Covariant Genetic Dynamics

Chryssomalis Chryssomalakos chryss@nucleares.unam.mx
Instituto de Ciencias Nucleares
Universidad Nacional Autónoma de México
A. Postal 70-543, 04510 México, D.F., MEXICO

Christopher R. Stephens stephens@nucleares.unam.mx
Instituto de Ciencias Nucleares
Universidad Nacional Autónoma de México
A. Postal 70-543, 04510 México, D.F., MEXICO

Abstract

We present a covariant form for the dynamics of a canonical GA of arbitrary cardinality, showing how each genetic operator can be uniquely represented by a mathematical object — a tensor — that transforms simply under a general linear coordinate transformation. For mutation and recombination these tensors can be written as tensor products of the analogous tensors for one-bit strings thus giving a greatly simplified formulation of the dynamics. We analyze the three most well known coordinate systems — string, Walsh and Building Block — discussing their relative advantages and disadvantages with respect to the different operators, showing how one may transform from one to the other, and that the associated coordinate transformation matrices can be written as a tensor product of the corresponding one-bit matrices. We also show that in the Building Block basis the dynamical equations for all Building Blocks can be generated from the equation for the most fine-grained block (string) by a certain projection ("zapping").

Keywords

Genetic dynamics, genetic algorithms, recombination, Building Blocks, coordinate transformations, tensor analysis.

1 Introduction

It is well known that the description of a system and/or its dynamics may be much more amenable to analysis and possible solution in one coordinate system than in another — this is a common occurrence in physics, engineering and chemistry, among other fields. For instance, the dynamics of a set of coupled oscillators has a simpler description in terms of the normal modes of the system, rather than the positions of the oscillators. This is true also of the dynamics of complicated molecules in chemistry. Underlying the dynamics, there is often a set of "fundamental" degrees of freedom, for instance, in the above examples, the deviations of the oscillators, or the atoms in the molecule, from their equilibrium positions. However, the dynamics, more often than not, is complicated in terms of these fundamental degrees of freedom, and so one searches for a description in terms of more appropriate "effective" degrees of freedom[1]. As another simple example: the dynamics of a system of particles is often simplified

[1] The term "effective" might carry an implication of coarse-graining, i.e., loss of information in the description. We emphasize that this is *not* the case here, rather, we use "effective degrees of freedom" the way, e.g., "normal modes" is used in mechanics.

when one introduces the center-of-mass coordinates and the relative separation of the particles.

Although the search for a more appropriate description of a system can be understood, in a dynamical setting, in terms of the search for effective degrees of freedom, the general concept, of course, goes well beyond this. Principal Components Analysis, for instance, is based on the same notion. The place where coordinate transformations play a truly fundamental role though, is in the theory of relativity in physics. An important concept there is the concept of *covariance*, which is most naturally discussed in the language of tensors[2], mathematical objects with simple transformation properties under linear coordinate transformations. The idea behind covariance in physics is that the "laws of physics" should have the same form in any coordinate system. This has the mathematical implication that such laws should be expressible in terms of equations among tensors, the transformation properties of the latter guaranteeing form invariance, or "covariance". To write dynamical equations in covariant form means that all the power of the tensor machinery can be brought to bear, so that the "book-keeping" associated with how the equations and their solutions change from one coordinate system to another is taken care of. Covariance also implies that, if one can generate a solution of the dynamics in covariant form in one coordinate system, then the solution in any other system can be obtained by a simple linear transformation. Having advertised the charms of the tensor formulation, we should also assure the reader unfamiliar with the subject that our use of it in this paper amounts to no more than standard multilinear algebra — the intricacies of tensor *analysis* do not show up in our approach.

So, what has this to do with genetic dynamics? It has been known for some time now that certain genetic operators are most naturally associated with a particular coordinate system. Selection, for instance, is "simple" in the string basis, in that each of the unnormalized string frequencies, y_I, evolves independently from the others. The only non-linearity comes from an overall normalization constant which is implicit in the relation $P_I = y_I / \sum_J y_J$ between the unnormalized variables y_I and the string frequencies P_I. This is an old result, quite well known across several different communities. In the context of EC one may consult, for example, (Reeves and Rowe, 2003). Mutation, on the other hand, couples strings together by actually converting one string into another. However, it is well known that in the Walsh basis (Bethke, 1980; Goldberg, 1989a; Goldberg, 1989b; Vose and Wright, 1998a; Vose and Wright, 1998b) the mutation operator is diagonal, with a consequent simplification of the analysis. In this case, the effective degrees of freedom are the Walsh modes rather than the strings. Finally, recombination is, in general, a more complicated operator than selection or mutation. It has been found recently though (Stephens, 2002; Chryssomalakos and Stephens, 2004) that it, too, admits a simpler description in a particular coordinate system — the Building Block (BB) basis, where the effective degrees of freedom are not strings but particular schemata — Building Block schemata. The BB basis has already been found useful in concrete calculations (McPhee and Crane, 2005), as well as being interestingly related to geometric quantities in the theory of information (Toussaint, 2004).

The transformation to the BB basis, in fact, leads to dynamical equations that are identical to those found via a coarse-graining (Stephens and Waelbroeck, 1997; Stephens and Waelbroeck, 1999; Stephens, 2001), and that have been extended to variable-length linear representations and trees (Poli, 2000; Poli, 2001a; Poli 2001b). Such formulations have led to many new results, including new exact Schema theo-

[2]We will introduce in what follows all the needed concepts regarding tensors — further information can be found in (Akivis and Goldberg, 1977).

rems for Genetic Algorithms (GAs) and Genetic Programming (GP). This, in its turn, has led (Langdon and Poli, 2002; Stephens and Poli, 2004; Stephens and Zamora, 2003) to a more unified point of view of Evolutionary Computation (EC). Other coordinate systems, besides these three principal ones, have also been considered, such as the Taylor basis (Weinberger, 1991) in the context of the analysis of fitness functions. Unlike the Walsh basis, however, no real gain or insight seems to have been gleaned in this latter basis. Toussaint has given an interesting picture (Toussaint, 2004) of how bases that naturally appear in the dynamics of GAs also appear as natural bases for the space of probability distributions.

Given the importance of different coordinate systems for different genetic operators it is clear that a mathematical formulation of the dynamics that is covariant would be very useful, facilitating calculations as well as giving greater insight into the mathematical structure underlying the dynamics. The "laws of physics" alluded to above, that must remain the same in any coordinate system, have their analog in genetic dynamics, in that the intrinsic dynamics of a population, in evolutionary terms, should be independent of the coordinate system that we use to analyze it. Hence, in this paper, extending greatly the preliminary work in (Stephens, 2002; Stephens and Zamora, 2003; Chryssomalakos and Stephens, 2004), we present for the first time a tensorial formulation of the dynamics of an arbitrary cardinality GA with any homologous crossover, arbitrary selection and point mutation within which transformations between the different coordinate bases on the configuration space of length-N, cardinality-K strings may be considered.

The format of the paper is as follows: in Section 2 we introduce necessary mathematical preliminaries, concerning characteristic functions, vector spaces, their duals, arbitrary tensors, and tensor products. We emphasize, in particular, how tensors transform under linear coordinate transformations. Note that the use of tensor product spaces in the theory of GAs is common, as the configuration space for a finite population can be naturally written as a tensor product of the configuration spaces of the individual strings in the population. For example, in (Schmitt, 2004), a tensor product construction is used for the configuration space of populations, in the case of crossover and mutation, and for the configuration space of an individual string, in the case of mutation. However, there is no analysis of crossover as a tensor product on the latter space and, moreover, there is no discussion of the tensorial nature of either crossover or mutation with respect to coordinate transformations and the related concept of covariance.

In Section 3 we introduce the three principal bases — string, Walsh and Building Block — in the context of characteristic functions for binary strings. We show that the Walsh and BB bases can be obtained from the string basis by linear coordinate transformations, writing the associated transformation matrices as tensor products of the corresponding one-bit matrices. In Section 4 we introduce the duals of these bases, and find that the Taylor and BB bases are dual to each other. In Section 5 we extend these results from the binary alphabet case to that of arbitrary cardinality[3]. In Section 6 we present a covariant form of the evolution equation for a canonical GA, showing how it is constructed of three tensors, one for each of the genetic operators — selection, mutation and recombination.

In Section 7 we use this covariant formulation to show how the dynamical equations look in the different coordinate systems for the different genetic operators, and

[3]There is a great deal of literature on GAs extended to the case of arbitrary cardinality, see for example, (Koehler et al, 1997; Rowe et al 2004).

discuss the relative advantages and disadvantages of each coordinate system for analyzing a given operator. We pay particular attention to the most complicated operator — recombination. We emphasize especially the benefits of the BB basis in the analysis of recombination. We also explicitly show how, by writing solutions of the dynamical equations found in a particular coordinate system in a covariant form, one can find the solutions in any other coordinate system via an appropriate linear transformation. We show then, for cardinality-K alphabets, that the fundamental tensors that govern mutation and recombination can be written as tensor products of 1-Kit tensors (we use "Kit" as the natural upgrade of "bit" in the higher cardinality case). Although this is well known in the case of mutation (see for example, (Griffiths and Tavaré, 1997)), to our knowledge it is a new result in the case of crossover. The beauty of this is that it tells us that the mutation and recombination of N-Kit strings can be understood very simply from that of 1-Kit strings. Although this is somewhat intuitive and, as mentioned, is known for mutation, it is less so for recombination. As the coordinate transformation matrices can also be generated as Kit-wise tensor products we show that the entire apparatus of transforming coordinate system to another can very naturally be built up by taking tensor products. An important result that can be simply gleaned from the covariant nature of our equations is the "skew-diagonal" nature of the recombination tensor. This has been derived and discussed previously (Stephens, 2002; Stephens and Zamora, 2003; Chryssomalakos and Stephens, 2004) and is implicit in the nature of the coarse-grained evolution equations for Exact Schema theorems. Here, we explicitly identify for the first time the origin of this skew-diagonalization. Finally, we show that it is possible to generate the dynamical equation for any BB schema from that of an arbitrary string by acting on the latter with a "zapping" operator that also has a natural tensor product structure. In Section 8 we draw some conclusions.

