎

Goldberg errs in calculating the effect of the operators jointly. He adds the effects of the operators when they should be multiplied. If parity is switched by both crossover and mutation, then parity has reverted back to the original parity. Therefore, our reformulation shows that the WFs are less susceptible to the operators than Goldberg had characterized. The following example shows how Goldberg's operator-adjusted Walsh coefficient differs from our upper bound on expected contribution.

Example. $N = 30$, $\delta(\phi) = 29$, $o(\phi) = 29$, $p_c = 0.6$, and $p_m = 0.01$. Then

$$\begin{aligned} {}^{\text{adj}}w_i &= w_i \cdot [1 - 0.6 - 0.58] &&= w_i \cdot [-0.18] \\ {}^E w_i &\leq w_i \cdot [1 - (0.6)] \cdot (0.58) &&= w_i \cdot [0.168] \end{aligned}$$

4. Variance and Walsh functions

In this section we compute schema variance with the Walsh coefficients. Applying schema variance will be the topic of the next section. Claim 3 calculates population variance using the Walsh coefficients. Population variance can also be interpreted as the variance of the trivial schema. The notation $\{\mathbf{f}\}$ denotes that we are viewing the vector $\mathbf{f}$ as a set with S members.

Claim 3. *The variance of the string values is given by*

$$\text{Var}(\{\mathbf{f}\}) = \sum_{i=1}^{S-1} (w_i)^2$$

Proof. From (d7),

$$f(\mathbf{s}) = \sum_{i=0}^{N-1} w_i \cdot {}^i\phi(\mathbf{s})$$

Recall that w_0 is the population mean; the contribution to the total variance from string $\mathbf{s}$ is given by

$$v(\mathbf{s}) = \left[\sum_{i=1}^{S-1} w_i \cdot {}^i\boldsymbol{\phi}(\mathbf{s})\right]^2 \Big/ S$$

Each term in [ ]'s can be written as the sum of the squares of two w_i's plus an interactive term of the form $\pm 2 \cdot w_i \cdot w_j$. From Lemma 0.B, for any ${}^i\boldsymbol{\phi}$ and ${}^j\boldsymbol{\phi}$ the set $\{\mathbf{s} : {}^i\boldsymbol{\phi}(\mathbf{s}) = {}^j\boldsymbol{\phi}(\mathbf{s})\}$ has cardinality $S/2$. If ${}^i\boldsymbol{\phi}(\mathbf{s}) = {}^j\boldsymbol{\phi}(\mathbf{s})$ the interactive term equals $+2 \cdot w_i \cdot w_j$, otherwise it equals $-2 \cdot w_i \cdot w_j$. The result follows immediately. ∎

This result requires modification for extension to variance of arbitrary schemata. Some WFs have the same parity relationship for all strings in the schema.

Example. Walsh functions ${}^5\phi$ (000101) and ${}^1\phi$ (000001) have the same parity for all strings lying in the schema 0**0**. WFs ${}^7\phi$ (000111) and ${}^5\boldsymbol{\phi}$, on the other hand, have opposite parity on exactly half of the strings and the same parity on half of the strings in 0**0**. It follows that the interactive term $+2w_1 \cdot w_5$ appears in the variance of 0**0**, whereas $w_7 \cdot w_5$ does not.

Renumbering the ${}^i\boldsymbol{\phi}$'s relative to the schema $\mathbf{h}$ simplifies the analysis. Two WFs are assigned the same number if they are identical on the non-$*$ bits of the schema $\mathbf{h}$:

$$C(\mathbf{h}) = \{j : z_j = *\} \tag{d17}$$

$$\hat{}\mathbf{h}(i) = \sum_{j \in C(\mathbf{h})} {}^i\boldsymbol{\phi}_j \cdot 2^{(j-1)} \tag{d18}$$

Note that $\hat{}\mathbf{h}(i) = 0$ is precisely the set $d(\mathbf{h})$:

Define $\mathbf{h}$ such that $\mathbf{h}$: $S \times S \rightarrow \{0, 1\}$ according to (d19)

$$\mathbf{h}(i,j) = 1 \quad \text{if } \hat{}\mathbf{h}(i) = \hat{}\mathbf{h}(j) > 0$$
$$\mathbf{h}(i,j) = 0 \quad \text{else}$$

Example. If $\mathbf{h} = **11**$ then $\hat{}\mathbf{h}(63) = \hat{}\mathbf{h}(51) = 51$ and $\mathbf{h}(51, 63) = 1$, where ${}^{63}\boldsymbol{\phi} = 111111$ and ${}^{51}\boldsymbol{\phi} = 110011$.

