Form Invariance and Implicit Parallelism

Michael D. Vose

Alden H. Wright
University of Montana - Missoula, alden.wright@umontana.edu

Form Invariance and Implicit Parallelism

Michael D. Vose vose@cs.colostate.edu
Computer Science Department, 107 Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301, USA

Alden H. Wright wright@cs.umt.edu
Computer Science Department, University of Montana, Missoula, MT 59812-1008, USA

Abstract
Holland's schema theorem (an inequality) may be viewed as an attempt to understand genetic search in terms of a coarse graining of the state space. Stephens and Waelbroeck developed that perspective, sharpening the schema theorem to an equality. Of particular interest is a "form invariance" of their equations; the form is unchanged by the degree of coarse graining. This paper establishes a similar form invariance for the more general model of Vose et al. and uses the attendant machinery as a springboard for an interpretation and discussion of implicit parallelism.

Keywords
Course graining, form invariance, genetic algorithms, implicit parallelism, intrinsic parallelism, mixing scheme, schemata.

1 Introduction

Whereas Holland's schema theorem (Holland, 1975) has been widely appealed to in theoretical analysis of genetic algorithms, it has severe limitations which call into question the logical validity of arguments based upon it (Vose, 1993; Juliany and Vose, 1994). The first attempt to address shortcomings of the schema theorem was by Bridges and Goldberg (1987) who derived an exact expression for the expected next generation for a simple binary GA using proportional selection and one-point crossover. Vose (1990) extended Bridges and Goldberg's work (using an independent model) to include the effects of mutation with crossover and proportional selection.[1] Whitley (1993) and Stephens and Waelbroeck (1997) also obtain similar results for one-point crossover (but no mutation) and binary strings.

What set the work of Stephens et al. apart from results prior to 1997 was the "form invariance" of their equations. Schemata can be viewed as a means of coarse graining the state space, and irrespective of that coarse graining, the form of the evolution equation, as given by Stephens et al., remains the same. Coarse graining may be desirable as a means to produce a simplified model of reduced complexity in which many states have been collapsed or aggregated together. For example, population geneticists typically model higher-level organisms that may have on the order of 100,000 genes with n-locus models, for small n.

As explained by Vose (1998, 1999) and Vose and Rowe (2000), coarse graining (referred to by Ξ) of the representation space Λ is not without peril. Of particular significance is the consistency of the coarse-grained model, call it $\tilde{h}$, with respect to the

[1]This model has since been simplified and generalized to include arbitrary mutation types, crossover types, and selection schemes, as well as multicardinality alphabets (Koehler et al., 1997; Vose, 1999).

system it is modeling, call it h.[2] Without some analogue of the following commutative diagram,

$$\begin{tikzcd}
\Lambda \arrow[r, "h"] \arrow[d, "\Xi"'] & \Lambda \arrow[d, "\Xi"] \\
\Xi\Lambda \arrow[r, "\ddot{h}"] & \Xi\Lambda
\end{tikzcd}$$

there is no guarantee that predictions made by the coarse-grained model $\tilde{h}$ have any significance to the original system h. Vose (1999) has shown that the coarse graining *must* be based on schemata if a consistent model of crossover is to be obtained (Vose, 1999, Theorem 17.13), and in that case, the following diagram commutes (Vose, 1999, Theorem 19.4)

$$\begin{tikzcd}
x \arrow[r] \arrow[d, "\Xi"'] & \mathcal{M}(x) \arrow[d, "\Xi"] \\
\Xi x \arrow[r, "\ddot{\mathcal{M}}"] & \Xi\mathcal{M}(x)
\end{tikzcd}$$

where $\mathcal{M}$ denotes the mixing scheme. Of particular interest is the fact that the schema-based model $\tilde{\mathcal{M}}$ *has the same functional form as* $\mathcal{M}$. Thus the model of Vose et al. has "form invariance" as well, and since that model includes both mutation and crossover (by way of arbitrary masks), it is a more general result than Stephens et al. (1997). The result, however, was proved for the binary case.

This paper extends the result above to arbitrary finite cardinality alphabets, generalizing schemata and schema families in terms of quotient rings along the way, and specializes conclusions about mixing to mutation and crossover separately. A further contribution is that computational issues are touched upon. Using the ancillary machinery of previous results, this paper concludes with a discussion and geometric interpretation of "implicit parallelism."

2 Notation

This section and the next two follow Vose (1999) and are included for completeness. The search space Ω is the set of length ℓ c-ary strings. Integers in the interval $[0, n)$, where $n = c^\ell$, are identified with the elements of Ω through their c-ary representations. Elements of Ω can be regarded as column vectors in $\mathcal{C}^\ell$ with the least significant c-ary digit as the topmost element (here $\mathcal{C}$ denotes the set of complex numbers). Indexing of vectors and matrices begins with 0.

The search space Ω can also be regarded as the product group

$$Z_c \times \ldots \times Z_c$$

where the group operation $\oplus$ is componentwise addition modulo c. Let $\ominus$ denote componentwise subtraction modulo c, and let $\otimes$ denote componentwise multiplication modulo c. With respect to $\oplus$ and $\otimes$, Ω is a commutative ring.

[2]Here it is system dynamics (i.e., how the system transforms state) that is being modeled; thus h and $\ddot{h}$ are functions.

An element of Ω is called *binary* if all its nonzero components are ones. Let $\mathbf{1}$ denote the binary vector of all ones, and let $\overline{u}$ denote $\mathbf{1} \ominus u$. Let $\#u$ denote the number of nonzero components in u. Thus $u^T v = \#(u \otimes v)$ when u and v are binary.

Given binary $k \in \Omega$, let $\Omega_k = \{j \in \Omega \;:\; j \otimes k = j\}$. Each Ω_k is an ideal (in the ring Ω) containing k. For example, if $c = 3$ and $\ell = 4$, then

$$\begin{aligned} \Omega_{10} &= \{0,1,2,9,10,11,18,19,20\} \\ &= \{0000,0001,0002,0100,0101,0102,0200,0201,0202\} \\ &= \{\langle 0000\rangle,\langle 1000\rangle,\langle 2000\rangle,\langle 0010\rangle,\langle 1010\rangle,\langle 2010\rangle,\langle 0020\rangle,\langle 1020\rangle,\langle 2020\rangle\}, \end{aligned}$$

where angle brackets $\langle \ldots \rangle$ denote a tuple that is regarded as a column vector.

