# Genetic Algorithms and the Variance of Fitness

*David E. Goldberg*
*University of Illinois at Urbana-Champaign*
*&*
*Mike Rudnick*
*Oregon Graduate Institute*

Oregon Graduate Institute
Department of Computer Science
and Engineering
19600 N.W. von Neumann Drive
Beaverton, OR 97006-1999 USA

Technical Report No. CS/E 90-011

December, 1990

# Genetic Algorithms and the Variance of Fitness

**David E. Goldberg**
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801-2996
goldberg@vmd.cso.uiuc.edu

**Mike Rudnick**
Department of Computer Science and Engineering
Oregon Graduate Institute
Beaverton, OR 97006-1999
rudnick@cse.ogi.edu

**Abstract**

This paper presents a method for calculating the variance of schema fitness using Walsh transforms. The computation is important for understanding the performance of genetic algorithms (GAs), because most GAs depend on the sampling of schema fitness in populations of modest size, and the variance of schema fitness is a primary source of noise that can prevent proper evaluation of building blocks, thereby causing convergence to other-than-global optima. The paper also applies these calculations to the sizing of GA populations and to the adjustment of the schema theorem to account for fitness variance; the extension of the variance computation to nonuniform populations is also considered. Taken together, this paper may be viewed as a step along the road to rigorous convergence proofs for recombinative genetic algorithms.

## 1 Introduction

It is well known that genetic algorithms (GAs) work best when building blocks—short, low-order schemata containing the optimum or desired near-optimum—are expected to grow, thereby permitting crossover to generate the desired solution or solutions. The schema theorem (Goldberg, 1989a; Holland, 1975) is widely and rightly recognized as the cornerstone of GA theory that has something to say about whether building blocks are at all likely to grow. It is less widely recognized that the schema theorem in its present form is only a result in *expectation* and does not guarantee that a building block will grow, even when the theorem's inequality is satisfied. In the usual small-population GA, stochastic effects can cause the algorithm to stray from the trajectory of the mean (De Jong, 1975; Goldberg & Segrest, 1987), and surprisingly few studies have considered these effects or allowed for their existence in the design of genetic algorithms.

In this paper, we consider one important source of stochastic variation, the variance of a schema's fitness or what we call *collateral noise.* Specifically, a method for calculating fitness variance from a function's Walsh transform is derived and applied to a number of problems in GA analysis.

In the remainder, Walsh functions and their application to the calculation of schema average fitness are reviewed; a formula for the calculation of schema fitness variance is derived using

Walsh transforms. The variance computation is then applied to two important problems in genetic algorithm theory: population sizing and the calculation of rigorous probabilistic convergence bounds. Extending the technique to the analysis of nonuniform populations is also discussed.

## 2 Review of Walsh-Schema Analysis

Walsh functions simplify calculations of schema average fitness, as was first pointed out by Bethke (1981). Using the notation developed elsewhere (Goldberg, 1989b), we consider fitness functions[1] $f$ mapping $l$-bit strings into the reals: $f : \{0,1\}^l \rightarrow R$. Bit strings are denoted by the symbol $\mathbf{x}$, which is also used to refer to the integer associated with the bit string, and individual bits may be referenced with an appropriate subscript; highest order bits are assumed to be leftmost: $\mathbf{x} = x_l \ldots x_2 x_1$. A schema is denoted by the symbol $\mathbf{h}$, which refers both to the schema itself (the similarity subset or that subset of strings with similarity at specified positions) or its string representation (the similarity template or that $l$-position string drawn from the alphabet $\{0,1,*\}$, where a 0 matches a 0, a 1 matches a 1, and a $*$ matches either).

There are a number of different ways of ordering and interpreting Walsh functions, but for this study we may most easily think of the Walsh functions, $\psi_j(\mathbf{x})$, $j = 0, \ldots, 2^l - 1$, as a set of $2^l$ partial parity functions, each returning a $-1$ or a $+1$ as the number of ones in its argument is odd or even over the set of bit positions defined by the ones in the binary representation of its index $j$. For example, consider the bit strings of length $l = 3$. For $j = 6 = 110_2$, the associated Walsh function considers the parity of strings at bits 2 and 3 (the bits that are set in the binary representation of the Walsh function index). Thus, $\psi_6(100) = \psi_6(011) = -1$ and $\psi_6(110) = \psi_6(001) = +1$. The manipulations are straightforward, but it is remarkable that the usual table lookup used to define functions over binary strings (one function value, one string) may be replaced by a linear combination of the Walsh functions: $f(\mathbf{x}) = \sum_{j=0}^{2^l-1} w_j \psi_j(\mathbf{x})$.