Note that throughout the paper small white squares denote the end of an example, whereas small black ones the end of a proof.

2 Mathematical Preliminaries

2.1 Sets and Characteristic Functions

Consider a discrete, finite set A and the commutative algebra $\mathcal{F}_A$ of real-valued functions on A. We will refer to the elements of A as points. To each subset B of A there corresponds the *characteristic function* (CF) $f_B \in \mathcal{F}_A$, which takes the value 1 on each element of B and is zero elsewhere. Conversely, every function $f \in \mathcal{F}_A$, with values in $\{0, 1\}$, defines a subset of A as the locus of the points where it takes the value 1. Thus, one can denote subsets of A by listing the collection of points that make up the subset or by giving the corresponding characteristic function.

2.2 Tensor Products

Given two vector spaces V and W, with bases $B_V = \{f^1, \ldots, f^m\}$ and $B_W = \{\tilde{f}^1, \ldots, \tilde{f}^n\}$ respectively, one defines their *tensor product* $V \otimes W$, as their cartesian product $V \times W$, with the identification $(\lambda v, w) = (v, \lambda w)$, $\lambda \in \mathbb{R}$, so that $(\lambda v) \otimes w = v \otimes (\lambda w) = \lambda(v \otimes w)$, $v \in V$, $w \in W$. Clearly, this is an intrinsic construction since no particular basis is singled out in the definition. A standard basis in $V \otimes W$ is

$$B_{V\otimes W} = \{f^1 \otimes \tilde{f}^1, f^1 \otimes \tilde{f}^2, \ldots, f^1 \otimes \tilde{f}^n, f^2 \otimes \tilde{f}^1, \ldots, f^m \otimes \tilde{f}^n\} \equiv \{f^{ij} \equiv f^i \otimes \tilde{f}^j\}\,, \quad (1)$$

which shows that $\dim(V \otimes W) = \dim(V)\dim(W) = mn$. If $v = (v_1, \ldots, v_m)$, $w = (w_1, \ldots, w_n)$ are two vectors in V and W respectively, then their tensor product is

given by $z \equiv v \otimes w = (v_i f^i) \otimes (w_j \tilde{f}^j) = v_i w_j (f^i \otimes \tilde{f}^j) \equiv z_{ij} f^{ij}$, the last equation defining the coordinates z_{ij} of z, where (ij) is a composite index, taking on the mn values $11, 12, \ldots, 1n, 21, 22, \ldots, 2n, \ldots, m1, m2, \ldots, mn$. In a similar fashion one can construct tensor products of two covectors or of a vector with a covector.

Given linear transformations $A\colon V \to V$, $B\colon W \to W$, with corresponding matrices $(A_i{}^j)$, $(B_k{}^l)$, so that, e.g., $(Av)_i = A_i{}^j v_j$, one may define a new linear transformation, their *tensor product* $D = A \otimes B\colon V \otimes W \to V \otimes W$, with $(mn \times mn)$-matrix $(D_{ik}{}^{jl})$ given by $D_{ik}{}^{jl} = A_i{}^j B_k{}^l$. For example, in the special case of 2×2 matrices,

$$A \otimes B = \begin{pmatrix} aB & bB \\ cB & dB \end{pmatrix} = \begin{pmatrix} ax & ay & bx & by \\ az & aw & bz & bw \\ cx & cy & dx & dy \\ cz & cw & dz & dw \end{pmatrix}, \qquad (2)$$

$$A \equiv \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad B \equiv \begin{pmatrix} x & y \\ z & w \end{pmatrix},$$

the ordering of the composite matrix indices being $(11, 12, 21, 22)$.

2.3 Dual Vector Spaces and the Canonical Element

Given an n-dimensional vector space V, with basis $B_V = \{f^1, \ldots, f^n\}$, an arbitrary vector a in V can be written as $a = \sum_{i=1}^n a_i f^i$, a_i being the component of a along the basis vector f^i. Note that the superscript i in f^i does not refer to a component of a vector but rather labels the n basis vectors of B_V. As sums over indices occur frequently when dealing with tensors we will use throughout the Einstein summation convention, wherein the appearance of a repeated index, once upper and once lower, implies a summation over that index. Thus, for example, the vector a above can be written as $a = a_i f^i$.

The set V^* of real-valued linear functionals on V is also an n-dimensional vector space, called the *dual* of V. Elements of V^*, in analogy with elements of V, are known as covectors. Formally, two vector spaces V, V^*, are dual to each other if there exists a nondegenerate bilinear *inner product*,

$$\langle \cdot, \cdot' \rangle : V^* \otimes V \to \mathbb{R}, \quad x \otimes a \mapsto \langle x, a \rangle \equiv x(a), \qquad (3)$$

i.e., given a covector $x \in V^*$ and a vector $a \in V$, a real number $\langle x, a \rangle$ is assigned which is linear in each of the arguments. The "nondegenerate" qualification above refers to the requirement that if a vector has zero inner products with all covectors then that vector must be zero (and similarly for covectors). Also, the tensor product sign $\otimes$ appearing in (3) denotes cartesian product modulo the relation $\lambda x \otimes a = x \otimes \lambda a$, for any number λ (see Section 2.2). A basis $B_{V^*} = \{e_1, \ldots, e_n\}$ is called *dual* to B_V if $\langle e_i, f^j \rangle = \delta_i^j$. The element $C = e_i \otimes f^i \in V^* \otimes V$ is called the *canonical element* and satisfies

$$\langle e_i, a \rangle f^i = a, \qquad \langle x, f^i \rangle e_i = x, \qquad (4)$$

for all a in V, x in V^*. This implies that the components a_i of a and x^i of x are given by $a_i = \langle e_i, a \rangle$ and $x^i = \langle x, f^i \rangle$ respectively. The inner product $\langle x, a \rangle$ can be expressed as follows

$$\langle x, a \rangle = \langle x^i e_i, a_j f^j \rangle = x^i a_j \langle e_i, f^j \rangle = x^i a_j \delta_i{}^j = x^i a_i. \qquad (5)$$

Under a linear change of basis of V, effected by an invertible matrix Λ, $\mathbf{f}$ and its dual basis $\mathbf{e}$ transform as

$$\mathbf{f} \to \mathbf{f}' = \Lambda \mathbf{f} \qquad \mathbf{e} \to \mathbf{e}' = \mathbf{e} \Lambda^{-1}, \qquad (6)$$

where $\mathbf{f}$ is the column vector $(f^1, \ldots, f^n)^T$ and $\mathbf{e}$ is the row vector $(e_1, \ldots, e_n)$. One easily checks that $\left\langle e_{i'}, f^{j'} \right\rangle = \delta_{i'}{}^{j'}$, i.e., the transformation law (6) preserves duality of the bases. Given that vectors and covectors are geometrical entities that do not depend on the coordinate system used to describe them, we must have $v = v_i f^i = (v')_j (f')^j \equiv v_{j'} f^{j'}$, for $v \in V$, which shows that vector components transform as

$$v_{i'} = v_j (\Lambda^{-1})^j{}_{i'} \,, \tag{7}$$

while, similarly, covector components satisfy ($a \in V^*$)

$$a^{i'} = \Lambda^{i'}{}_j a^j \,. \tag{8}$$

Notice the slight abuse of notation by which, e.g., $v_{i'}$ (rather than $(v')^i$) denotes the i-th component of v in the primed frame.

2.4 Transformation Law for Tensors

Above we defined a covector as a linear functional on vectors, mapping a vector to a real number. One can also think of a vector as a linear functional on covectors. We extended these concepts to tensor products of vectors and covectors. A further generalization is possible — to tensors, which are real-valued linear functionals on an arbitrary (but fixed) number of vectors and covectors.

For example, a matrix (A^{ij}) can be thought of as a linear functional on pairs of vectors, assigning to $(v_1, v_2) \in V \times V$ the number $A(v_1, v_2) = A^{ij}(v_1)_i (v_2)_j$. Notice that the above definition implies that $A(\lambda v_1, v_2) = A(v_1, \lambda v_2)$, showing that A is actually defined on the tensor square of V, $V \otimes V$. In order for A to represent a geometrical entity, whose action is independent of the coordinate system employed, its components $A^{i'j'}$ in the primed frame must be related to those in the unprimed frame by

$$A^{i'j'} = \Lambda^{i'}{}_i \Lambda^{j'}{}_j A^{ij} \,, \tag{9}$$

so that $A(v_1, v_2) = A^{ij}(v_1)_i (v_2)_j = A^{i'j'}(v_1)_{i'} (v_2)_{j'}$. Similar remarks can be made for a matrix $(A^i{}_j)$, considered as a linear functional on $V \otimes V^*$. Invariance of its action implies the transformation law

$$A^{i'}{}_{j'} = \Lambda^{i'}{}_i A^i{}_j (\Lambda^{-1})^j{}_{j'} \,. \tag{10}$$

Notice that the upper index transforms with Λ, while the lower one with Λ^{-1}, i.e., the position of the indices is a mnemonic device for their transformation properties. More generally, linear functionals T may be introduced, which accept p vectors and q covectors as arguments, $T(v_1, \ldots, v_p, a_1, \ldots, a_q) \in \mathbb{R}$, $v_i \in V$, $a_j \in V^*$. The components of a tensor are obtained by evaluating it on basis vectors and covectors

$$T_{j_1 \ldots j_q}{}^{i_1 \ldots i_p} = T(e_{j_1}, \ldots, e_{j_q}, f^{i_1}, \ldots, f^{i_p}) \,. \tag{11}$$

Linear functionals like T above are called (p, q)-*tensors*. According to this definition, vectors are $(0, 1)$-tensors, covectors are $(1, 0)$-tensors, matrices (A_{ij}) are $(0, 2)$-tensors[4], etc. Similarly, the tensor product of two vectors is a tensor of type $(0, 2)$, while the tensor

[4]It goes without saying that the opposite convention is also well represented in the literature, e.g., vectors are elsewhere defined as $(1, 0)$-tensors etc.

product of two covectors is a tensor of type $(2,0)$. More generally, the tensor product of p vectors and q covectors, taken with a certain ordering, is a tensor of type (q,p). A general tensor is a sum of tensor products of the above type.