Let $k = c^{l_0} + \cdots + c^{l_{m-1}}$, where $l_0 < \ldots < l_{m-1}$ are integers and $m = \#k$. The set $\{c^{l_0}, \ldots, c^{l_{m-1}}\}$ is a basis for the ideal Ω_k. Integers in the interval $[0, c^m)$ are identified with Ω_k through *the injection corresponding to* k, determined by the map $K : \{0, \ldots, c-1\}^m \longrightarrow \Omega_k$ given by

$$\langle \alpha_0, \ldots, \alpha_{m-1} \rangle \longmapsto \alpha_0 c^{l_0} + \cdots + \alpha_{l_{m-1}} c^{l_{m-1}}.$$

Embedding an integer of the interval $[0, c^m)$ via K amounts to distributing its c-ary digits among the locations where k is nonzero. For example, if $c = 3$, $\ell = 3$, and $k = 10$, then K determines the function

$$\{(0,\langle 000\rangle),(1,\langle 100\rangle),(2,\langle 200\rangle),(3,\langle 001\rangle),(4,\langle 101\rangle),$$

$$(5,\langle 201\rangle),(6,\langle 002\rangle),(7,\langle 102\rangle),(8,\langle 202\rangle)\}.$$

A schema is a subset of Ω where some string positions are specified (fixed) and some are unspecified (variable). Schemata have traditionally been denoted by pattern strings, where a special symbol such as $*$ is used to denote an unspecified bit (Holland used #). Thus, the schema denoted by $2{*}0{*}$ is the set of strings

$$\{2000,2001,2002,2100,2101,2102,2200,2201,2202\}.$$

Equivalently, a schema is a coset $\Omega_u \oplus v$ of the ideal Ω_u (where u is binary). Without loss of generality, $u \otimes v = 0$ since

$$\begin{aligned} \Omega_u \oplus v &= \Omega_u \oplus (\mathbf{1} \otimes v) \\ &= \Omega_u \oplus ((u \oplus \overline{u}) \otimes v) \\ &= \Omega_u \oplus (u \otimes v \oplus \overline{u} \otimes v) \\ &= (\Omega_u \oplus u \otimes v) \oplus \overline{u} \otimes v \\ &= \Omega_u \oplus (\overline{u} \otimes v). \end{aligned}$$

Given schema $\Omega_u \oplus v$, one may regard u as a mask for the variable positions and v as determining values for the fixed positions. For example, the schema $\Omega_{10} \oplus 54$ is the schema $2{*}0{*}$ displayed above.

A population is a multiset of cardinality r (the population size) of elements of Ω. A population is represented by a population vector p according to the rule

$$p_j = \text{the proportion in the population of } j.$$

Thus populations are represented by elements of the simplex

$$^{n}\Lambda = \{x \in \mathcal{C}^n : \sum x_i = 1 \text{ and } x_i \geq 0\}.$$

The simplex can also be interpreted as the space of probability distributions over Ω.

The *indicator function* $[\ldots]$ is defined by

$$[expr] = \begin{cases} 1 & \text{if } expr \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

3 Schema Families and Schema Representation

Given binary ξ, elements of the quotient ring $\Omega/\Omega_{\overline{\xi}}$ are a family of *competing schemata*; they have similar variable positions (indicated by nonzero positions of $\overline{\xi}$) but differ in their fixed positions (indicated by nonzero positions of ξ). The ring Ω_ξ is naturally isomorphic to the schema family $\Omega/\Omega_{\overline{\xi}}$ by the map

$$x \longmapsto \Omega_{\overline{\xi}} \oplus x.$$

The schema family is, therefore, said to be *represented by* ξ. Let $n_\xi = \text{card}(\Omega_\xi) = c^{\#\xi}$, and let $\Lambda_\xi = {}^{n_\xi}\Lambda$. The linear map

$$\Xi : \Lambda \longrightarrow \Lambda_\xi$$

with matrix

$$\Xi_{i,j} = [j \otimes \xi = i]$$

is called the *operator associated with the family.* Here columns are indexed by Ω (i.e., $0 \leq j < n$) and rows are indexed by Ω_ξ (which is interpreted via the injection corresponding to ξ to mean $0 \leq i < n_\xi$). Since Ω_ξ is isomorphic to $\Omega/\Omega_{\overline{\xi}}$, the operator Ξ maps a population vector p such that the components of Ξp are in one-to-one correspondence with the schema family represented by ξ. If $\xi = c^{l_0} + \cdots + c^{l_{m-1}}$, and $l_0 < \ldots < l_{m-1}$ are integers, this correspondence is given explicitly by

$$(\Xi p)_{\sum \alpha_k c^k} \longleftrightarrow \Omega_{\overline{\xi}} \oplus (\alpha_0 c^{l_0} + \cdots + \alpha_{l_{m-1}} c^{l_{m-1}}).$$

As the following computation shows, the i th component of Ξp is simply the proportion of the population (represented by p) contained in the schema corresponding, as above, to i. Let i be identified (via the injection corresponding to ξ) with $\sum \alpha_k c^{l_k}$. Then

$$\begin{aligned}
(\Xi p)_i &= \sum_j \Xi_{i,j}\, p_j \\
&= \sum_j [j \otimes \xi = \textstyle\sum \alpha_k c^{l_k}]\, p_j \\
&= \sum_j [j \otimes \xi = \textstyle\sum \alpha_k c^{l_k}]\, p_j \displaystyle\sum_{u \in \Omega_{\overline{\xi}}} [j \otimes \overline{\xi} = u] \\
&= \sum_{u \in \Omega_{\overline{\xi}}} \sum_j [j \otimes \xi = \textstyle\sum \alpha_k c^{l_k}][j \otimes \overline{\xi} = u]\, p_j \\
&= \sum_{u \in \Omega_{\overline{\xi}}} \sum_j [j = u \oplus \textstyle\sum \alpha_k c^{l_k}]\, p_j \\
&= \sum_{j \in \Omega_{\overline{\xi}} \oplus \sum \alpha_k c^{l_k}} p_j .
\end{aligned}$$

The previous equality can be expressed more succinctly as

$$(\Xi p)_i = \sum_{i' \in \Omega_{\overline{\xi}}} p_{i \oplus i'}\,.$$

As an example, let $c = 3$, $\ell = 2$, $\xi = 3$. Then $\overline{\xi} = 1$, $m = 1$, and Ξ is represented by the matrix

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

The family of competing schemata in pattern-string notation is $\Omega/\Omega_{\overline{\xi}} = \{0*, 1*, 2*\}$.