It is, perhaps, more remarkable that schema average fitness may be written directly as a partial Walsh sum (Bethke, 1981). An intuitive proof is given in Goldberg (1989b), but the main result states that the expected fitness of a schema may be calculated as follows:

$$f(\mathbf{h}) = \sum_{j \in J(\mathbf{h})} w_j \psi_j(\mathbf{h}), \tag{2.1}$$

where the argument of each Walsh function, $\mathbf{h}$, is interpreted as a string by replacing $*$'s with 0's and where the index set $J(\mathbf{h})$ is itself a similarity subset ($J : \{0,1,*\}^l \rightarrow \{0,1\}^l$) created by replacing $*$'s by 0's and fixed positions (1's and 0's) by $*$'s:

$$J_i(\mathbf{h}) = \begin{cases} 0, & \text{if } \mathbf{h}_i = *; \\ *, & \text{if } \mathbf{h}_i = 0, 1. \end{cases} \tag{2.2}$$

In words, the index set contains those terms that "make up" the schema in the sense that associated Walsh functions determine parity within the fixed positions of the schema. Thus, we see that a schema's average fitness may be calculated as a partial, signed sum of the Walsh coefficients specified by its index set, with the sign determined by the parity of the schema at the positions appropriate to the particular Walsh term.

To make this concrete, we return to three-bit examples. The expected fitness of the schema $*1*$ may be written as $f(*1*) = w_0 - w_2$, because the index set $J(*1*) = \;\; 0*0 \;\; = \{0,2\}$ and

[1] We adopt the standard GA practice of calling any non-negative figure of merit a fitness function, even though doing so is not necessarily consistent with biological usage of the term.

because the parity of any schema over no fixed positions ($\psi_0$) is even and because the parity of $*1*$ is odd over the position associated with $\psi_2$ (the middle position, $2 = 010_2$). Likewise, the expected fitness of schema $*10$ may be written as follows:

$$f(*10) = w_0 + w_1 - w_2 - w_3.$$

Here the index set generator is $J(*10) = 0** = \{0, 1, 2, 3\}$ and the associated signs may be determined by evaluating the associated Walsh functions using the schema. Continuing on to consider the fitness of a string, the expected fitness of schema (string) 110 may be written as

$$f(110) = w_0 + w_1 - w_2 - w_3 - w_4 - w_5 + w_6 + w_7,$$

because $J(110) = ***$, which dictates (as it must) a full Walsh sum for the string.

Note that as schemata become more refined—as they become more specific—their fitness sums include more Walsh terms. This contrasts starkly with using the table-lookup basis, where fitness average computations for specific schemata contain few terms and general schemata contain many. If we are to understand the relationships among low-order schemata and how they lead toward (or lead away from) optimal points, the Walsh basis is clearly the more convenient. Because of this, and because of the orthogonality of the Walsh basis, we use the Walsh-schema calculation to calculate the variance of schema fitness in the next section.

# 3 Computing Fitness Variance

The expected fitness of a schema is an important quantity, because it indicates whether, in a particular problem, a GA may be able to find optimal or near-optimal points through the recombination of building blocks. On the other hand, because most GAs depend upon statistical sampling, knowing schema average fitness is not enough; we must also consider the statistical variation of fitness to determine the amount of sampling required to accept or reject a building block with respect to one of its competitors. This requires that we also calculate the *variance* of schema fitness or what we call *collateral noise.*[2]

## 3.1 Variance from Walsh transforms

A real, discrete random variable $X$ may be viewed as an ordered pair $X = (S, p)$, where the variable takes a value chosen from a finite subset $S$ of the reals according to the probability density function $p(x)$. Variance is defined as the expected squared difference between a random variable and its mean:

$$\mathrm{var}(X) \quad = \quad \sum_{x \in S} p(x)(x - \overline{x})^2, \tag{3.3}$$

[2] Note that collateral noise arises in the context of deterministic fitness functions, because most genetic algorithms attempt to evaluate substrings (schemata) in the context of a full and varying whole (a full string) through limited statistical sampling. An experimentalist with such sloppy technique would never be sure of any of his conclusions, and it is for this reason that a strikingly different type of GA, a so-called *messy genetic algorithm* or mGA (Goldberg, Deb, & Korb, 1990; Goldberg, Korb, & Deb, 1989), seeks to sidestep collateral noise by evaluating substrings in the context of a temporarily invariant *competitive template*, a locally optimal string obtained by a messy GA run at a lower level. This technique appears to have wide applicability, but the variance calculations of this paper are important to messy GAs, because the issue of collateral noise cannot be sidestepped once recombination (the juxtapositional phase of an mGA) is invoked.