The tensor product of matrices we saw above generalizes to arbitrary tensors. In this case, it maps two tensors of type (q_1,p_1) and (q_2,p_2), respectively, to a tensor of type (q_1+q_2,p_1+p_2). For example, the tensor product of the $(0,2)$-tensor $T=T^{ij}e_i\otimes e_j$ and the $(1,0)$-tensor $u=u_kf^k$ is the $(1,2)$-tensor $T\otimes u=T^{ij}u_ke_i\otimes e_j\otimes f^k$. Notice that, traditionally, all indices of the same type (upper or lower) are grouped together, otherwise preserving ordering, when taking tensor products, e.g., $(T^i{}_je_i\otimes f^j)\otimes(u^ke_k)=T^i{}_ju^ke_i\otimes e_k\otimes f^j$.

A fundamental property of tensors, as with vectors and covectors, is that they are geometric objects and therefore invariant under any change of coordinate basis. Thus, similar to (9), (10), we obtain the general transformation law for the components of T,

$$T^{i'_1\ldots i'_p}{}_{j'_1\ldots j'_q}=\Lambda^{i'_1}{}_{i_1}\ldots\Lambda^{i'_p}{}_{i_p}T^{i_1\ldots i_p}{}_{j_1\ldots j_q}(\Lambda^{-1})^{j_1}{}_{j'_1}\ldots(\Lambda^{-1})^{j_q}{}_{j'_q}\,. \tag{12}$$

It follows that the summation over one upper and one lower index, still denoted by the Einstein summation convention defined earlier, is an invariant operation (i.e., independent of the basis used), no matter whether the indices belong to the same or two different tensors.

One of the most important advantages of working with tensors is that one may easily distinguish between accidental equalities and true, geometrical ones. For example, one may find, working in a particular basis, that $v^i=a_i$ for the components of a covector v and a vector a. However, since the two transform differently under a change of basis, the equality found is an accidental one, and does not correspond to any geometrical truth. On the other hand, if $v^i=T^{ij}a_j$ holds true in a particular coordinate system, it will hold true in any other, as the reader can easily demonstrate using only (12) (we call such equations *covariant*). This fact can often be taken advantage of by choosing a particular coordinate system that is convenient for a particular calculation.

3 Bases for Characteristic Functions: The Binary Case

Having introduced tensors in a general setting we now restrict to the case of interest for GAs: fixed-length, binary strings. In this case the natural configuration space consists of the vertices of an N-dimensional unit cube C_N, which can be embedded in $\mathbb{R}^N$, so that the points $(0,0,\ldots,0)$ and $(1,1,\ldots,1)$ are antipodal points of the cube. For $\mathbb{R}^N$ we use coordinate functions $\{x_1,\ldots,x_N\}$. Then, restricted to C_N, each x_i, $1\le i\le N$, takes the value 0 or 1. We define $\bar{x}_i\equiv e-x_i$, where e is the unit function on C_N, taking the value 1 on each vertex. For the rest of this section, and the next one, all functions are considered restricted to C_N. In that case, one may impose algebraic relations on the coordinate functions, compatible with their allowed numerical values

$$x_i^2=x_i\,,\qquad\qquad \bar{x}_i^2=\bar{x}_i\,,\qquad\qquad x_i\,\bar{x}_i=0\,. \tag{13}$$

Notice that x_i is the CF for half of the cube (all vertices with $x_i=1$ that lie on the $(N-1)$-dimensional cube (hyperplane) defined by $x_i=1$), and therefore corresponds to the schema $*^{i-1}\;1\;*^{N-i}$, where $*$ is the standard wildcard symbol and $*^i$ means $*$ repeated i times. Similarly, $\bar{x}_i$ is the CF of the other half of the cube, and corresponds to the schema $*^{i-1}\;0\;*^{N-i}$, while e is the CF of the entire cube, in accordance with $x_i+\bar{x}_i=e$. To specify lower dimensional k-cubes, where $k<N$, one needs products

of the coordinates, e.g., for $N = 3$, x_1x_2 specifies the edge connecting the points $(1, 1, 0)$ and $(1, 1, 1)$ (schema $11*$), while $x_1\bar{x}_2\bar{x}_3$ specifies the point $(1, 0, 0)$. In general, there exists a one-to-one correspondence between monomials of degree $N - k$ in x_i, $\bar{x}_i$, and k-cubes, which, in their turn, correspond to particular schemata.

3.1 The δ-Basis

The standard basis in $\mathcal{F}_{C_N}$, i.e., the vector space of real-valued functions on C_N, is the *δ-basis* B_δ, consisting of the CF's of all 2^N vertices of C_N, i.e., of delta-like functions with support on the vertices of the cube,

$$B_\delta = \{\bar{x}_1\bar{x}_2 \dots \bar{x}_N,\ \bar{x}_1\bar{x}_2 \dots \bar{x}_{N-1}x_N,\ \dots,\ x_1x_2 \dots x_N\}\,. \tag{14}$$

We have arbitrarily singled out above the point $(1, 1, \dots, 1)$, its CF being the last element of the basis. The same construction can be based on an arbitrary vertex P, by defining

$$B_\delta^P = \{\bar{\alpha}_1\bar{\alpha}_2 \dots \bar{\alpha}_N,\ \bar{\alpha}_1\bar{\alpha}_2 \dots \bar{\alpha}_{N-1}\alpha_N,\ \dots,\ \alpha_1\alpha_2 \dots \alpha_N\}\,, \tag{15}$$

where the CF of the vertex P is $\alpha_1\alpha_2 \dots \alpha_N$, with each of the α_i being either x_i or $\bar{x}_i$, and defining the bar operation to be involutive ($\bar{\bar{x}} = x$).

The particular ordering of the basis elements we choose above is "odometer"-like: referring to the choice $P = (1, 1, \dots, 1)$, we start at the origin (the antipode of P in the cube), with CF $\bar{x}_1\bar{x}_2 \dots \bar{x}_N$, and let the last factor take on all possible values ($\bar{x}_N$ and x_N, in this case), then the next-to-last factor advances etc.. This is the standard ordering for tensor products of vector spaces mentioned already in Section 2.2 — see Equation (1), where, in our case, a k-cube is considered as a k-fold tensor product of 1-cubes.

3.2 The Walsh Basis

The other basis for fixed length binary strings that has been extensively studied in the dynamics of EAs is the *Walsh basis* (Bethke, 1980; Goldberg, 1989a; Goldberg, 1989b; Vose and Wright, 1998a; Vose and Wright, 1998b) — we will illustrate it for $N = 1$.

Example 1 *The δ and Walsh bases for $N = 1$*

For $N = 1$, the δ-basis of Subsect. 3.1 (corresponding to the vertex with CF x_1) is $B_\delta^{x_1} = \{\bar{x}_1,\ x_1\}$, while the corresponding Walsh basis is $B_{\mathrm{W}}^{x_1} = \{\hat{y}_1, y_1\} = \{(\bar{x}_1 + x_1)/\sqrt{2},\ (\bar{x}_1 - x_1)/\sqrt{2}\}$. Thus we have,

$$\mathbf{x}_{\mathrm{W}}^{x_1} = \hat{\Lambda}_1 \mathbf{x}_\delta^{x_1}\,, \tag{16}$$

where $\mathbf{x}_{\mathrm{W}}^{x_1} = (\hat{y}_1, y_1)^T$, and $\hat{\Lambda}_1 \equiv 2^{-1/2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ does not depend on P (hatted Λ matrices will effect the transition from the δ to the Walsh basis). □

For an N-bit string, the Walsh basis corresponding to the vertex $P = (11 \dots 1)$ can be written as

$$B_{\mathrm{W}}^P = \{\hat{y}_1\hat{y}_2 \dots \hat{y}_N,\ \hat{y}_1\hat{y}_2 \dots \hat{y}_{N-1}y_N,\ \dots,\ y_1y_2 \dots y_N\}\,, \tag{17}$$

in an obvious notation. The column vector of the basis elements in the Walsh basis for N bits can be obtained from the corresponding one for the δ-basis by multiplication with the Walsh matrix $\hat{\Lambda}_N = (\hat{\Lambda}_1)^{\otimes N}$, i.e., $\mathbf{x}_{\mathrm{W}}^P = \hat{\Lambda}_N \mathbf{x}_\delta^P$, relation that remains valid, with the same $\hat{\Lambda}_N$, regardless of the choice of P. From the fact that $\hat{\Lambda}_{n+1} = \hat{\Lambda}_1 \otimes \hat{\Lambda}_n$, one infers the recursion relation

$$\hat{\Lambda}_{n+1} = \frac{1}{\sqrt{2}} \begin{pmatrix} \hat{\Lambda}_n & \hat{\Lambda}_n \\ \hat{\Lambda}_n & -\hat{\Lambda}_n \end{pmatrix}\,. \tag{18}$$

3.3 The Monomial Basis

We now turn to an alternative basis which is a more recent development - the *monomial basis* $B^P_{\rm m}$ - associated to the vertex P of the unit cube. It consists of the CF's of all k-cubes containing P, which are all monomials in the variables that appear in the CF of P. Anticipating the discussion of recombination in Section 7.3, we point out that the monomial basis is essentially identical to the BB basis, which, as was shown in (Stephens, 2002), most naturally enters in the description of recombination. We clarify the construction of $B^P_{\rm m}$ by first considering a couple of examples.