Lemma 4.A. *The interactive term* $\pm 2 \cdot w_i \cdot w_j$ *occurs in* $\mathrm{Var}(\{\mathbf{h}\})$ *if and only if* $\mathbf{h}(i,j) = 1$.

Proof. Case 1: $\hat{}\mathbf{h}(i) = 0$. It follows from (d18) that w_i appears in the average value of $\mathbf{h}$, so w_i cannot appear in the variance.

Case 2: $\hat{}\mathbf{h}(i) > 0$ and $\mathbf{h}(i,j) = 0$. Choose $k \in C(\mathbf{h})$ such that ${}^j\boldsymbol{\phi}_k = (1 - {}^i\boldsymbol{\phi}_k)$. Wolog assume that ${}^i\boldsymbol{\phi}_k = 1$. Let ${}^1\mathbf{h}$ be the schema with ${}^1\mathbf{h}_k = 1$ and ${}^1\mathbf{h}_i = *$ for all other i. Let ${}^0\mathbf{h}$ be the schema with ${}^0\mathbf{h}_k = 0$ and ${}^0\mathbf{h}_i = *$ for all other i. Define four sets

$$\{\mathbf{s} : \mathbf{s} \in \mathbf{h} \cap {}^1\mathbf{h} \text{ and } {}^i\boldsymbol{\phi}(\mathbf{s}) = {}^j\boldsymbol{\phi}(\mathbf{s})\} = B_{+1}$$
$$\{\mathbf{s} : \mathbf{s} \in \mathbf{h} \cap {}^1\mathbf{h} \text{ and } {}^i\boldsymbol{\phi}(\mathbf{s}) = -{}^j\boldsymbol{\phi}(\mathbf{s})\} = B_{-1}$$
$$\{\mathbf{s} : \mathbf{s} \in \mathbf{h} \cap {}^0\mathbf{h} \text{ and } {}^i\boldsymbol{\phi}(\mathbf{s}) = {}^j\boldsymbol{\phi}(\mathbf{s})\} = B_{+0}$$
$$\{\mathbf{s} : \mathbf{s} \in \mathbf{h} \cap {}^0\mathbf{h} \text{ and } {}^i\boldsymbol{\phi}(\mathbf{s}) = -{}^j\boldsymbol{\phi}(\mathbf{s})\} = B_{-0}$$

For every string $\mathbf{s} \in B_{+1}$, switching the value of the kth bit gives a string in B_{-0}, and vice versa. Similarly, for every string in B_{-1}, switching the value of the kth bit gives a string in B_{+0}, and vice versa. Therefore, exactly half of the strings in $\mathbf{h}$ have the same parity with ${}^j\phi$ and ${}^i\phi$.

Case 3: $\mathbf{h}(i,j) = 1$. For $k \in C(\mathbf{h})\, {}^j\phi_k = {}^i\phi_k$ and for some $k' \in C(\mathbf{h})$, ${}^j\phi_{k'} = 1$. From (d5), if there exists an $\mathbf{s} \in \mathbf{h}$ such that ${}^i\phi(\mathbf{s}) = {}^j\phi(\mathbf{s})$, then ${}^i\phi(\mathbf{s}') = {}^j\phi(\mathbf{s}')$ for all $\mathbf{s}' \in \mathbf{h}$. Therefore, the interactive term $+2w_i \cdot w_j$ appears in Var($\mathbf{h}$). Alternatively, if there exists an $\mathbf{s} \in \mathbf{h}$ such that ${}^i\phi(\mathbf{s}) = -{}^j\phi(\mathbf{s})$, then ${}^i\phi(\mathbf{s}') = -{}^j\phi(\mathbf{s}')$ for all $\mathbf{s}' \in \mathbf{h}$, and $-2w_i \cdot w_j$ appears in Var($\{\mathbf{h}\}$). ∎

Schema variance can be now be written in terms of the w_i's.

Claim 4. *The variance over a schema* $\mathbf{h}$ *is given by*

$$Var(\{\mathbf{h}\}) = \sum_{i=1}^{S-1}\sum_{i=1}^{S-1}[{}^i\phi(\mathbf{h}) \cdot {}^j\phi(\mathbf{h}) \cdot \mathbf{h}(i,j) \cdot w_i \cdot w_j]$$

Proof. Follows immediately from Claim 1, which showed that the mean of strings in $\mathbf{h}$ is given by summing over w_i where $i \in d(\mathbf{h})$, and from Lemma 4.A, which showed that the interactive term will appear iff $\mathbf{h}(i,j) = 1$.