where $\overline{x}$ denotes the expected value of $x$. From this definition and assuming a uniform, full population, it is easy to show that the variance of a schema **h**'s fitness may be calculated as

$$\mathrm{var}(f(\mathbf{h})) = \frac{1}{|\mathbf{h}|}\sum_{\mathbf{x}\in\mathbf{h}}[f(\mathbf{x}) - \overline{f(\mathbf{h})}]^2. \tag{3.4}$$

Expanding and simplifying yields

$$\mathrm{var}(f(\mathbf{h})) = \overline{f^2(\mathbf{h})} - \overline{f(\mathbf{h})}^2. \tag{3.5}$$

The notation $\overline{f^2(\mathbf{h})}$ indicates that the expectation of $f^2$ is calculated, and the notation $\overline{f(\mathbf{h})}^2$ indicates that the expected value of $f$ is squared; in both cases, the argument **h** indicates that the expectation operation ranges over the elements of the schema. Using the Walsh-schema transform presented in the previous section, we derive equations for $\overline{f(\mathbf{h})}^2$ and $\overline{f^2(\mathbf{h})}$ separately, thereafter substituting each expression into equation 3.5.

Squaring the expression for schema average fitness yields

$$\begin{aligned}\overline{f(\mathbf{h})}^2 &= \left[\sum_{j\in J(\mathbf{h})} w_j\psi_j(\mathbf{h})\right]^2; \\ &= \sum_{j,k\in J(\mathbf{h})} w_j w_k \psi_j(\mathbf{h})\psi_k(\mathbf{h}).\end{aligned} \tag{3.6}$$

The quantity $\psi_j(\mathbf{x})\psi_k(\mathbf{x})$ is sometimes called the two-dimensional Walsh function $\psi_{j,k}(\mathbf{x})$. Straightforward arguments (Goldberg, 1989b) may be used to show that $\psi_{j,k}(\mathbf{x}) = \psi_{j\oplus k}(\mathbf{x})$, where $\oplus$ denotes bitwise addition modulo 2. Thus,

$$\overline{f(\mathbf{h})}^2 = \sum_{j,k\in J(\mathbf{h})} w_j w_k \psi_{j\oplus k}(\mathbf{h}). \tag{3.7}$$

Counting the number of quadratic terms is enlightening. There are $|J(\mathbf{h})|^2 = 2^{2o(\mathbf{h})}$ possibly non-zero terms in the indicated sum. It is interesting that this number is never more than the number of terms in $\overline{f^2(\mathbf{h})}$, as we shall soon see.

To derive an equation for $\overline{f^2(\mathbf{h})}$ in terms of the Walsh coefficients, start with the definition

$$\overline{f^2(\mathbf{h})} = \frac{1}{|\mathbf{h}|}\sum_{\mathbf{x}\in\mathbf{h}} f^2(\mathbf{x}), \tag{3.8}$$

and substitute the full Walsh expansion for $f(\mathbf{x})$,

$$\overline{f^2(\mathbf{h})} = \frac{1}{|\mathbf{h}|}\sum_{\mathbf{x}\in\mathbf{h}}\left(\sum_{j=0}^{2^l-1} w_j\psi_j(\mathbf{x})\right)^2. \tag{3.9}$$

Expanding, changing the order of summation, and recognizing the 2-D Walsh function, we obtain

$$\overline{f^2(\mathbf{h})} = \frac{1}{|\mathbf{h}|}\sum_{j=0}^{2^l-1}\sum_{k=0}^{2^l-1} w_j w_k \sum_{\mathbf{x}\in\mathbf{h}} \psi_{j\oplus k}(\mathbf{x}). \tag{3.10}$$

Further progress may be made by considering the summation

$$S(\mathbf{h}, j, k) = \sum_{\mathbf{x}\in\mathbf{h}} \psi_{j\oplus k}(\mathbf{x}), \tag{3.11}$$

which is virtually identical to the analogous summation $S(\mathbf{h}, j)$ in the Walsh-schema transform derivation (Goldberg, 1989b). As in the earlier derivation, each term of equation 3.11 is +1 or −1 since each is a Walsh function. Moreover, appealing to the earlier result, each sum is exactly $+|\mathbf{h}|$, $-|\mathbf{h}|$, or zero, the non-zero terms occurring when $j \oplus k \in J(\mathbf{h})$ and the associated sign determined by $\psi_{j\oplus k}(\mathbf{h})$. Thus, equation 3.10 may be rewritten as

$$\overline{f^2(\mathbf{h})} \quad = \sum_{j\oplus k \in J(\mathbf{h})} w_j w_k \psi_{j\oplus k}(\mathbf{h}). \tag{3.12}$$