Example 2 *The δ and monomial binary bases for $N = 1$ and $N = 3$*

For $N = 1$, the δ-basis of the previous subsection (corresponding to the vertex with CF x_1) is $B^{x_1}_\delta = \{\bar{x}_1,\, x_1\}$, while the monomial basis (corresponding to the same vertex) is $B^{x_1}_{\rm m} = \{e,\, x_1\}$. Arranging the basis elements in columns, $\mathbf{x}^{x_1}_\delta = (\bar{x}_1, x_1)^T$, $\mathbf{x}^{x_1}_{\rm m} = (e, x_1)^T$ we have,

$$\mathbf{x}^{x_1}_{\rm m} = \Lambda_1 \mathbf{x}^{x_1}_\delta \,, \tag{19}$$

where $\Lambda_1 \equiv \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ does not depend on the chosen vertex P.

For $N = 3$, the vertex $P = (1, 0, 0)$, with CF $x_1\bar{x}_2\bar{x}_3$, induces the following δ and monomial bases

$$B^P_\delta = \{\bar{x}_1x_2x_3,\, \bar{x}_1x_2\bar{x}_3,\, \bar{x}_1\bar{x}_2x_3,\, \bar{x}_1\bar{x}_2\bar{x}_3,\, x_1x_2x_3,\, x_1x_2\bar{x}_3,\, x_1\bar{x}_2x_3,\, x_1\bar{x}_2\bar{x}_3\} \tag{20}$$

$$B^P_{\rm m} = \{e,\, \bar{x}_3,\, \bar{x}_2,\, \bar{x}_2\bar{x}_3,\, x_1,\, x_1\bar{x}_3,\, x_1\bar{x}_2,\, x_1\bar{x}_2\bar{x}_3\}\,, \tag{21}$$

the latter consisting of the CF's of all k-cubes, with $0 \le k \le N = 3$, containing the above vertex, ranging from the entire 3-cube to the vertex itself. The use of the standard tensor product ordering for the basis elements in $B^P_{\rm m}$, mentioned earlier, becomes clear if one substitutes e's for the missing coordinates in each of the above monomials, i.e., writing the basis as $\{eee, ee\bar{x}_3,\, e\bar{x}_2e, \ldots\}$. The matrix Λ_3 that effects the transition between the two bases is the tensor cube of Λ_1 above, $\Lambda_3 = \Lambda_1^{\otimes 3} \equiv \Lambda_1 \otimes \Lambda_1 \otimes \Lambda_1$. □

In the general case, a vertex P with CF $\alpha_1\alpha_2\cdots\alpha_N$, induces the monomial basis

$$B^P_{\rm m} = \{e,\, \alpha_N,\, \ldots,\, \alpha_1 \ldots \alpha_{N-2}\alpha_N,\, \alpha_1 \ldots \alpha_{N-1},\, \alpha_1 \ldots \alpha_{N-1}\alpha_N\}\,, \tag{22}$$

with

$$\mathbf{x}^P_{\rm m} = \Lambda_N \mathbf{x}^P_\delta \,, \qquad\qquad \Lambda_N \equiv \Lambda_1^{\otimes N} \,. \tag{23}$$

Λ_N does not depend on P, as long as the two bases are chosen according to Equations (15), (22) and, owing to the second of (23), it satisfies the recursion relation

$$\Lambda_{n+1} = \Lambda_1 \otimes \Lambda_n = \begin{pmatrix} \Lambda_n & \Lambda_n \\ 0 & \Lambda_n \end{pmatrix} . \tag{24}$$

4 Bases for Points: The Binary Case

We turn now from the vector space $\mathcal{F}_{C_N}$, of real valued functions on C_N, to its dual. Its elements are identified with linear combinations of the 2^N vertices of the cube. Notice that each vertex is an independent basis element — the reader should not confuse the 2^N-dimensional vector space in question with the N-dimensional configuration space (in the latter, the position vectors of the vertices are obviously linearly dependent). The duality mentioned above is via pointwise evaluation, i.e., given a function f and a

vertex g, their inner product is simply the value of f at g, $\langle g, f\rangle = f(g)$, extended by linearity in each of the arguments. We now consider the duals of the δ, Walsh and monomial bases of $\mathcal{F}_{C_N}$.

4.1 The Vertex Basis

We define the *vertex basis* $B_{\rm v}$, as the dual of B_δ — it clearly consists of the vertices g_R of the cube, appropriately ordered, $B_{\rm v} \equiv B_\delta^* = \{g_R\}$. R here is a multi-index, $R = (r_1 \dots r_n)$, with each r_i being either 0 or 1. Taking for concreteness the reference point $P = (11\dots1)$, we arrange the vertices in a row vector, $\mathbf{g}_{\rm v} = (g_{00\dots0},\, g_{00\dots01},\, \dots,\, g_{11\dots1})$. It will prove convenient, in what follows, to endow the vector space generated by the vertices with an abelian group structure, given by translations, $g_R g_S = g_{R+S}$, where the sum of the indices is bit-wise, modulo 2, e.g., $g_1^2 = g_0$, $g_{10}g_{01} = g_{11}$, etc.. The mod 2 feature converts the configuration space to an N-dimensional torus.

4.2 The Dual Walsh Basis

The dual Walsh basis $B_{\rm W^*}$ is given in (row) vector form by $\mathbf{g}_{\rm W} = \mathbf{g}_{\rm v}(\hat\Lambda_N)^{-1}$, which gives, e.g., for $N = 1$, $\mathbf{g}_{\rm W} = (g_0 + g_1, g_0 - g_1)/\sqrt{2}$. These linear combinations are formally identical to the ones for $B_{\rm W}$ (see Example 1), which is actually true for any N, owing to the fact that $\hat\Lambda_N = \hat\Lambda_N^{-1} = \hat\Lambda_N^T$. Needless to say, this is *not* a special property of the Walsh basis, but rather, it can be regarded as a special relation between the δ and the Walsh bases, or their duals.

4.3 The Taylor Basis

Dual to the basis $B_{\rm m}$ of the k-cube CF's (arranged in the column vector $\mathbf{x}_{\rm m} = \Lambda_N \mathbf{x}_\delta$) is the *Taylor basis* $B_{\rm T} \equiv B_{\rm m}^*$, given in (row) vector form by $\mathbf{g}_{\rm T} = \mathbf{g}_{\rm v}\Lambda_N^{-1}$. To illustrate its geometrical meaning consider the following[5]

Example 3 *δ, monomial, vertex and Taylor binary bases for N=2*

We have

$$\mathbf{x}_\delta = \begin{pmatrix} \bar x_1 \bar x_2 \\ \bar x_1 x_2 \\ x_1 \bar x_2 \\ x_1 x_2 \end{pmatrix}, \qquad \mathbf{x}_{\rm m} = \Lambda_2 \mathbf{x}_\delta = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \bar x_1 \bar x_2 \\ \bar x_1 x_2 \\ x_1 \bar x_2 \\ x_1 x_2 \end{pmatrix} = \begin{pmatrix} e \\ x_2 \\ x_1 \\ x_1 x_2 \end{pmatrix}, \tag{25}$$

while, for the dual bases, we compute

$$\begin{aligned} \mathbf{g}_{\rm v} &= \begin{pmatrix} g_{00},\, g_{01},\, g_{10},\, g_{11} \end{pmatrix} \\ \mathbf{g}_{\rm T} &= \mathbf{g}_{\rm v}\Lambda_2^{-1} = \begin{pmatrix} g_{00},\, g_{01},\, g_{10},\, g_{11} \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} g_{00},\ g_{01} - g_{00},\ g_{10} - g_{00},\ g_{11} - g_{10} - g_{01} + g_{00} \end{pmatrix}. \end{aligned} \tag{26}$$

Apply now the first of (4) to an arbitrary function $f \in \mathcal{F}_{C_N}$, using the pair of dual bases $B_{\rm m}$, $B_{\rm T}$,

$$\begin{aligned} f &= \langle (\mathbf{g}_{\rm T})_i, f\rangle\, (\mathbf{x}_{\rm m})^i \\ &= f(g_{00})\, e + \big[f(g_{01}) - f(g_{00})\big]\, x_2 + \big[f(g_{10}) - f(g_{00})\big]\, x_1 \\ &\quad + \big[f(g_{11}) - f(g_{10}) - f(g_{01}) + f(g_{00})\big]\, x_1 x_2 \,. \end{aligned} \tag{27}$$

[5]The Taylor basis was considered, in a different context, by Weinberger in (Weinberger, 1991).