5. Schema variance and deception

From the schema theorem we know that if two schemata $\mathbf{h}$ and $\mathbf{h}'$ satisfy $C(\mathbf{h}) = C(\mathbf{h}')$, then in the next generation the population will drift toward the schema with the higher average value. However, as the following example shows, looking two generations ahead this result need not hold.

Example. For $\mathbf{s} \in 0{*}{*}{*}{*}{*}$, let $f(\mathbf{s}) = 10$. For $\mathbf{s} \in 1{*}{*}{*}{*}{*}$, let $f(\mathbf{s}) = 18$ if $\mathbf{s}$ has an even number of bits equal to 1 and $f(\mathbf{s}) = 0$ if $\mathbf{s}$ has an odd number of bits equal to 1. In expected value for the initial population, $0{*}{*}{*}{*}{*}$ should have higher average value. The strings reproduced from $1{*}{*}{*}{*}{*}$ will all have value 18, so in generation two the drift should be toward the schema $1{*}{*}{*}{*}{*}$ provided the crossover and mutation rates are sufficiently small.

Accordingly, theoretical results pertaining to genetic algorithm performance should take into account not only the schemata means but also their distributions.

Claim 5 below shows that for a minimal, completely deceptive problem the average variance of the schemata with one and two defined bits containing the optimal string is greater than the average variance of the schemata that do not. In a minimally deceptive problem the schemata with higher average values do not contain the optimal string. Such problems can, but need not, be difficult for GAs. The inequivalence of GA deceptive and GA hard supports the argument that GAs implicitly take schemata distributions as well as means into account. Claim 5 can be interpreted as a partial explanation of this inequivalence. We follow Goldberg [3, 4] in defining a minimal, completely deceptive problem of length three. The basic idea is that, for schemata of order one and two defined on the same bits, 0's are better than 1's, but the string 111 has the highest value. For example, the schema 0** has higher average value than 1**; *00 has higher average value than *01; and *10 in turn has higher average value than *11. The requirement for schemata of order one is characterized by C1 below. C2 and C3 are necessary conditions for the schemata of order two. Finally, C4 and C5 are optimality conditions; they guarantee that 111 has the highest value.

A *minimal, completely deceptive problem* [3, 4] satisfies:

(C1) $w_1 > 0$, $w_2 > 0$, $w_4 > 0$

(C2) At least two of w_3, w_5, and w_6 are greater than 0

(C3) $w_3 + w_6 > w_1 + w_4$, $w_3 + w_5 > w_2 + w_4$, $w_5 + w_6 > w_1 + w_2$

(C4) $-w_7 > w_1 + w_2 + w_4$

(C5) $w_1 + w_3 > 0$, $w_2 + w_3 > 0$, $w_1 + w_5 > 0$, $w_4 + w_5 > 0$, $w_2 + w_6 > 0$, $w_4 + w_6 > 0$

Claim 5. *For a minimal, completely deceptive problem the following hold:*

(1) $\mathrm{Var}(1{*}{*}) + \mathrm{Var}({*}1{*}) + \mathrm{Var}({*}{*}1) > \mathrm{Var}(0{*}{*}) + \mathrm{Var}({*}0{*}) + \mathrm{Var}({*}{*}0)$

(2) $\mathrm{Var}(11{*}) + \mathrm{Var}({*}11) + \mathrm{Var}(1{*}1) > \mathrm{Var}(00{*}) + \mathrm{Var}({*}00) + \mathrm{Var}(0{*}0)$

Proof. For (1), the following can be derived from Claim 4:

$$\begin{aligned}
\mathrm{Var}(1{*}{*}) &= (w_6 - w_2)^2 + (w_5 - w_1)^2 + (w_7 - w_3)^2 \\
\mathrm{Var}({*}1{*}) &= (w_6 - w_4)^2 + (w_7 - w_5)^2 + (w_3 - w_1)^2 \\
\mathrm{Var}({*}{*}1) &= (w_7 - w_6)^2 + (w_5 - w_4)^2 + (w_3 - w_2)^2 \\
\mathrm{Var}(0{*}{*}) &= (w_6 + w_2)^2 + (w_5 + w_1)^2 + (w_7 + w_3)^2 \\
\mathrm{Var}({*}0{*}) &= (w_6 + w_4)^2 + (w_7 + w_5)^2 + (w_3 + w_1)^2 \\
\mathrm{Var}({*}{*}0) &= (w_7 + w_6)^2 + (w_5 + w_4)^2 + (w_3 + w_2)^2
\end{aligned}$$