If $\xi = 1$, then Ξ is represented by the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The family of competing schemata in pattern-string notation is $\Omega/\Omega_{\overline{\xi}} = \{*0, *1, *2\}$.

4 The Simple Genetic Algorithm

Let p be a population vector. The *selection scheme* $\mathcal{F} : \Lambda \to \Lambda$ is determined by

$$\mathcal{F}(p)_i = \text{the probability that } i \text{ is selected for the gene pool.}$$

The *mixing matrix* M is determined by

$$M_{i,j} = \text{the probability that } 0 \text{ results from mutation and crossover applied to } i \text{ and } j.$$

If mutation is performed before crossover, then

$$M_{i,j} = \sum_{u,v,w} \mu_u \mu_v \frac{\chi_w + \chi_{\overline{w}}}{2} [(i \oplus u) \otimes w \oplus \overline{w} \otimes (j \oplus v) = 0],$$

and if mutation is performed after crossover, then

$$M_{i,j} = \sum_{u,v} \mu_u \frac{\chi_v + \chi_{\overline{v}}}{2} [i \otimes v \oplus \overline{v} \otimes j \oplus u = 0].$$

Here μ_m is the probability m is used as a mutation mask to perform the mutation

$$x \longmapsto x \oplus m,$$

and χ_m is the probability m is used as a crossover mask to perform the crossover

$$\langle x,\, y \rangle \longmapsto \langle x \otimes m \oplus y \otimes \overline{m},\, y \otimes m \oplus x \otimes \overline{m} \rangle.$$

The vectors μ and χ are called the *mutation distribution* and *crossover distribution*, respectively. In the zero mutation case (meaning that $i > 0 \Rightarrow \mu_i = 0$), both mixing matrices reduce to

$$M_{i,j} = \sum_{v} \frac{\chi_v + \chi_{\overline{v}}}{2} [i \otimes v \oplus \overline{v} \otimes j = 0],$$

and in the zero crossover case (meaning that $i > 0 \Rightarrow \chi_i = 0$), they reduce to

$$M_{i,j} = \frac{\mu_{0 \ominus i} + \mu_{0 \ominus j}}{2}.$$

The *mixing scheme* $\mathcal{M} : \Lambda \to \Lambda$ is determined by

$$\mathcal{M}(p)_i = \sum_{u,v} p_{u \oplus i} p_{v \oplus i} M_{u,v}$$

and reduces in the zero crossover case to

$$\mathcal{M}(p)_i = \sum_{u} p_{i \ominus u} \mu_u.$$

The *heuristic* of the simple genetic algorithm is $\mathcal{G} = \mathcal{M} \circ \mathcal{F}$ and satisfies

$$\mathcal{G}(p)_i = \text{the probability that } i \text{ is produced for the next generation.}$$

Let $\mathcal{M}_\chi$ be the mixing scheme for zero mutation; let $\mathcal{M}_\mu$ be the mixing scheme for zero crossover; let $\mathcal{M}_{\chi\mu}$ be the mixing scheme for mutation before crossover; and let $\mathcal{M}_{\mu\chi}$ be the mixing scheme for crossover before mutation. The following theorem follows immediately from the definitions by expanding right hand sides and simplifying.

THEOREM 4.1: *$\mathcal{M}_{\chi\mu} = \mathcal{M}_\chi \circ \mathcal{M}_\mu$ and $\mathcal{M}_{\mu\chi} = \mathcal{M}_\mu \circ \mathcal{M}_\chi$.*

5 Selection

As demonstrated in Vose (1999), there are problems with commutativity of the sort

$$\begin{tikzcd} x \arrow[r] \arrow[d, "\Xi"'] & \mathcal{F}(x) \arrow[d, "\Xi"] \\ \Xi x \arrow[r, "\ddot{\mathcal{F}}"] & \Xi\mathcal{F}(x) \end{tikzcd}$$

Nevertheless, if one was interested in modeling the state space by elements of the schema family represented by ξ (or by a collection of schema families represented by $\{\xi^0, \ldots, \xi^h\}$), then it might be of interest, given population p, to determine $\Xi\mathcal{F}(p)$. Using previous notation,

$$(\Xi\mathcal{F}(p))_i = \sum_{i' \in \Omega_{\overline{\xi}}} \mathcal{F}(p)_{i \oplus i'}.$$

There are at most r nonzero terms in the sum to consider, since general selection schemes satisfy

$$p_j = 0 \Longrightarrow \mathcal{F}(p)_j = 0.$$

Let $\mathcal{F}$ be proportional selection with fitness vector f (i.e., f_i = fitness of i) and define a "vector product" denoted by $x \cdot y$, as opposed to the *scalar* product $x^T y$, by $(x \cdot y)_i = x_i y_i$. Then

$$\mathcal{F}(p) = \frac{f \cdot p}{f^T p}.$$

In terms of this notation, the *utility* U_i of schema $i \oplus \Omega_{\overline{\xi}}$ relative to population p is classically defined as

$$U_i = \frac{f^T p \, (\Xi \mathcal{F}(p))_i}{(\Xi p)_i}$$

(as follows from expanding the right hand side). For example, if $c = 3$, $\ell = 2$, $\xi = 3$,

$$U = \left\langle \frac{f_0 \, p_0 + f_1 \, p_1 + f_2 \, p_2}{p_0 + p_1 + p_2}, \; \frac{f_3 \, p_3 + f_4 \, p_4 + f_5 \, p_5}{p_3 + p_4 + p_5}, \; \frac{f_6 \, p_6 + f_7 \, p_7 + f_8 \, p_8}{p_6 + p_7 + p_8}. \right\rangle$$