Thus, one obtains the Taylor expansion of f around the origin, with

$$\partial_1 \equiv g_{10} - g_{00}\,, \qquad \partial_2 \equiv g_{01} - g_{00}\,, \qquad \partial_{12} \equiv g_{11} - g_{10} - g_{01} + g_{00}\,, \tag{28}$$

being discrete derivative operators which, given that f is at most linear in the x_i's, coincide with the exact ones. Notice that $\partial_{12} = \partial_1\partial_2$, where the product is that of the abelian group mentioned earlier. □

Since, for higher N, $B_{\rm m}$ still consists of all monomials in the x_i's, it is easy to see that the above interpretation of the elements of $B_{\rm T}$ persists for all N. Using the group structure, one may describe $B_{\rm m}$ as the set of all monomials in the first order derivatives that, when expressed in terms of the vertices, do not "wrap around" the torus, e.g., in the above example, only the product $\partial_1\partial_2$ appears as a basis element, ∂_1^2, ∂_2^2 etc.would involve wrapping around.

5 Bases for Length-K Alphabets

The results of the previous two sections can be generalized to the case of N-Kit[6] strings of a cardinality-K alphabet. The configuration space consists now of the vertices of an N-dimensional cubic lattice of length $K-1$ in each (basic) direction, i.e., of the points with cartesian coordinates $(r_1, r_2, \ldots, r_N)$, with $r_i = 0, 1, \ldots, K-1$, $1 \le i \le N$. Restricting the coordinates x_i on this lattice, one may impose the relations

$$x_i^{(K)} \equiv x_i(x_i-1)\ldots(x_i-K+1) = 0\,, \quad i = 1, \ldots, N\,. \tag{29}$$

5.1 δ and Vertex bases

The generalization of the δ and vertex bases is straightforward: the δ-basis B_δ in the space of characteristic functions consists of K^N unit-amplitude delta-like functions, each with support on a single vertex. When the coordinates of the latter are $(n_1, n_2, \ldots, n_N)$, the corresponding delta function $\delta_{n_1\ldots n_N}$ is given by

$$\delta_{n_1\ldots n_N} = \delta^1_{n_1}\ldots\delta^N_{n_N}\,, \tag{30}$$

where

$$\delta^i_{n_j} \equiv \mathcal{N}_j x_i(x_i-1)\ldots\widehat{(x_i-n_j)}\ldots(x_i-K+1)\,, \qquad \mathcal{N}_j^{-1} \equiv (-1)^{K-n_j-1} n_j!(K-n_j-1)! \tag{31}$$

(hats denote omission). Notice that $\delta^i_{n_j}$ is the CF of the $x_i = n_j$ hyperplane, and the above product formula for $\delta_{n_1\ldots n_N}$ defines the vertex $g_{n_1\ldots n_N}$ as the intersection of the $x_i = n_i$ hyperplanes. The expansion of an arbitrary function f in the δ-basis is particularly simple, the coefficient of a basis element (delta function) being the value of f at the corresponding vertex,

$$f = \sum_g f(g)\delta_g\,, \tag{32}$$

where g ranges over all vertices and δ_g has support on g (this is an application of the general formula (4)). The dual vertex basis $B_{\rm v}$ is the collection of the vertices of the lattice — we use again the odometer ordering.

Example 4 *δ and vertex bases for $N = 2$, $K = 3$ strings*

[6] As mentioned in the introduction, we propose "Kit" as a natural upgrade of "bit".

The configuration space is $3^2 = 9$-dimensional. Taking the alphabet to be $\{0, 1, 2\}$, the vertex basis, in the odometer ordering, is given by

$$B_{\mathrm{v}} = \{g_{00}, g_{01}, g_{02}, g_{10}, g_{11}, g_{12}, g_{20}, g_{21}, g_{22}\}\,. \tag{33}$$

The dual δ-basis is

$$\begin{aligned} B_\delta &= \{\delta_{00},\, \delta_{01}, \ldots,\, \delta_{22}\} \\ &= \{\delta_0^1\delta_0^2,\, \delta_0^1\delta_1^2, \ldots,\, \delta_2^1\delta_2^2\} \\ &= \{(x_1-1)(x_1-2)(x_2-1)(x_2-2)/4,\, -(x_1-1)(x_1-2)x_2(x_2-2)/2, \\ &\quad \ldots,\, x_1(x_1-1)x_2(x_2-1)/4\}\,. \end{aligned} \tag{34}$$

One may express the coordinates x_i in terms of the δ's either by inverting the above formulas or by direct use of (32), e.g.,

$$x_1 = \delta_{10} + \delta_{11} + \delta_{12} + 2\delta_{20} + 2\delta_{21} + 2\delta_{22}\,. \tag{35}$$

□

5.2 Monomial and Taylor Bases

Our starting point for the generalization of the monomial and Taylor bases to longer alphabets is the interpretation, in the binary alphabet case, of the elements of the Taylor basis as discrete differential operators. As we will see later on in Section 7 this is not the only possibility, but it is a natural one from the algebraic point of view. We start therefore from the construction of the Taylor basis, which we take to consist of all monomials in the discrete derivatives $\partial_i \equiv g_{0\ldots1\ldots0} - g_{0\ldots0}$ (with the 1 in the i-th position), appropriately normalized[7],

$$\begin{aligned} B_{\mathrm{v}} &= \left\{\partial_R \equiv \frac{1}{r_1!\ldots r_N!}\partial_1^{r_1}\ldots\partial_N^{r_N} \,\middle|\, r_i = 0,\ldots,K-1,\, i = 1,\ldots,N\right\} \\ &= \left\{1, \partial_1, \ldots,\, \partial_N,\, \frac{1}{2}\partial_1^2,\, \partial_1\partial_2,\, \frac{1}{2}\partial_2^2, \ldots,\, \frac{1}{\big((K-1)!\big)^N}\partial_1^{K-1}\ldots\partial_N^{K-1}\right\} \end{aligned} \tag{36}$$

(R here again is a composite index, $R = (r_1 \ldots r_N)$). The dual basis B_{m} is given by

$$\begin{aligned} B_{\mathrm{m}} &= \left\{x^{(S)} \equiv x_1^{(s_1)}\ldots x_N^{(s_N)} \,\middle|\, s_i = 0,\ldots,K-1,\, i = 1,\ldots,N\right\} \\ &= \left\{1, x_1, \ldots,\, x_N,\, x_1^{(2)},\, x_1x_2,\, x_2^{(2)}, \ldots,\, x_1^{(K-1)}\ldots x_N^{(K-1)}\right\}, \end{aligned} \tag{37}$$

with $x^{(n)} \equiv x(x-1)\ldots(x-n+1)$ (so that $x^{(1)} = x$) and $x^{(0)} \equiv 1$. It is easily checked that $\left\langle \partial_R, x^{(S)}\right\rangle = \delta_R{}^S \equiv \delta_{r_1}{}^{s_1}\ldots\delta_{r_N}{}^{s_N}$. Notice that the non-locality of the difference operators ∂_i results in the "spreading out" of the standard momomials x_i^m into $x_i^{(m)}$.

The matrix $\tilde{\Lambda}_{N,K}$ that effects the change of basis, from the δ's to the "monomials" $x^{(r)}$, can be easily computed. Indeed, our analysis, in Section 7.5 below, of its tensor product structure, implies that we only need compute $\tilde{\Lambda}_{1,K}$. This is a $K \times K$ matrix, connecting $x^{(r)}$ to the delta functions on the line, δ_m (we drop here the index i as we

[7] The product employed in the monomials derives again from an abelian group structure, given by translations wrapping around the torus, i.e., by Kit-wise addition of the indices of the vertices, modulo K.

are dealing with a single Kit) . Now, $x^{(r)}$ takes the value $m!/(m-r)!$ at $x=m$, when $m \geq r$, and zero otherwise, implying that

$$x^{(r)} = \sum_{m=r}^{K-1} \frac{m!}{(m-r)!} \delta_m \,, \tag{38}$$

from which we infer

$$(\tilde{\Lambda}_{1,K})_r{}^m = \begin{cases} \frac{m!}{(m-r)!} & \text{if } m \geq r \\ 0 & \text{otherwise} \end{cases} . \tag{39}$$

For the inverse matrix, we observe that

$$\begin{aligned} \frac{1}{m!}\partial^m &= \frac{1}{m!}(g_1 - g_0)^m \\ &= \sum_{r=0}^{m} \frac{(-1)^{m-r}}{r!(m-r)!} g_1^r g_0^{m-r} \\ &= \sum_{r=0}^{m} \frac{(-1)^{m-r}}{r!(m-r)!} g_r \,, \end{aligned} \tag{40}$$

where the last step uses the abelian (additive) group structure. We conclude that

$$(\tilde{\Lambda}^{-1}_{1,K})_r{}^m = \begin{cases} \frac{(-1)^{m-r}}{r!(m-r)!} & \text{if } m \geq r \\ 0 & \text{otherwise} \end{cases} . \tag{41}$$

Notice that K does not enter explicitly in the above expressions, making $\tilde{\Lambda}_{1,K}$ a submatrix of $\tilde{\Lambda}_{1,K+1}$ (similarly for $\tilde{\Lambda}^{-1}$).