The inequality reduces to

$$\begin{aligned}
&w_7(w_3 + w_5 + w_6) + w_4(w_5 + w_6) + w_2(w_3 + w_6) \\
&\quad + w_1(w_3 + w_5) < 0
\end{aligned} \qquad (*)$$

Using (C4) this would follow from

$$\begin{aligned}
&w_4(w_5 + w_6) + w_2(w_3 + w_6) + w_1(w_3 + w_5) \\
&\quad < (w_1 + w_2 + w_4) \cdot (w_3 + w_5 + w_6)
\end{aligned}$$

which reduces to $0 < w_3w_4 + w_5w_2 + w_6w_1$.

From (C1) and (C2), if all three of w_3, w_5, and w_6 are greater than 0, then the claim is true. Wolog suppose that $w_3 < 0$. By (C3) and $w_3 < 0$, it follows that $w_6 > w_4$. By (C5), $w_1 > -w_3$. These last two inequalities together imply that $-w_3w_4 < w_6w_1$, which proves (1).

From Claim 4, (2) reduces to

$$2 \cdot w_7(w_3 + w_5 + w_6) + w_4(w_5 + w_6) + w_2(w_3 + w_6) + w_1(w_3 + w_5) < 0$$

which follows from $w_7 < 0$, $(w_3 + w_5 + w_6) > (w_1 + w_2 + w_4)$, and $(*)$ above. ■

6. Conclusions

Theoretical justification for genetic algorithm performance relies on the schema theorem. Recent research on deception has attempted to explore the weaknesses of GA search. A function is characterized as deceptive if the small-order schemata lead the search away from the optimal point. Our research represents an alternative vantage point from which to gauge GAs and other search techniques. We believe that other statistical measures on schemata, such as variance, deserve consideration and might ultimately lead to greater understanding at the level of string value. The schema theorem says that above-average schemata will predominate. High-valued strings are generally members of schemata with above-average value. An alternative belief is that those characteristics shared by the best sampled strings will predominate, a subtle but important distinction. A complete theory of the role of variance must recognize that high-valued strings are reproduced and that schemata are a mere by-product. Such an understanding yields the result that high variance and average mean for a schema in generation t will generally be correlated with high mean for the schema in successive generations. A more formal presentation of this argument might include a two-generation schema theorem that gives the number of strings in schema h in generation $t + 2$ as a function of the number in generation t. Such a theory should also include Davidor's [2] two types of variance: sample variance (parasitic epistasis) and nonlinearity (base epistasis).

As noted earlier, Weitzman's [9] result that sources' means are less important than their tail distributions when searching for an optimal point should be included in the theory of GA performance. The logical first step toward inclusion is to consider both schema mean and schema variance. Whether genetic algorithms properly weigh the relative benefits of variance and mean in their search for optimal points is an open and exciting problem.

Acknowledgments

The authors would like to thank Stan Reiter and an anonymous referee for their comments and suggestions.

References

[1] A. D. Bethke, "Genetic Algorithms as Function Optimizers" (Doctoral Dissertation, University of Michigan), *Dissertation Abstracts International*, **41**(9) (1981) 3503B (University Microfilms No. 8106101).

[2] Y. Davidor, "Epistasis Variance: Suitability of a Representation to Genetic Algorithms," *Complex Systems*, **4** (1990) 369–383.

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

[4] D. E. Goldberg, "Genetic Algorithms and Walsh Functions: Part II, Deception and Its Analysis," *Complex Systems*, **3** (1989) 153–171.

[5] J. J. Greffenstette and J. E. Baker, "How Genetic Algorithms Work: A Critical Look at Implicit Parallelism," pages 20–27 in *Proceedings of the Third International Conference on Genetic Algorithms*, edited by H. Schaffer (San Mateo, Morgan Kaufmann, 1989).

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

[7] G. E. Liepins and M. D. Vose, "Polynomials, Basis Sets, and Deceptiveness in Genetic Algorithms," *Complex Systems*, **5** (1991) 45–61.

[8] R. Tanese, "Distributed Genetic Algorithms for Function Optimization" (Doctoral Dissertation, University of Michigan, 1989).

[9] M. L. Weitzman, "Optimal Search for the Best Alternative," *Econometrica*, **47** (1979) 641–654.