Expressed in vector form, this can be rearranged to yield

$$\Xi \mathcal{F}(p) = \frac{U \cdot (\Xi p)}{U^T (\Xi p)}.$$

Although this expression for $\Xi \mathcal{F}(p)$ is reminiscent of proportional selection (with the vector U of utilities playing the role of f and Ξp playing the role of p), form invariance does not follow. Form invariance refers to a situation where if one starts with a functional form and performs a coarse graining, then the result has the same functional form but in the new coarse-grained variables. Functional form is *not* preserved in the case of proportional selection because the left hand side above is $\Xi \mathcal{F}(p)$ rather than $\mathcal{F}(\Xi p)$. Moreover, attempting to force form invariance via defining a selection scheme $\mathcal{F}'$ by

$$\mathcal{F}'(\Xi p) = \frac{U \cdot (\Xi p)}{U^T (\Xi p)}$$

is really nothing more than notational sleight of hand; the utility vector U, however one may attempt to define it, is, in general, *not* capable of being well-defined in the equation above.[3]

One might think form invariance would hold if the fitness function were linear. Even this strong assumption, however, is insufficient to imply form invariance, as the following example shows. The following also clarifies, by way of a concrete example, the general abstract remarks made in the preceding paragraph.

Let $c = 3$ and $\ell = 2$. A linear fitness function is described by the fitness vector

$$f = \langle 0, \; b_0, \; 2b_0, \; b_1, \; b_0 + b_1, \; 2b_0 + b_1, \; 2b_1, \; b_0 + 2b_1, \; 2b_0 + 2b_1 \rangle,$$

where b_0 and b_1 are the fitness coefficients on the first and second ternary digits, respectively. If $\xi = 3$, then $\Xi \mathcal{F}(p)$ is

$$\frac{1}{f^T p} \left\langle b_0 (p_1 + 2p_2), \; b_0 (p_4 + 2p_5) + b_1 (p_3 + p_4 + p_5), \; b_0 (p_7 + 2p_8) + 2b_1 (p_6 + p_7 + p_8) \right\rangle .$$

If it were possible to define a selection scheme $\mathcal{F}'$ as above, then $\mathcal{F}'(\Xi p) = \Xi \mathcal{F}(p)$, which implies that the displayed vector above (i.e., $\Xi \mathcal{F}(p)$) could be written as a function of $\Xi p = \langle p_0 + p_1 + p_2, \, p_3 + p_4 + p_5, \, p_6 + p_7 + p_8 \rangle$, which is an obvious contradiction.

The reader is referred to Vose (1999) for an explanation and discussion of compatibility issues as they pertain, in particular, to the compatibility of selection with a coarse grained model based on schemata, and as they pertain, in general, to the use, design, and interpretation of approximate models.

[3] See Vose (1999) for a general discussion, but note in the context of this paper that the utility U as displayed above is defined with respect to p rather than with respect to Ξp; it has no definition in the coarse-grained model.

6 Mixing

Unlike selection, mixing *does* enjoy form invariance. Let $\mathcal{M}_\xi : \Lambda_\xi \to \Lambda_\xi$ be the mixing scheme corresponding to the search space $\Omega/\Omega_{\overline{\xi}}$ with mutation distribution $\Xi\mu$ and crossover distribution $\Xi\chi$.

THEOREM 6.1: *If mutation is performed after crossover, then*

$$\Xi\mathcal{M}(x) = \mathcal{M}_\xi(\Xi x).$$

PROOF: The kth component of $\Xi\mathcal{M}(x)$ is

$$\begin{aligned}\sum_{k'\in\Omega_{\overline{\xi}}} \mathcal{M}(x)_{k\oplus k'} &= \sum_{u,v\in\Omega_\xi}\sum_{u',v'\in\Omega_{\overline{\xi}}} x_{u\oplus u'}x_{v\oplus v'}\sum_{k'\in\Omega_{\overline{\xi}}} M_{u\oplus u'\ominus k\ominus k',v\oplus v'\ominus k\ominus k'}\\ &= \sum_{u,v\in\Omega_\xi}\sum_{u'\in\Omega_{\overline{\xi}}} x_{u\oplus k\oplus u'}\sum_{v'\in\Omega_{\overline{\xi}}} x_{v\oplus k\oplus v'}\sum_{k'\in\Omega_{\overline{\xi}}} M_{u\ominus k',v\ominus u'\oplus v'\ominus k'}.\end{aligned}$$

The innermost sum above is

$$\sum_{k'\in\Omega_{\overline{\xi}}}\sum_{i,j\in\Omega_\xi}\sum_{i',j'\in\Omega_{\overline{\xi}}} \mu_{i\oplus i'}\frac{\chi_{j\oplus j'}+\chi_{\overline{j\oplus j'}}}{2}$$
$$[(u\ominus k')\otimes(j\oplus j')\oplus(\overline{j\oplus j'})\otimes(v\ominus u'\oplus v'\ominus k')\oplus i\oplus i' = 0].$$

Note that the indicator function is equivalent to

$$[(0\ominus k')\otimes j'\oplus(\overline{\xi}\ominus j')\otimes(0\ominus u'\oplus v'\ominus k')\oplus i' = 0][u\otimes j\oplus(\xi\ominus j)\otimes v\oplus i = 0].$$

The first factor of this is equivalent to $[k' = (\overline{\xi}\ominus j')\otimes(v'\ominus u')\oplus i']$, which determines k'. It follows that the sum above is

$$\sum_{i,j\in\Omega_\xi}[u\otimes j\oplus(\xi\ominus j)\otimes v\oplus i = 0]\sum_{i'\in\Omega_{\overline{\xi}}}\mu_{i\oplus i'}\sum_{j'\in\Omega_{\overline{\xi}}}\frac{\chi_{j\oplus j'}+\chi_{\overline{j\oplus j'}}}{2} = (M_\xi)_{u,v}.$$