Counting the number of terms in this sum is also enlightening. Thinking of the terms as being arrayed in a matrix with the $j$ index naming rows and the $k$ index naming columns, if we fix a row (if we fix $j$) there are at most $|J(\mathbf{h})|$ non-zero terms in the row. Each row has the same number of terms, because addition modulo 2 can do no more than translate each term to another position. Since there are $2^l$ rows, there are a total of $2^l|J(\mathbf{h})| = 2^{l+o(\mathbf{h})}$ possibly non-zero terms. This is never less than the number of terms in the $\overline{f(\mathbf{h})}^2$ sum. Actually the relationship between the two sums is much closer than this, as we shall soon see.

Finally, equation 3.5 may be rewritten using equations 3.12 and 3.7, producing

$$\mathrm{var}(f(\mathbf{h})) \quad = \sum_{(j,k)\in J^2_\oplus(\mathbf{h})} w_j w_k \psi_{j\oplus k}(\mathbf{h}) \quad - \sum_{(j,k)\in J^2(\mathbf{h})} w_j w_k \psi_{j\oplus k}(\mathbf{h}). \tag{3.13}$$

where $J^2(\mathbf{h}) = J(\mathbf{h}) \times J(\mathbf{h})$ and $J^2_\oplus = \{(j,k) : j \oplus k \in J(\mathbf{h})\}$. Noting the two summations have the same form, it is easy to show that the second sum is taken over a subset of the terms in the first. Remembering that $J(\mathbf{h})$ is a schema with $*$'s replacing the fixed positions of $\mathbf{h}$ and zeroes replacing the $*$'s, it is immediately clear that for any $(j,k) \in J^2(\mathbf{h})$ that $j \oplus k \in J(\mathbf{h})$. Thus, the terms in the second sum are a subset of those in the first. Therefore,

$$\mathrm{var}(f(\mathbf{h})) \quad = \sum_{(j,k)\ \in\ J^2_\oplus(\mathbf{h})-J^2(\mathbf{h})} w_j w_k \psi_{j\oplus k}(\mathbf{h}), \tag{3.14}$$

where the minus sign in the summation index denotes the usual set difference. In effect, we have converted a difference of summations to a sum over a difference of index sets.

The calculation is straightforward and not open to question, but counting the number of possibly non-zero terms is useful once again. The total number of non-zero terms in the overall sum is $2^{o(\mathbf{h})+l} - 2^{2o(\mathbf{h})} = 2^{o(\mathbf{h})}\left(2^l - 2^{o(\mathbf{h})}\right)$. Of course when the schemata are strings (when $o(\mathbf{h}) = l$), the sum vanishes as it must because the fitness function is deterministic. At other times, it is interesting that the Walsh sum potentially requires more computation than a direct calculation of variance using the table-lookup basis. Of course, we can always calculate fitness variance directly using the table-lookup basis if it is more convenient, but the insight gained by understanding the relationship between partitions is worth the price of admission. To better understand the structure of variance, we next consider the change in fitness variance that occurs as a fairly general schema is made more specific by fixing one or more of its free bits.

## 3.2 Changes in variance

We examine changes in variance by first considering the variance in fitness of the most general schema—by considering the variance of the function mean. Using equation 3.14 with $l = 3$, we obtain that the variance of $f(***)$ is simply $\mathrm{var}(f(***)) = w_1^2 + w_2^2 + w_3^2 + w_4^2 + w_5^2 + w_6^2 + w_7^2$, because $J(***) = \{0\}$ and $j \oplus j = 0$. In general, the variance of the function mean is the full sum of the squared Walsh coefficients less the squared $w_0$ term. The reasons for this are straightforward enough: orthogonality of the basis insures that all cross product terms drop out and subtraction of the square of the function mean simply deletes the term $w_0^2$.

Now consider the variance of a more specific schema, for example $**0$:

$$\mathrm{var}(**0) = w_2^2 + w_3^2 + w_4^2 + w_5^2 + w_6^2 + w_7^2 + 2\,(w_2w_3 + w_4w_5 + w_6w_7)\,.$$

Taking the difference between the fitness variance of $**0$ and that of $***$ yields:

$$\Delta\mathrm{var}(**0) = -w_1^2 + 2\,(w_2w_3 + w_4w_5 + w_6w_7)\,. \tag{3.15}$$

Note that the change in fitness variance comes from two sources:

1. removal of a diagonal (squared) term;

2. addition (or deletion) of off-diagonal (cross-product) terms.