Example 5 *Monomial and Taylor bases for $N=2$, $K=3$ strings*

The monomial and its dual Taylor bases are

$$\begin{aligned} B_{\mathrm{m}} &= \left\{1,\, x_1,\, x_2\,, x_1^{(2)},\, x_1 x_2,\, x_2^{(2)},\, x_1^{(2)} x_2,\, x_1 x_2^{(2)},\, x_1^{(2)} x_2^{(2)}\right\} \\ B_{\mathrm{T}} &= \left\{1, \partial_1, \partial_2, \frac{1}{2}\partial_1^2,\, \partial_1\partial_2,\, \frac{1}{2}\partial_2^2,\, \frac{1}{2}\partial_1^2\partial_2,\, \frac{1}{2}\partial_1\partial_2^2,\, \frac{1}{4}\partial_1^2\partial_2^2\right\}. \end{aligned} \tag{42}$$

The matrix $\tilde{\Lambda}_{2,3}$ effecting the change of basis according to $\mathbf{x}_{\mathrm{m}} = \tilde{\Lambda}_{2,3}\mathbf{x}_\delta$, where $\mathbf{x}$ denotes, in each case, the column vector of the basic CF's, is given by the tensor square of $\tilde{\Lambda}_{1,3}$. The latter, as well as its inverse, can be computed from (39), (41), giving

$$\tilde{\Lambda}_{1,3} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix} , \qquad \tilde{\Lambda}^{-1}_{1,3} = \begin{pmatrix} 1 & -1 & \frac{1}{2} \\ 0 & 1 & -1 \\ 0 & 0 & \frac{1}{2} \end{pmatrix} , \tag{43}$$

so that

$$\tilde{\Lambda}_{2,3} = \begin{pmatrix} \tilde{\Lambda}_{1,3} & \tilde{\Lambda}_{1,3} & \tilde{\Lambda}_{1,3} \\ 0 & \tilde{\Lambda}_{1,3} & 2\tilde{\Lambda}_{1,3} \\ 0 & 0 & 2\tilde{\Lambda}_{1,3} \end{pmatrix} , \qquad \tilde{\Lambda}^{-1}_{2,3} = \begin{pmatrix} \tilde{\Lambda}^{-1}_{1,3} & -\tilde{\Lambda}^{-1}_{1,3} & \frac{1}{2}\tilde{\Lambda}^{-1}_{1,3} \\ 0 & \tilde{\Lambda}^{-1}_{1,3} & -\tilde{\Lambda}^{-1}_{1,3} \\ 0 & 0 & \frac{1}{2}\tilde{\Lambda}^{-1}_{1,3} \end{pmatrix} . \tag{44}$$

□

6 Covariant Genetic Dynamics

We now consider dynamical evolution in the different bases for an arbitrary cardinality, in the presence of the genetic operators of selection, mutation and recombination, focusing primarily on the latter. The state of the system, at time t, is described by the K^N-component vector $\mathbf{P}(t) = P_I(t)e^I$, the physical interpretation of the components of which is basis-dependent, and where the e^I form a basis in the vector space spanned by the vertices of C_N. In the vertex basis associated with the strings, i.e., when e^I are the vertices of C_N, the components $P_I(t)$ give the proportion of the string I at time t. P_I is therefore a stochastic variable (when the string population is finite), whose time evolution is governed by

$$E(P_I(t+1)) = M_I{}^J P_J^c(t)\,, \tag{45}$$

where $E(P_I(t))$ is the expected value of $P_I(t)$, $P_I^c(t)$, in the vertex basis, is the proportion of string J after selection and recombination and $M_I{}^J$ are the components of the mutation matrix $\mathbf{M}$. For example, for standard point-like mutation for binary strings $M_I{}^J$ is given, in the string basis, by $M_I{}^J = p^{d(I,J)}(1-p)^{N-d(I,J)}$, where $d(I,J)$ is the Hamming distance between the strings I and J, and p is the individual locus mutation probability.

Interpreting (45) in the light of our discussion on tensors in Section 2.4, $P_I(t)$ and $P_J^c(t)$ are the components of $(0,1)$ tensors, while the mutation matrix is a $(1,1)$ tensor. In the form shown, (45) is a tensor equation, meaning that, under a coordinate transformation to a primed system, both sides of the equation transform in the same way, with the result that the functional relations among the primed quantities are identical to the ones among the unprimed ones. We thus have a covariant formulation of genetic dynamics.

P_I^c can further be written as[8]

$$P_I^c = (1-p_c)P_I' + p_c(P_I' + G_I - L_I)\,, \tag{46}$$

p_c being the probability that recombination takes place. In the vertex basis, P_I' is the probability to select I. For example, in the case of proportional selection, P_I' is given by $P_I' = F_I{}^J P_J$, where $F_I{}^J = (f(I)/\bar{f}(t))\delta_I{}^J$ are the matrix elements of the fitness matrix $\mathbf{F}$ (still in the vertex basis)[9] $\bar{f}(t)$ is the average population fitness. The gains term G_I in (46) counts the total number of children of type I produced at time t, while disappearing parents of type I are counted by the losses term L_I[10]. Clearly, L_I is equal to P_I', since, by our definition, every parent participating in recombination is lost, while G_I can be written as

$$G_I = \sum_M \frac{1}{2}\big(p(M) + p(\bar{M})\big)\lambda_I{}^{JK}(M)P_J'P_K'\,, \tag{47}$$

where $\lambda_I{}^{JK}(M)$ is the conditional probability that given parent strings J and K, and a recombination mask M, the offspring I is formed[11]. This probability is either 0 or 1.

[8] Henceforth all time-dependent quantities are evaluated at time t, unless explicitly shown otherwise.

[9] Notice that the index I in $f(I)$ is not a tensor index — the transformation properties of the $f(I)$'s are determined by the fact that they are the diagonal entries of a $(1,1)$-tensor.

[10] This organization of terms is different from the one used in past work (Stephens and Waelbroeck, 1997; Stephens, 1999), where only net gains and losses were counted. Final answers are independent of these different ways of counting.

[11] We will here restrict to standard homologous recombination though by introducing the concept of a Generalized Crossover Mask (Poli and Stephens, 2005) more general forms of recombination can be treated in a similar fashion.

The recombination mask $M = (m_1, \ldots, m_N)$ specifies from which parent a particular offspring locus is obtained. $m_i = 0\ (1)$ signifies that the i-th locus of the offspring is taken from the i-th locus of the first (second) parent. $p(M)$ is the conditional probability of applying mask M, given that crossover is done. $\bar{M}$ denotes the conjugate mask, i.e., such that $M + \bar{M} = (11 \ldots 1)$ — we explain further the form of (47) in Section 7.3 below.

Substituting the expressions for G_I and L_I, (45) becomes

$$P_I(t+1) = M_I{}^J \left((1-p_c) P'_J + p_c \sum_M \frac{1}{2} \big(p(M) + p(\bar{M}) \big) \lambda_J{}^{KL}(M) P'_K P'_L \right) . \qquad (48)$$

which is a covariant form of the dynamics of a system of fixed length strings evolving in the presence of selection, mutation and homologous recombination. In the vertex basis it is equivalent to standard formulations in population genetics (Buerger 2000) and EC (Vose 1999)).

Despite the covariance of (48), the facility of its analysis as well as its physical interpretation are basis-dependent. The advantage of a covariant formulation is that we may choose a convenient coordinate system for a given problem, while always dealing with the same underlying equation, coordinate transformation matrices being used to transform from one coordinate system to another. Fundamentally, the dynamics is governed by: the mutation matrix, a $(1,1)$ tensor with components $M_I{}^J$; the $(2,1)$ tensor $\lambda(M)$, with components $\lambda_I{}^{JK}(M)$; the mask probability distribution $p(M)$ and the fitness values $f(I)$. In this sense the evolutionary algorithm is a "black box" whose output depends on a large set of parameters. It is therefore worth looking for symmetries and regularities that may be exploited to effect a natural coarse graining, making manifest the effective degrees of freedom of the dynamics.

7 What Basis for Genetic Dynamics?

In Section 6 we wrote down the fundamental dynamical equation for the canonical GA in covariant form. In Sections 3.1, 3.2 and 3.3 we introduced three separate bases in which the dynamics may be examined. We will now see how this dynamics appears in these bases. Rather than immediately considering all three operators — selection, mutation and recombination — acting simultaneously, we will first consider them one by one, showing that each basis that we have considered is best adapted for one particular operator. Regarding notation, we will generally denote quantities in the Walsh basis (or its dual) by a hat, and those in the Building Block/monomial (or Taylor) basis by a tilde — the δ (or vertex) basis quantities will often carry no distinctive mark. The Λ matrices connecting the δ-basis to the Walsh and monomial bases, will likewise carry a hat and a tilde respectively.

7.1 Selection

We consider first selection-only dynamics. In Equation (48), selection is "hidden" inside $P'_I(t)$. In the absence of any further information, all that can be said is that the components of $\mathbf{P}'$ transform like a vector. When $\mathbf{P}'$ is given in terms of a fitness matrix, $P'_I = F_I{}^J P_J$, the covariant evolution equation is

$$E(P_I(t+1)) = F_I{}^J P_J(t) , \qquad (49)$$

which in the vertex basis becomes, for proportional selection,

$$E(P_I(t+1)) = \frac{f(I)}{\bar{f}(t)} P_I(t) , \qquad (50)$$

meaning that the strings only couple to each other through $\bar{f}(t)$. In terms of unnormalized variables x_I, where $P_I = x_I / \sum_I x_I$, we may write the equation,

$$E(x_I(t+1)) = f(I)x_I(t)\,, \tag{51}$$

all solutions of which can easily be shown to provide solutions of equation (50).