Therefore, the The kth component of $\Xi\mathcal{M}(x)$ is

$$\begin{aligned}\sum_{u,v\in\Omega_\xi}(M_\xi)_{u,v}\sum_{u'\in\Omega_{\overline{\xi}}} x_{u\oplus k\oplus u'}\sum_{v'\in\Omega_{\overline{\xi}}} x_{v\oplus k\oplus v'} &= \sum_{u,v\in\Omega_\xi}(M_\xi)_{u,v}(\Xi x)_{u\oplus k}(\Xi x)_{v\oplus k}\\ &= \mathcal{M}_\xi(\Xi x)_k.\end{aligned}$$

□

COROLLARY 6.2: *Whether or not mutation is performed after crossover,*

$$\Xi\mathcal{M}(x) = \mathcal{M}_\xi(\Xi x).$$

In particular,

$$\begin{aligned}\Xi\mathcal{M}_\mu(x) &= (\mathcal{M}_\mu)_\xi(\Xi x)\\ \Xi\mathcal{M}_\chi(x) &= (\mathcal{M}_\chi)_\xi(\Xi x).\end{aligned}$$

PROOF: Special cases of Theorem 6.1 are $\Xi\mathcal{M}_\mu(x) = (\mathcal{M}_\mu)_\xi(\Xi x)$ and $\Xi\mathcal{M}_\chi(x) = (\mathcal{M}_\chi)_\xi(\Xi x)$. Using these together with Theorem 4.1 gives

$$\begin{aligned}
\Xi\mathcal{M}_{\chi_\mu}(x) &= \Xi\mathcal{M}_\chi \circ \mathcal{M}_\mu(x) \\
&= (\mathcal{M}_\chi)_\xi(\Xi\mathcal{M}_\mu(x)) \\
&= (\mathcal{M}_\chi)_\xi((\mathcal{M}_\mu)_\xi(\Xi x)) \\
&= (\mathcal{M}_\chi)_\xi \circ (\mathcal{M}_\mu)_\xi(\Xi x) \\
&= (\mathcal{M}_{\chi_\mu})_\xi(\Xi x).
\end{aligned}$$

□

7 Computational Issues

As discussed in Vose and Wright (1998a, 1998b), calculation of $\mathcal{M}$ is more efficient by way of the Fourier basis. Since $\mathcal{M}_\xi$ is simply a "reduced cardinality" version of $\mathcal{M}$ (i.e., the dimension of Λ_ξ does not exceed the dimension of Λ), nothing further about this needs to be said.

It is of interest, however, to further clarify the relationship between the mutation and crossover distributions μ and χ and their counterparts $\Xi\mu$ and $\Xi\chi$. It is *not* generally true that if the crossover distribution χ corresponds to 1-point or 2-point crossover, then so does the crossover distribution $\Xi\chi$. When the crossover distribution is very sparse (as for 1-point and 2-point crossover), $\Xi\chi$ might be computed for string lengths approaching a million by summing over schemata. In other cases where that is infeasible, the components of $\Xi\chi$ must be determined analytically. The situation for mutation and uniform crossover is considerably nicer, as the following theorem shows.

THEOREM 7.1: *If the mutation distribution μ corresponds to a mutation rate, then so does the mutation distribution $\Xi\mu$. If the crossover distribution χ is uniform, then so is the crossover distribution $\Xi\chi$. Moreover, the mutation and crossover rates are unchanged.*

PROOF: Consider first mutation with rate ε. Using previous notation, $(\Xi\mu)_i$ is given by

$$\begin{aligned}
&\sum_{j\in\Omega_{\overline{\xi}}\oplus\sum\alpha_k c^{l_k}} (\varepsilon/(c-1))^{\#j}(1-\varepsilon)^{\ell-\#j} \\
&= (\varepsilon/(c-1))^{\sum[\alpha_k>0]}(1-\varepsilon)^{\#\xi-\sum[\alpha_k>0]} \sum_{j'\in\Omega_{\overline{\xi}}} (\varepsilon/(c-1))^{\#j'}(1-\varepsilon)^{\#\overline{\xi}-\#j'} \\
&= (\varepsilon/(c-1))^{\sum[\alpha_k>0]}(1-\varepsilon)^{\#\xi-\sum[\alpha_k>0]} \sum_{j=0}^{\#\overline{\xi}} \binom{\#\overline{\xi}}{j}(c-1)^j(\varepsilon/(c-1))^j(1-\varepsilon)^{\#\overline{\xi}-j} \\
&= (\varepsilon/(c-1))^{\#i}(1-\varepsilon)^{\#\xi-\#i}.
\end{aligned}$$

Next consider uniform crossover with crossover rate α.

$$\begin{aligned}
(\Xi\chi)_i &= \sum_{j\in\Omega_{\overline{\xi}}\oplus\sum\alpha_k c^{l_k}} \alpha c^{-\ell}[j>0] + (1-\alpha+\alpha c^{-\ell})[j=0] \\
&= [\textstyle\sum\alpha_k c^{l_k}>0]\displaystyle\sum_{j\in\Omega_{\overline{\xi}}}\alpha c^{-\ell} + [\textstyle\sum\alpha_k c^{l_k}=0](1-\alpha+\displaystyle\sum_{j\in\Omega_{\overline{\xi}}}\alpha c^{-\ell})
\end{aligned}$$

$$= \quad [i > 0]\alpha c^{\#\overline{\xi}-\ell} + [i = 0](1 - \alpha + \alpha c^{\#\overline{\xi}-\ell})$$

$$= \quad \alpha c^{-\#\xi}[i > 0] + (1 - \alpha + \alpha c^{-\#\xi})[i = 0].$$

□

8 Implicit Parallelism

The phrase "intrinsic parallelism" was used by Holland (1975) in connection with the fact that a single population member simultaneously belongs to a plethora of schemata. Therefore, a sequence of "trials" (i.e., a number of samples from Ω made according to $\mathcal{G}$) "is at the same time a sequence of trials for each of a large number of schemata." This phenomenon has a straightforward geometric interpretation as will be developed below. The reader is cautioned in advance that the notation used in this section may appear to conflict with that used previously.