These sources of variance change are the only ones that occur generally, as we now show by considering the change to the Walsh sum of a function when a bit is fixed.

Another study (Goldberg, 1989b) made the point that the Walsh functions may be thought of as polynomials if each bit, $x_i \in \{0,1\}$, is mapped to an auxiliary variable, $y_i \in \{-1,1\}$, where 0 maps to 1 and 1 maps to -1 (a linear mapping). Each Walsh function may be thought of as a monomial term, where the $y_i$'s included in the product are those with 1s in the binary representation of the Walsh function index. For example, $\psi_1(\mathbf{x}) = y_1$, $\psi_5(\mathbf{x}) = y_3y_1$, and $\psi_7(\mathbf{x}) = y_3y_2y_1$, where the change from $x_i$ to $y_i$ is understood. This way of thinking about the Walsh functions makes it easy to consider changes in variance and why they occur. To simplify matters further, we examine the different types of change in variance separately: the removal of diagonal terms and the addition or deletion of off-diagonal terms.

To isolate the change in variance due to the removal of diagonal terms, consider a function $f(\mathbf{x}) = w_0 + w_1\psi_1(\mathbf{x})$ only. The function's mean has variance $w_1^2$. When the first bit is fixed, we note an interesting thing: $f(**0) = w_0 + w_1\psi_1(**0) = w_0 + w_1y_1 = w_0 + w_1$. Fixing bit 1 fixes the associated Walsh function fully, causing the once-linear coefficient to become a constant. Since constant terms in a Walsh sum do not contribute to the variance, in evaluating the change of fitness, $w_1^2$ must be removed. Viewed in this way, it is not surprising that the same reasoning applies to any on-diagonal term whose associated Walsh function becomes fixed by the schema under consideration.

To isolate the change in variance that occurs through the addition or deletion of off-diagonal terms, consider a function $f(\mathbf{x}) = w_0+w_2\psi_2(\mathbf{x})+w_3\psi_3(\mathbf{x})$ only. The function mean has variance $w_2^2 + w_3^2$. When bit one is set to 0, we note an interesting thing:

$$\begin{aligned} f(**0) &= w_0 + w_2\psi_2(**0) + w_3\psi_3(**0); \\ &= w_0 + w_2y_2 + w_3y_2y_1 = w_0 + w_2y_2 + w_3y_2; \\ &= w_0 + (w_2 + w_3)\psi_2(\mathbf{x}),\ \mathbf{x} \in **0. \end{aligned}$$

In words, the fixed position of the schema fixes a bit in the once-quadratic $\psi_3$. The new linear term has variance $(w_2 + w_3)^2 = w_2^2 + 2w_2w_3 + w_3^2$, but the two squared terms in this sum are already contained in the original computation for the lower order schema. Thus, the change in variance as a result of the fixing is $2w_2w_3$. Depending on whether the bit is set to a 0 or a 1, this change in variance can be positive or negative. For example, if we had considered the schema $**1$, the change in fitness would have been negative, because $(w_2 - w_3)^2 = w_2^2 - 2w_2w_3 + w_3^2$

(as before, the squared terms are already accounted in the more general schema). It should also be noted that what bit fixing can giveth, bit fixing can taketh away. In the example, the fixing of the second bit after the rightmost bit has already been fixed will cause the previously added, off-diagonal variance term to be removed. This type of mechanism is analogous to that discussed above in connection with the removal of on-diagonal terms.

Although we simplified matters to isolate the different types of variance change, the correct variance given by equation 3.14 may be thought of as taking the variance of the mean and simply removing all diagonal terms whose Walsh functions are fixed by the schema, adding all off-diagonal terms that result when a schema transmutes a high-degree Walsh term to one of lower degree, and removing all previously added off-diagonal terms that become fully fixed. This view may followed fairly easily level by level as was done elsewhere (Goldberg, 1989b) in connection with schema averages by using an approximate variance $\hat{\mathrm{var}}^{(k)}$ at level $k$ and considering the difference in variance $\Delta\mathrm{var}^{(k)}$ at each level. The calculations are straightforward and are not pursued here. Instead, we consider some examples of variance calculations.