In the infinite population limit, where $E(P_I(t)) = P_I(t)$, and when the fitness landscape is time-independent, the above equation can be easily integrated, and the transformation back to the P_I done, to give the well known result (see, for example (Buerger 2000))

$$P_I(t) = \frac{f(I)^t P_I(0)}{\sum_I f(I)^t P_I(0)}\,. \tag{52}$$

Simple as it may look, Equation (52) is not satisfactory, as the transformation properties of the r.h.s. are far from clear — we refer to this as lack of explicit covariance. The remedy, though, is simple: rewrite (52) using only tensors and invariant contractions of indices — the reader should have no trouble verifying that the result is

$$P_I(t) = \frac{(\mathbf{F}^t)_I{}^J P_J(0)}{V^I(\mathbf{F}^t)_I{}^J P_J(0)}\,, \tag{53}$$

where the vector $\mathbf{V}$ is given, in the δ-basis, by $\mathbf{V} = (1, 1, \ldots, 1)$. One can now pass to another (primed) coordinate system, transforming the tensors in (53) according to

$$P_{I'} = \Lambda_{I'}{}^J P_J\,, \qquad F_{I'}{}^{J'} = \Lambda_{I'}{}^R F_R{}^S (\Lambda^{-1})_S{}^{J'}\,, \qquad V^{I'} = V^R (\Lambda^{-1})_R{}^{I'}\,. \tag{54}$$

Taking the primed system to be in the Walsh basis, the transformed fitness matrix $\hat{\mathbf{F}} = \hat{\Lambda}\mathbf{F}\hat{\Lambda}^{-1}$ looks complicated, the number and position of non-zero elements depending on the degree of epistasis in the landscape. In the monomial basis, $\tilde{\mathbf{F}} = \tilde{\Lambda}\mathbf{F}\tilde{\Lambda}^{-1}$ is not diagonal either, however, it can be shown (Rowe and Stephens, in preparation) that $\tilde{\mathbf{F}} = \tilde{\mathbf{F}}_D + \mathbf{A}$, where $\tilde{\mathbf{F}}_D$ is diagonal and $\mathbf{A}\tilde{\mathbf{P}} = 0$. Hence the dynamics is given essentially by a diagonal matrix, as in the δ-basis. For proportional selection, $\tilde{\mathbf{F}}_I{}^J = (f(I,t)/\bar{f}(t))\delta_I{}^J$, where $f(I,t)$ is the fitness of the Building Block schema I and is population- (and hence time-) dependent. To illustrate this, for $N = 1$, we transform $\mathbf{FP}$ from the δ to the monomial basis to get

$$\begin{aligned}
\tilde{\mathbf{F}}\tilde{\mathbf{P}} &= (\tilde{\Lambda}\mathbf{F}\tilde{\Lambda}^{-1})\tilde{\Lambda}\mathbf{P} \\
&= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} f_0/\bar{f}(t) & 0 \\ 0 & f_1/\bar{f}(t) \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \end{pmatrix} \\
&= \frac{1}{\bar{f}(t)} \begin{pmatrix} f_0 & -f_0 + f_1 \\ 0 & f_1 \end{pmatrix} \begin{pmatrix} P_* \\ P_1 \end{pmatrix} \\
&= \frac{1}{\bar{f}(t)} \begin{pmatrix} f_* & 0 \\ 0 & f_1 \end{pmatrix} \begin{pmatrix} P_* \\ P_1 \end{pmatrix} + \frac{1}{\bar{f}(t)} \begin{pmatrix} (f_0 - f_1)P_1 & f_1 - f_0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} P_* \\ P_1 \end{pmatrix} \\
&= \frac{1}{\bar{f}(t)} \begin{pmatrix} f_* & 0 \\ 0 & f_1 \end{pmatrix} \begin{pmatrix} P_* \\ P_1 \end{pmatrix} = \frac{1}{\bar{f}(t)} \begin{pmatrix} f_* P_* \\ f_1 P_1 \end{pmatrix}
\end{aligned} \tag{55}$$

where $f_* \equiv f_0 P_0 + f_1 P_1$ and $P_* = P_0 + P_1 = 1$. The important point here is that the diagonal elements of the fitness matrix are precisely the natural definitions of fitness

for the corresponding elements of the BB basis, i.e., for the associated Building Block schemata. Thus, although mathematically one can always define a diagonal fitness matrix in any basis, there is no guarantee that the elements of the fitness matrix are the "natural" fitnesses of the corresponding transformed basis elements.

We argued above that strings were the natural effective degrees of freedom for selection, in that, in the δ-basis, the dynamics is diagonal. The Walsh basis does not look natural in this sense, as in that basis the different Walsh modes, viewed as effective degrees of freedom, are coupled together. On the other hand, in the monomial basis the dynamics is also essentially diagonal and therefore different Building Block schemata are not coupled by selection. In this sense Building Block schemata are every bit as natural as effective degrees of freedom as strings. The difference between the two lies in the fact that in the δ-basis, when the fitness landscape is time independent, the selection-only equation can be simply integrated, as in (52). On the other hand, as the Building Block fitnesses, in the above case, are time and population dependent, the integration of the corresponding equation, $x_I(t+1) = f(I,t)x_I(t)$, is, in general, non-trivial.

Note also, that in the particular case of a multiplicative fitness landscape, where $f(I) = \prod_{r=1}^{N} f(i_r)$, and a similarly multiplicative initial population composition, $P_I(0) = P_{i_1}(0)\dots P_{i_N}(0)$, one can generate the N-bit solution from the tensor product of N 1-bit solutions. Indeed, writing $\bar{f}(t) = \sum_{i_1}\dots\sum_{i_N} f(i_1)\dots f(i_N)P_{i_1}(t)\dots P_{i_N}(t) = \bar{f}(i_1,t)\dots\bar{f}(i_N,t)$, one obtains

$$P_I(t+1) = \frac{f(i_1)\dots f(i_N)}{\bar{f}(i_1,t)\dots\bar{f}(i_N,t)} P_{i_1}(t)\dots P_{i_N}(t)\,, \tag{56}$$

which has as solution, $P_I(t) = \prod_{r=1}^{N} P_{i_r}(t)$, where $P_{i_r}(t) = f(i_r)^t P_{i_r}(0)/\sum_{r,i_r} f(i_r)^t P_{i_r}(0)$, i.e., every locus evolves independently.

7.2 Mutation

The dynamical equation for mutation only is

$$E(P_I(t+1)) = M_I{}^J P_J(t)\,, \tag{57}$$

where $M_I{}^J$ are the components of the mutation matrix $\mathbf{M}$, that transforms under a basis transformation Λ as $\mathbf{M} \to \Lambda\mathbf{M}\Lambda^{-1}$. In the δ-basis, for binary strings, $M_I{}^J = p^{d(I,J)}(1-p)^{N-d(I,J)}$, and the mutation matrix is evidently non-diagonal. Thus, we see that the δ-basis is not particularly convenient for the study of mutation, as the string degrees of freedom are coupled, i.e., the mutation operator converts one string into another. Changing to the Walsh basis however, $\mathbf{M}$ transforms like $\mathbf{M} \to \hat{\mathbf{M}} = \hat{\Lambda}\mathbf{M}\hat{\Lambda}^{-1}$, with the well known result (see, for example (Vose 1999)

$$\hat{\mathbf{M}}_I{}^J = (1-2p)^{|I|}\delta_I{}^J \tag{58}$$

where $|I|$ is the order of the Walsh mode, defined as the Hamming distance of mode I from mode $(00\dots0)$. Thus, $\hat{\mathbf{M}}$ is diagonal, with the entries $(1-2p)^{|I|}$ being its eigenvalues, each with $\binom{N}{|I|}$ associated degenerate eigenvectors. In the Walsh basis, equation (57) gives

$$E(\hat{P}_I(t+1)) = (1-2p)^{|I|}\hat{P}_I(t) \tag{59}$$

where $\hat{P}_I = \hat{\Lambda}_I{}^J P_J$, and which has as solution, in the infinite population limit,

$$\hat{P}_I(t) = (1-2p)^{|I|t}\hat{P}_I(0)\,. \tag{60}$$

The above can be seen even more simply by noticing that, when the mutation probability p_i of the i-th bit is independent of the other bits, the N-bit mutation matrix factorizes into the tensor product of N 1-bit factors,

$$\mathbf{M}_N = \mathbf{M}(p_1) \otimes \mathbf{M}(p_2) \otimes \ldots \otimes \mathbf{M}(p_N)\,, \tag{61}$$

where, for binary strings, $\mathbf{M}(p_i) \equiv \begin{pmatrix} (1-p_i) & p_i \\ p_i & (1-p_i) \end{pmatrix}$ in the δ-basis. The factorizability of $\mathbf{M}_N$ is then preserved in all bases,

$$\mathbf{M}_N \to \Lambda_N \mathbf{M}_N {\Lambda_N}^{-1} = \Lambda_1 \mathbf{M}(p_1) {\Lambda_1}^{-1} \otimes \ldots \otimes \Lambda_1 \mathbf{M}(p_N) {\Lambda_1}^{-1}\,. \tag{62}$$

In the Walsh basis, $\hat{\mathbf{M}}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1-2p_i \end{pmatrix}$, while in the monomial basis, $\tilde{\mathbf{M}}_1 = \begin{pmatrix} 1 & 0 \\ p_i & 1-2p_i \end{pmatrix}$. Thus, we see that, in the monomial basis, the mutation matrix is triangular and hence, as in the case of the Walsh basis, its eigenvalues can simply be read off from the diagonal.

Once again, if we consider the case where the initial probability distribution, in the δ-basis, factorizes, i.e., $P_I(t) = P_{i_1}(t) \ldots P_{i_N}(t)$, then it will factorize in any other basis connected to the δ-basis by a transformation $\Lambda_N = \Lambda_1^{\otimes N}$, as is the case for the Walsh and monomial bases.

Finally, it is clear that the product of factorizable operators is also factorizable. This implies, e.g., that when both selection and mutation are present, and the conditions of factorizability of each of them mentioned earlier are satisfied, each bit evolves essentially independently of the others.