8.1 Schemata

Expressing schemata in functional terms and interpreting the result geometrically clarifies the relationship between populations and schemata. Identify schemata with integers in the set $\{0, \ldots, m-1\}$ by regarding a schema as an integer written in base three; 0 represents zero, 1 represents one, $*$ represents two. For example,

$$0\,0\,1 * * 0\,0\,0\,0\,1\,1\,1 * * * * \; = \; 3,012,632.$$

Let $n = 2^\ell$, $m = 3^\ell$ and recall that

$${}^m\Lambda = \{\langle x_0, \ldots, x_{m-1}\rangle : x_i \geq 0, \sum x_i = 1\}.$$

Note that ${}^m\Lambda$ has one component for each schema. Define the linear map $\Xi : {}^n\Lambda \longrightarrow {}^m\Lambda$ by

$$\Xi_{i,j} = \begin{cases} 1 & \text{if element } j \text{ of } \Omega \text{ is contained in schema } i \\ 0 & \text{otherwise.} \end{cases}$$

Identifying schema i with the transpose of the i th row of Ξ, schemata are seen to be vectors in Euclidean space. Moreover, the components of Ξp are simply the inner products of these vectors with the population p.

This formalization of schemata, as vectors, is functional in the sense that any vector v can be interpreted as the function $v : x \mapsto v^T x$. Moreover, when a schema (i.e., vector) is interpreted in this way (as a function), it simply maps a population p to the proportion of p contained in the schema.

The previous paragraphs imply that schemata, collectively represented by the matrix Ξ, are simply directions in Euclidean space and therefore form an alternate coordinate system. To explain this clearly, the concept of a generalized coordinate system will first be presented. Before doing so, however, a few remarks will be made to make contact with previous sections.

Whereas the formalism above is clean and simple, it appears to conflict with previously used notation. Because low-level detail was important to results in previous sections, there schema were grouped into schema families, and Ξ was defined in association with a family. Here, however, the subject of concern is schemata in general, thus Ξ is defined in terms of all schemata. Note that Ξ is nevertheless the same type of object here as it was in previous sections; just as before,

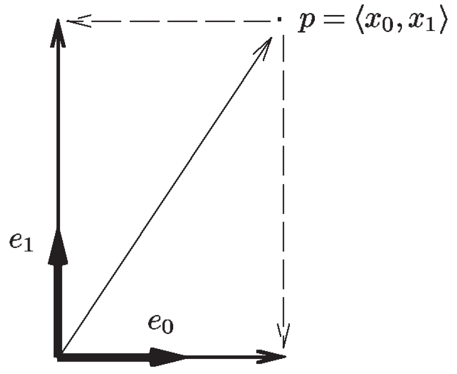

Figure 1: Standard coordinates.

- Ξ is a binary matrix.
- Rows of Ξ correspond to schemata.
- $\Xi_{i,j} = 1 \Longleftrightarrow j \in$ (the schema corresponding to) i.
- $(\Xi p)_i$ = the proportion of (the population represented by) p contained in (the schema corresponding to) i.

8.2 Generalized Coordinates

Consider a point p that we choose to coordinatize. In the two-dimensional diagram above (Figure 1), e_0 and e_1 are mutually orthogonal unit vectors that are used to determine the coordinates $\langle x_0, x_1 \rangle$ of the point p. The component of the vector from the origin to p in the direction of e_0 has magnitude x_0, and the component of the vector from the origin to p in the direction of e_1 has magnitude x_1. If these objects have algebraic form, say they are represented by 2×1 matrices, then the coordinates of p are given by

$$x_0 = e_0^T p$$
$$x_1 = e_1^T p$$

However, the vectors e_0 and e_1 need not have unit length, neither must they be mutually orthogonal, neither must there be only two of them to represent the two-dimensional situation above. All that is necessary to determine the two-dimensional point p from the "generalized coordinates" $\langle \ldots, e_i^T p, \ldots \rangle$ is that the matrix C with i th row e_i^T has left inverse C^{-1}:

$$Cp = \begin{pmatrix} \vdots \\ e_i^T p \\ \vdots \end{pmatrix} \Longrightarrow p = C^{-1} \begin{pmatrix} \vdots \\ e_i^T p \\ \vdots \end{pmatrix}$$

Moreover, the generalized coordinate $e_i^T p$ *represents* the component in the direction of e_i in any case; when e_i is not a unit vector, the quantity $e_i^T p$ is simply that component measured to a different scale (i.e., with respect to a different choice of unit).

In general, there are no requirements placed on the rows of C. Left multiplication of a column vector p by a matrix C simply computes the generalized coordinates of p

with respect to C, and those generalized coordinates are, for suitable choice of units, components in the directions of (the transpose of) the rows of C. If a left inverse C^{-1} does not exist, that simply indicates the generalized coordinates Cp contain insufficient information to recover p. This occurs exactly when the rows of C do not span, as would be the case if there were fewer rows than columns. If C has more rows than columns, they cannot be independent since row rank equals column rank. In that case, p has fewer coordinates than generalized coordinates, and some generalized coordinates therefore will be redundant.

Returning to schemata, $\tilde{p} = \Xi p$ is simply the vector of generalized coordinates of p expressed with respect to the vectors that represent schemata. To illustrate with a small example, take $\ell = 2$, so $n = 4$ and $m = 9$.

integer	schema	vector
0	00	$\langle \mathbf{1,0,0,0} \rangle$
1	01	$\langle \mathbf{0,1,0,0} \rangle$
2	0∗	$\langle 1,1,0,0 \rangle$
3	10	$\langle \mathbf{0,0,1,0} \rangle$
4	11	$\langle \mathbf{0,0,0,1} \rangle$
5	1∗	$\langle 0,0,1.1 \rangle$
6	∗0	$\langle 1,0,1,0 \rangle$
7	∗1	$\langle 0,1,0,1 \rangle$
8	∗∗	$\langle 1,1,1,1 \rangle$

Let, for example, $p = \langle .1, .2, .3, .4 \rangle$ then

$$\Xi p = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{.1} \\ \mathbf{.2} \\ \mathbf{.3} \\ \mathbf{.4} \end{pmatrix} = \begin{pmatrix} \mathbf{.1} \\ \mathbf{.2} \\ .3 \\ \mathbf{.3} \\ \mathbf{.4} \\ .7 \\ .4 \\ .6 \\ 1. \end{pmatrix}$$