## 3.3 Example: linear functions

The variance of linear functions and their schemata is easy to calculate using the Walsh basis. Since the function is linear, the only non-zero terms in the summation are those associated with the constant and order-1 Walsh functions. Thus,

$$f(\mathbf{x}) = w_0 + \sum_{j:o(j)=1} w_j \psi_j(\mathbf{x}). \tag{3.16}$$

As a result, the fitness variance of a linear function is the sum of the squared linear terms, and schema fitness variance is simply the sum of the squared linear terms whose associated Walsh functions are not fixed by the fixed positions of the schema. Here, the change in variance with progressively more specific schemata is all of the diagonal-removing variety, because the lack of nonlinearity precludes the transmutation of a high-degree monomial into one of lower degree as bits are fixed.

An illustrative numerical example can be generated by considering the function $f(u) = u$, where $u$ is a 3-bit unsigned integer, $u(\mathbf{x}) = \sum_{i=1}^{3} 2^{i-1}x_i$, $x_i \in \{0,1\}$. Calculating the Walsh transform of $f$ (Goldberg, 1989b), we obtain $w_0 = 3.5$, $w_1 = -0.5$, $w_2 = -1.0$, $w_4 = -2.0$, and $w_i = 0$ otherwise. The variance values are tabulated for all schemata in table 1, where the shorthand notation of an "f" is used to denote a fixed position in the schema.

Besides illustrating the simple structure of schema fitness variance in linear functions, the tabulation may be used to make an important point about GA convergence. It has often been observed that high bits in binary-coded GAs converge much sooner than low bits. The table helps explain why this is so. Scanning the table, we see that the high-bit schemata have lower variance than the others. Thinking of the square root of the fitness variance as the amount of noise faced by a schema when it is sampled in a randomly chosen population, we see that high-bit schemata exist in a less noisy environment than their low-bit cousins. Moreover, the signal difference between competing high-bit schemata is also higher than that of low-bit schemata. The double whammy of higher signal difference and lower noise forces the higher bits to convergence faster. Explicit and rigorous accounting of the signal-difference-to-noise ratio will be necessary in a moment when we calculate the population size necessary for low error rates in the presence of collateral noise. Before considering this, however, we examine whether specific schemata are always less noisy than their more general forebears.

Table 1: Variance Tabulation for a Linear 3-bit Function

| Schema | Symbolic | Numeric |
|---|---|---|
| *** | $w_1^2 + w_2^2 + w_4^2$ | 5.25 |
| **f | $w_2^2 + w_4^2$ | 5.00 |
| *f* | $w_1^2 + w_4^2$ | 4.25 |
| f** | $w_1^2 + w_2^2$ | 1.25 |
| *ff | $w_4^2$ | 4.00 |
| f*f | $w_2^2$ | 1.00 |
| ff* | $w_1^2$ | 0.25 |
| fff | 0 | 0.00 |

### 3.4 Example: refinement does not imply variance reduction

In linear functions, fixing one or more bits means that a more specific schema will have lower fitness variance than one in which that schema is properly contained. In nonlinear functions, this need not be the case. Here, we construct an example of a simple function where fixing a bit increases the variance.

Doing so is straightforward. Setting the expression for $\Delta\text{var}(**0)$ greater than zero yields:

$$2(w_2w_3 + w_4w_6 + w_5w_7) > w_1^2.$$

Similar inequalities may be derived for the other zero-containing, order-1 schemata. Setting all Walsh coefficients of identical order $i$ equal to $w_i'$, each of these inequalities has the same form: $2(2w_1'w_2' + w_2'w_3') > {w_1'}^2$. Choosing not to use the third-order coefficient (setting $w_3' = 0$) yields $w_2' > w_1'/4$. Thus, we have created a function that varies more at order 1 (with a zero set) than at order 0. It is interesting that the order-1 schemata with ones set are less noisy than their competitors (and less noisy than the most general schema). This is so, because the signs on the cross-product terms are negative. In general, it is also interesting that if the variance values for all competing schemata over a particular partition are summed, the cross-product terms drop out, because each competing schema has the same terms and half the signs are positive and half are negative. Although refinement need not lead to variance reduction for an individual schema, it does insure non-increasing partition variance.

These examples lead us to consider more general applications and extensions of the variance calculation in the next section.

## 4 Applications and Extensions

In this section, we consider two applications of the variance calculations: population sizing and a collateral-noise adjustment to the schema theorem. Additionally, we consider the extension of the Walsh-variance computation to nonuniform populations.

### 4.1 Population sizing in the presence of collateral noise

A previous study (Goldberg, 1989d) considered population sizing from the standpoint of schema turnover rate; that study knowingly ignored variance and its effects, but explicitly identified stochastic variation as a possibly important factor in determining appropriate population size.

Table 2: One-sided Normal Deviates $z$ and $c = z^2$ Values at Different Levels of Significance $\alpha$

| $\alpha$ | $z$ | $c$ |
|---|---|---|
| 0.1 | 1.28 | 1.64 |
| 0.05 | 1.65 | 2.71 |
| 0.01 | 2.33 | 5.43 |
| 0.005 | 2.58 | 6.66 |
| 0.001 | 3.09 | 9.55 |

Here we atone for that previous, albeit conscious, omission by considering a simple, yet rational, sizing formula that accounts for collateral noise.

We start by assuming that the function is linear or approximately linear and that all order-1 terms in the Walsh expansion are equal to $w_1'$. We consider all pairwise comparisons of competing $k$-bit schemata and choose a population size so the probability that the sample mean fitness of the best schema is less than the sample mean fitness of the second best schema is less than some specified value, $\alpha$. Posed in this way, we have a straightforward problem in decision theory.

Assuming that all variance is due to collateral noise (i.e., assuming that operator variance is small with respect to that of the function), and assuming that population sizes are large enough so the central limit theorem applies, the variance of the sample mean fitness of a single, order-$k$ schema is

$$\text{var}(\hat{f}(\mathbf{h})) = \frac{(l-k){w_1'}^2}{n/2^k}, \tag{4.17}$$

where the hat is used to denote the sample mean and $n$ is the population size. The numerator results from the Walsh-variance computation, and the denominator assumes that the schema is represented by its expected number of copies in the sample population. The sample mean fitness of the best and second best schemata have the same variance; the variance of the difference between their values is twice that amount. Taking the square root we obtain the standard deviation of the difference in sample mean fitness values as

$$\sigma = \sqrt{\frac{(l-k)2^{k+1}{w_1'}^2}{n}}. \tag{4.18}$$

Calculating the unit random normal deviate for the difference in sample mean fitness values, $z$, we obtain the following:

$$z = \frac{2w_1'}{\sigma} \tag{4.19}$$

Squaring $z$ and rearranging yields an expression for the population size:

$$n = c(l-k)2^{k-1}, \tag{4.20}$$

where $c = z^2$ and $z$ is chosen to make the probability that the difference between the sample mean fitness of the best and second best schemata is negative as small as desired. Values of $z$ and $c$ for different levels of significance $\alpha$ are shown in table 2. For example, considering $k = 1$ at a significance level of 0.1, the population sizing formula becomes $n = 1.64(l-1)$. Many problems are run with strings of length 30 to 100, from which the formula would suggest population sizes in the range 49 to 164. This range is not inconsistent with standard suggestions for population size (De Jong, 1975) that have been derived from empirical tests. Similar reasoning may be used to derive formulas for population sizing if the building blocks are scaled nonuniformly or if the function is nonlinear. Instead of pursuing these refinements, we consider collateral noise adjustments to the schema theorem.

## 4.2 Variance adjustments to the schema theorem

The schema theorem is a lower bound on the expected propagation of building blocks in subsequent generations, but it is important to keep in mind that it is only a result in *expectation* and does not bound the actual performance of any GA. By explicitly recognizing the importance of variance, however, we can calculate a lower bound that, to some specified level of significance, does account for the potential stochastic variations caused by collateral noise. Here we consider selection only, and even then, limit the adjustment we make to one for collateral noise, but the technique can be generalized to include variance adjustments for the selection mechanism itself and other operators.

For proportionate reproduction acting alone the schema theorem may be written: (Goldberg, 1989a)

$$\overline{m(\mathbf{h}, t+1)} = m(\mathbf{h}, t)\frac{f(\mathbf{h}, t)}{f(\Omega, t)}, \tag{4.21}$$

where $t$ is the generation number, $m(\mathbf{h}, t)$ is the number of representatives of a schema in the current generation, $f(\mathbf{h}, t)$ is the average fitness of the schema $\mathbf{h}$ in the current population, $f(\Omega, t)$ is the average fitness of the current population, and the overbar is the expectation operator as before. To adjust for the variance of the fitness function, we assume that we are consistently unlucky in both the numerator and the denominator:

$$m(\mathbf{h}, t+1) \geq m(\mathbf{h}, t)\frac{f(\mathbf{h}, t) - z\sigma(f(\mathbf{h}, t))/\sqrt{m(\mathbf{h}, t)}}{f(\Omega, t) + z\sigma(f(\Omega, t))/\sqrt{n}}, \tag{4.22}$$