Hence p expressed in generalized coordinates (with respect to the matrix Ξ) is

$$\tilde{p} = \langle \mathbf{.1}, \mathbf{.2}, .3, \mathbf{.3}, \mathbf{.4}, .7, .4, .6, 1. \rangle.$$

Letting Ξ^{-1} denote a left inverse of Ξ, the original components of p can be recovered from the generalized components via left multiplication by Ξ^{-1}

$$\Xi^{-1}\tilde{p} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{.1} \\ \mathbf{.2} \\ .3 \\ \mathbf{.3} \\ \mathbf{.4} \\ .7 \\ .4 \\ .6 \\ 1. \end{pmatrix} = \begin{pmatrix} \mathbf{.1} \\ \mathbf{.2} \\ \mathbf{.3} \\ \mathbf{.4} \end{pmatrix}$$

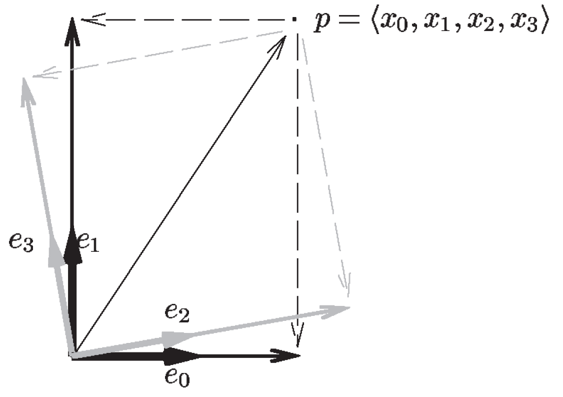

Figure 2: Generalized coordinates.

In the example above, bold font is used to emphasize how the original components of p are injected into the generalized components by way of the trivial schema – those schema that correspond exactly to elements of Ω. The other generalized components – those not in bold font (i.e., those corresponding to nontrivial schema) – are simply redundant coordinates of $\tilde{p}$.

It is to some extent difficult to visualize geometrically what is going on in the example because of the dimension involved. By way of analogy, Figure 2 (which is an augmentation of Figure 1 – superfluous vectors e_2 and e_3 have been added) may help clarify the situation. In the two-dimensional diagram above, the two-dimensional point p has four generalized coordinates x_0, x_1, x_2, x_3 with respect to the 4×2 matrix C having rows $e_0^T, e_1^T, e_2^T, e_3^T$. The component of the vector from the origin to p in the direction of e_0 has magnitude x_0; the component of the vector from the origin to p in the direction of e_1 has magnitude x_1; the component of the vector from the origin to p in the direction of e_2 has magnitude x_2; and the component of the vector from the origin to p in the direction of e_3 has magnitude x_3. Of course, the latter coordinates of the two-dimensional point p are superfluous; all coordinates beyond x_0, x_1 are redundant.

In the case of schemata, exponentially many superfluous coordinates beyond those corresponding to the standard basis vectors occur. They number

$$3^\ell - 2^\ell = 3^\ell(1 - (2/3)^\ell) \approx 3^\ell.$$

8.3 Processing Leverage

Another use of "intrinsic parallelism" by Holland (1975) was to describe the "tremendous power" crossover had by virtue of the fact that "each crossing-over affects great numbers of schemata." This has also a straightforward geometric interpretation.

In order to get at the heart of the matter, consider a two-dimensional example where an operator moves from point p in the state space to point q. In the two-dimensional diagram above (Figure 3), the operator might be said to *process* the information $\langle x_0, x_1 \rangle$ so as to produce the new state $\langle x'_0, x'_1 \rangle$. Here C is the matrix with rows e_0^T and e_1^T, and the generalized coordinates of p and q are $\langle x_0, x_1 \rangle$ and $\langle x'_0, x'_1 \rangle$, respectively; the processing is expressed with respect to C.

Next consider the *same* operator on the *same* two-dimensional state space, except that a large number of superfluous rows $e_2^T, \ldots, e_n^T$ are added to C so that the *same* two-dimensional points p and q have now generalized coordinates $\langle x_0, x_1, \ldots x_n \rangle$ and $\langle x'_0, x'_1, \ldots x'_n \rangle$, respectively, where, of course, the newly gained generalized coordinates

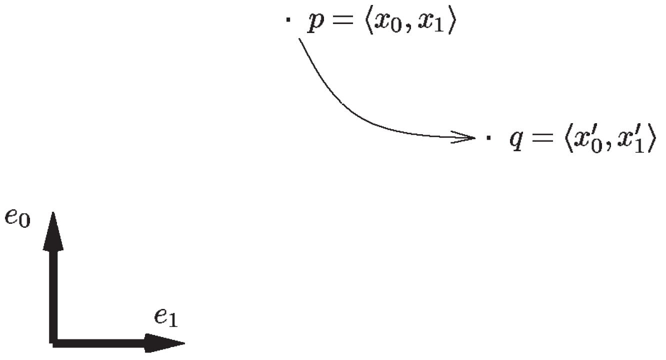
$p = \langle x_0, x_1 \rangle$
$q = \langle x'_0, x'_1 \rangle$
e_0
e_1

Figure 3: "Processing" coordinates.

$x_2, \ldots x_n$ and $x'_2, \ldots x'_n$ are redundant. If n is tremendously large, is it reasonable to assert that the operator has now somehow acquired "tremendous power" by virtue of the fact that it processes a great number of generalized components?

The situation for $\mathcal{M}$ is similar. When described in terms of elements of Ω, mixing simply processes strings, where the processing is expressed with respect to C = the identity matrix,

$$p \overset{\mathcal{M}}{\longmapsto} q.$$

If a large number of superfluous rows are added to enlarge the identity matrix to Ξ, then when expressed with respect to $C = \Xi$, it *is* true that crossover processes[4] a tremendous number of generalized components (i.e., schemata)

$$\Xi p \overset{\mathcal{M}}{\longmapsto} \Xi q.$$

However, all components beyond those corresponding to the identity matrix – all components corresponding to nontrivial schemata families – are *not* independently processed; they provide *no* additional information, and from a geometric point of view, they simply clutter the representation with redundant, useless components.