where the expectation operation (the bar) has been dropped, $z$ is the critical value of a one-sided normal test of significance at some specified level, $\sigma$ denotes the standard deviation of the specified quantity ($\sigma(x) = \sqrt{\mathrm{var}(x)}$), and $n$ is the population size. In this way, we have conservatively assumed that the fitness of the schema will be extraordinarily low, and the average fitness will be extraordinarily high (to some level of significance). If the population is sized properly, the desired schema will still grow when $m(\mathbf{h}, t+1)/m(\mathbf{h}, t) > 1$. Note that, strictly speaking, these computations require that we calculate variance over the nonuniformly distributed population that exists at generation $t$. We will outline that computation in a moment, but the variance computation for a uniform population should give a useful estimate. Moreover, although we have only made adjustments for collateral noise here, it is clear that further adjustments can and should be made to the schema theorem to include all additional stochastic variations:

1. true function noise (nondeterministic $f$s);
2. variance in the selection algorithm (aside from collateral noise);
3. variance from expected disruption rates due to crossover, mutation, and other genetic operators.

Any such adjustments should be conservative and assume that mean performance is worse than expected by an amount $z$ times the standard deviation of that operator acting alone. If all adjustments are made, then the resulting inequality will be a proper bound at a calculable level of significance. In other words, satisfaction of such a variance-adjusted schema theorem will assure that the probability that an advantageous schema loses proportion in some generation is below some specified amount. When done properly, these calculations should lead to rigorous convergence proofs for genetic algorithms.

### 4.3 Nonuniform populations

The calculation of the previous section assumed that the variance of a particular schema's fitness is well represented by the uniform, full population value. In a nonuniformly distributed population, a more accurate value can be obtained by calculating the variance of a schema's fitness directly.

Defining a proportion-weighted fitness value $\phi(\mathbf{x}) = f(\mathbf{x})P(\mathbf{x})2^l$ as in Bridges and Goldberg (1989), the calculation of variance proceeds immediately if we recognize that we must multiply two different Walsh expansions, one for $f$ (the usual transform, $w_i$) and one for $\phi$ (the transform of proportion-weighted fitness, call it $w_i''$). The mathematics follows exactly as in section 3, except that the terms of $\bar{f}^2$ involve only products of the $w_i''$ terms, while the terms of $\overline{f^2}$ involve products of $w_i$ and $w_j''$ terms. As a result the overall sum does not collapse to a single sum over a difference of index sets. Nonetheless, the structure of the terms present in the two sums (the ordered pairs in $J_{\oplus}^2$ and $J^2$) is the same as before, because the index sets are the same.

## 5 Conclusions

This paper has presented, interpreted, applied, and extended a method for calculating schema fitness variance using Walsh transforms. For some time, genetic algorithmists have been content to use results in expectation such as the schema theorem. Serious efforts at rigorous convergence proofs for recombinative GAs demand that we consider variance of the function, variance of the operators, and other sources of stochasticity. Some of these issues have been tackled here, and a rigorous approach to the others has been outlined, but it is apparent that this line of inquiry clears a first path to fully rigorous GA convergence theorems for populations of modest size.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants CTS-8451610 and ECS-9022007 and by U.S Army Contract DASG60-90-C-0153. The second author acknowledges departmental support from the Oregon Graduate Institute. Finally, we thank Clay Bridges, Kalyanmoy Deb, and Rob Smith for useful discussions related to this work.

## References

Bethke, A. D. (1981). Genetic algorithms as function optimizers (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International, 41*(9), 3503B. (University Microfilms No. 8106101)

Bridges, C. L., & Goldberg, D. E. (1989). *A note on the non-uniform Walsh-schema transform* (TCGA Report No. 89004). Tuscaloosa: The University of Alabama, The Clearinghouse for Genetic Algorithms.

De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International, 36*(10), 5140B. (University Microfilms No. 76-9381)

Goldberg, D. E. (1989a). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

Goldberg, D. E. (1989b). Genetic algorithms and Walsh functions: Part I, a gentle introduction. *Complex Systems, 3*, 129–152.

Goldberg, D. E. (1989c). Genetic algorithms and Walsh functions: Part II, deception and its analysis. *Complex Systems, 3*, 153–171.

Goldberg, D. E. (1989d). Sizing populations for serial and parallel genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, 70–79.

Goldberg, D. E., Deb, K., & Korb, B. (1990). Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems, 4*, 415-444.

Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems, 3*, 493-530.

Goldberg, D. E., & Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, 41–49.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.