Holland (1975) asserts that intrinsic parallelism provides genetic algorithms with "critical advantages" over enumerative processes. Goldberg (1989) uses the term "implicit parallelism," as opposed to intrinsic parallelism, to describe "process leverage" arising from the fact that "genetic algorithms inherently process a large quantity of schemata while processing a relatively small quantity of strings."

Observe that some representations are more compact than others. A population p is a point of Λ, and because of its relatively small size – the population size r is typically much smaller than n – there are many zero terms in its representation with respect to the standard basis

$$p = \sum p_i e_i.$$

In fact, there are at most r nonzero terms. Therefore, by the convention that zero terms need not be mentioned, describing a population in terms of the elements it contains is fairly compact. If one introduces exponentially many redundant special directions (i.e., schemata) with corresponding exponentially many superfluous coordinates (i.e., proportions of p represented by schemata), then one can have exponentially many nonzero generalized components and one might declare: "genetic algorithms inherently process

[4]If "process" is interpreted in the benign sense as used in discussing Figure 3.

a large quantity of schemata while processing a relatively small quantity of strings." Okay, but so what?

The notion that "leverage" or "critical advantages" are gained by contemplating a large number of redundant coordinates is dubious at best. The widespread belief that genetic algorithms are robust by virtue of their schema processing is, in view of the observations made in this paper, more the result of salesmanship than logical analysis. In fact, the no free lunch theorem (Wolpert and Macready, 1995) shows that genetic algorithms are on average *worse* than enumeration.[5]

Nevertheless, schema can be said to be processed by a genetic algorithm (if "processed" is interpreted in the benign sense as used in discussing Figure 3), and this paper shows, for mixing at least, that *the processing of a schemata family takes the same functional form independent of which family is being considered*. Moreover, taking $\xi = \mathbf{1}$ shows the functional form is that for strings.

One may further observe (Corollary 6.2) that *the following commutative diagram holds, in parallel, for every choice of schema family, simultaneously*.

$$\begin{tikzcd}
x \arrow[r] \arrow[d, "\Xi"'] & \mathcal{M}(x) \arrow[d, "\Xi"] \\
\Xi x \arrow[r] & \mathcal{M}_\xi(\Xi x)
\end{tikzcd}$$

Because this latter result does speak to parallelism and schemata – subjects which implicit parallelism has classically dealt with – Vose (1999) has redefined the phrase "implicit parallelism" to refer to it.[6] The reader is cautioned, however, that even though the commutative diagram holds, in parallel, for every choice of schema family, simultaneously, there is no "processing leverage" resulting from that fact which confers "tremendous power" or "critical advantages" to genetic algorithms over enumerative processes.

In view of implicit parallelism as (re)defined above, one might wonder whether a change of basis (from standard coordinates to coordinates that are somehow associated with schemata) could shed light on mixing. That is indeed the case, as is explained in Vose (1999).[7]

9 Conclusion

The main contributions of this paper are

- generalizing schemata and schema families in terms of quotient rings,
- proving, in general, the commutativity relation $\Xi\mathcal{M}(x) = \mathcal{M}_\xi(\Xi x)$,
- establishing, in general, the implicit parallelism result of Vose,
- discussing "implicit parallelism" and interpreting it geometrically.

[5]They are worse because of time spent resampling already seen points, and even when using a metric of *online performance*, a genetic algorithm is (on average) not superior to enumeration.

[6]**...**in the binary case. This paper establishes the result more generally.

[7]See, in particular, chapters 6,16,17,19.

References

Bridges, C. L. and Goldberg, D. E. (1987). An analysis of reproduction and crossover in a binary-coded genetic algorithm. In Grefenstette, J., editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 9–13, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley, Reading, Massachusetts.

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

Juliany, J. and Vose, M. D. (1994). The genetic algorithm fractal. *Evolutionary Computation*, 2(2):165–180.

Koehler, G., Bhattacharyya, S., and Vose, M. D. (1997). General cardinality genetic algorithms. *Evolutionary Computation*, 5(4):439–459.

Stephens, C. R. and Waelbroeck, H. (1997). Effective degrees of freedom in genetic algorithms and the block hypothesis. In Back, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 34–40, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Stephens, C. R., Waelbroeck, H., and Aguirre, R. (1999). Schemata as building blocks: does size matter? In Banzhaf, W. and Reeves, C., editors, *Foundations of Genetic Algorithms V*, pages 117–134, Morgan Kaufmann, San Mateo, California.

Vose, M. D. (1990). Formalizing Genetic Algorithms. In the proceedings of an IEEE workshop: *Genetic Algorithms, Neural Nets, and Simulated Annealing Applied to Problems in Signal and Image Processing*, IEEE Computer Society, Los Alamitos, California.

Vose, M. D. (1993). A critical examination of the schema theorem. Technical Report CS–93–212, University of Tennessee, Knoxville, Tennessee.

Vose, M. D. (1998). Random heuristic search. Technical Report CS–98–402, University of Tennessee, Knoxville, Tennessee.

Vose, M. D. (1999). *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, Massachusetts.

Vose, M. D. and Rowe, J. (2000). Random Heuristic Search, Applications to GAs and Functions of Unitation. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4): 195–220.

Vose, M. D. and Wright, A. H. (1998a). The Simple Genetic Algorithm and the Walsh Transform: Part I, Theory. *Evolutionary Computation*, 6(3):253–273.

Vose, M. D. and Wright, A .H. (1998b) The Simple Genetic Algorithm and the Walsh Transform: Part II, The Inverse. *Evolutionary Computation*, 6(3):275–289.

Whitley, D. (1993). An executable model of a simple genetic algorithm. In Whitley, L. D., editor, *Foundations of Genetic Algorithms 2*, pages 45–62, Morgan Kaufmann, San Mateo, California.

Wolpert, D. and Macready, W. (1995). No free lunch theorems for search. Technical Report SFI-TR-95-02-010,Santa Fe Institute, Santa Fe, New Mexico.