7.3 Binary Alphabet Recombination

We now turn our attention to recombination, the dynamical equation for which is

$$E(P_I(t+1)) = (1-p_c)P_I + p_c \sum_M \frac{1}{2}\big(p(M) + p(\bar{M})\big) {\lambda_I}^{JK}(M) P_J P_K\,. \tag{63}$$

Again, the covariance of (63) guarantees its validity in all bases, with ${\lambda_I}^{JK}$ transforming as a $(2,1)$-tensor, i.e., according to

$${(\lambda')_{I'}}^{J'K'} = {\lambda_I}^{JK} {\Lambda_{I'}}^{I} {(\Lambda^{-1})_{J'}}^{J} {(\Lambda^{-1})_{K'}}^{K} \tag{64}$$

and correspondingly $P_I \to {\Lambda_I}^J P_J$. We will now consider recombination in the δ- and monomial bases. For a discussion of recombination in the Walsh basis, in a different context, see (Wright, 2000).

7.3.1 Binary Alphabet Recombination in the δ ("string") Basis

In this basis P_I is the relative proportion of the string I. For each given mask M, there are generally several pairs of parent strings $\{J, K\}$ that produce I as their child. The tensor $\lambda(M)$ in Equation (47), in this basis, is given by ($\bar{m}_i = 1 - m_i$)

$${\lambda_I}^{JK}(M) = \prod_{r=1}^{N} \left(\bar{m}_r {\delta_{i_r}}^{j_r} + m_r {\delta_{i_r}}^{k_r} \right) \tag{65}$$

which is 1, if the first child of the recombination of J, K, with mask M, is I, and zero otherwise (we use the convention that a 0 (1) in the mask denotes that the first child obtains the corresponding bit from the first (second) parent). Then ${\lambda_I}^{KJ}(M) = {\lambda_I}^{JK}(\bar{M})$ checks whether I is being produced as a second child. Notice the product form of (65), which indicates that recombination functions "bit by bit", as will be further seen in Section 7.5.1, when its tensor product structure is considered.

One may define a mask-independent average λ_I by ${\lambda_I}^{JK} = \sum_M p(M){\lambda_I}^{JK}(M)$, whereupon (47) becomes, in matrix notation,

$$G_I(t) = \mathbf{P}^T R_I \mathbf{P}\,, \qquad\qquad R_I \equiv \frac{1}{2}\big(\lambda_I + \lambda_I^T\big)\,. \tag{66}$$

Once again, due to covariance, an equation analogous to the second of (66) is valid in all bases, since both matrix indices of λ_I are upper. For reasons explained in Section 7.5, λ_I is a more convenient object to work with than R_I. Ignoring selection and mutation, Equation (48) becomes

$$P_I(t+1) = (1-p_c)P_I + p_c \mathbf{P}^T R_I \mathbf{P}\,. \tag{67}$$

Example 6 $N = 2$ *binary alphabet recombination in the δ-basis*

We fix $I = 11$ and take $p(M) = 1/4$ (independent of M). From (65) we compute[12]

$$\lambda_{11}(00) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}\,, \qquad\qquad \lambda_{11}(01) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}\,, \tag{68}$$

while $\lambda_{11}(10) = \lambda_{11}(01)^T$ and $\lambda_{11}(11) = \lambda_{11}(00)^T$. Then

$$\lambda_{11} = \frac{1}{4}\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 \\ 1 & 2 & 2 & 4 \end{pmatrix}\,, \tag{69}$$

and $R_{11} = \lambda_{11}$. Equation (67) then gives

$$P_{11}(t+1) = P_{11} + \frac{p_c}{2}\left(P_{10}P_{01} - P_{11}P_{00}\right)\,. \tag{70}$$

The equations for the other strings can be simply obtained by renaming the indices. □

7.3.2 Binary Alphabet Recombination in the Monomial Basis

As the above example shows, recombination is rather complicated in the δ-basis. A more efficient organization of the various terms that contribute to $G_I(t)$ can be achieved if one thinks in terms of Building Block schemata. For example, 11 can be obtained by recombining the schemata $1*$ and $*1$, where $*$ denotes the standard wildcard symbol. Each string gives rise to 2^N corresponding Building Block schemata associated with it, by all possible substitutions of its bits by $*$'s — the corresponding set of schemata constitutes the BB basis for that string. For example, the string 11 generates the basis

[12] We write, for simplicity, $\lambda_{11}(00)$, rather than $\lambda_{11}((00))$ etc.

$\{**, *1, 1*, 11\}$. Recombination involves the interaction of conjugate schemata only[13], so one expects some sort of "skew diagonalization" of the process in this basis.[14] To connect with the discussion in Section 3.3, notice that substitution of a particular bit by a $*$ corresponds, at the level of CF's, to substitution of a coordinate x_i (or $\bar{x}_i$) by the unit function e. It is then clear that the CF's of the Building Blocks are exactly the elements of the monomial basis of Section 3.3. We conclude that the two bases are essentially the same, with the monomial version formalizing the preexisting Building Block concept. As a corollary, *the Taylor basis is dual to the Building Block basis.*

The CF corresponding to a schema is the sum of the CF's of all vertices (strings) that the schema matches. On the other hand, it is clear that the probability of a certain schema is likewise the sum of the probabilities of all strings that the schema matches. This implies that, in going from one basis to another, probabilities transform like CF's — in particular

$$\tilde{\mathbf{P}} = \tilde{\Lambda}_N \mathbf{P} \,. \tag{71}$$

Example 7 $N = 2$ *binary alphabet recombination in the monomial basis*

One can calculate the mask-averaged interaction term in the monomial basis, $\tilde{\lambda}_I{}^{JK}$, by transforming λ according to (64) (with $\Lambda \to \tilde{\Lambda}_2$), to find, for example, for $\tilde{\lambda}_{11}$,

$$\tilde{\lambda}_{11} = \frac{1}{4} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} . \tag{72}$$

As expected, it is skew diagonal. Regarding the structure of $\tilde{\lambda}_I$, there is one non-zero entry for every mask — in the example above they are all equal to $1/4$ because of the equal probabilities of all masks.

The dynamical equation for $P_{11}(t)$ is

$$P_{11}(t+1) = (1 - \frac{p_c}{2}) P_{11} + \frac{p_c}{2} P_{*1} P_{1*} \,, \tag{73}$$

which by substituting $P_{*1} = P_{11} + P_{01}$, and analogously for P_{1*}, can be seen to coincide with (70). □

The above result generalizes to arbitrary N (assuming again uniform mask probability, see Section 7.5 below)

$$\tilde{\lambda}_{11\ldots1}{}^{JK} = 2^{-N} \delta^{J,\, 2^N + 1 - K} \,. \tag{74}$$

In the δ-basis, the equations for the other elements of the basis can be obtained from the one for $11\ldots1$ by simply renaming the indices. In the monomial basis, the situation is even simpler: one obtains, for example, the equation for $11*$ from the one for 11 simply by attaching an extra $*$ to all indices — this generalizes in the obvious way to any number of $*$'s in any position, so that (74), inserted in (67), gives essentially the equations for all basis elements, for all N.

[13] We define the conjugation $\bar{\ }$ of schemata: the string $I = (i_1 i_2 \ldots i_N)$ generates the basis $\{** \ldots *, ** \ldots i_N, \ldots, i_1 i_2 \ldots i_N\}$ and $\bar{i}_r = *$ while $\bar{*} = i_r$, if the $*$ is in position r — the conjugate of a schema, in this sense, is the schema with conjugate bits.

[14] This skew-diagonalization was first noted in (Stephens 2002) and later discussed by others.

7.4 Recombination for K-Alphabets

The particular generalization of the concept of monomial and Taylor bases to the length-K case, presented in Section 5, was motivated primarily by aesthetic considerations, and what we considered a natural, from the algebraic point of view, way to proceed. Nevertheless, there is no *a priori* guarantee that these criteria would lead to a useful structure, from the point of view of recombination. But they do. Our starting point, in demonstrating this claim, will be the construction of the λ-tensor appropriate for the case at hand. We consider, in turn, the δ and monomial bases, the Walsh basis for K-alphabets was considered in (Vose and Wright, 1998a; Vose and Wright, 1998b) .

7.4.1 K-Alphabet ecombination in the δ Basis

As in the binary case, ${\lambda_I}^{JK}(M)$ is 1, if the first child of the recombination of the strings J, K, with mask M, is equal to I, and zero otherwise. The difference here is that the multiindices I, J, K range over K^N values each, while M is still composed of binary indices m_i, since we are still dealing with quadratic, i.e., two-parent, processes. Accordingly, λ^δ is given by the first of (65), while the mask-averaged ${\lambda_I}^{JK}$ and ${R_I}^{JK}$ are defined as before.

Example 8 $N = 2$, *3-alphabet recombination in the δ-basis*

We fix $I = 12$ and take $p(M) = 1/4$, independent of M. From the first of (65) we find

$$\lambda_{12}(00) = \begin{pmatrix} 0 & 0 & 0 \\ A & A & A \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda_{12}(01) = \begin{pmatrix} 0 & 0 & 0 \\ B & B & B \\ 0 & 0 & 0 \end{pmatrix}, \tag{75}$$

$$\lambda_{12}(10) = \begin{pmatrix} 0 & A & 0 \\ 0 & A & 0 \\ 0 & A & 0 \end{pmatrix}, \quad \lambda_{12}(11) = \begin{pmatrix} 0 & B & 0 \\ 0 & B & 0 \\ 0 & B & 0 \end{pmatrix}, \tag{76}$$

where

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}. \tag{77}$$

The mask-averaged λ is then given by

$$\lambda_{(12)} = \frac{1}{2}\begin{pmatrix} 0 & C & 0 \\ C & 2C & C \\ 0 & C & 0 \end{pmatrix}, \qquad \text{where} \quad C = \frac{1}{2}(A+B) = \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \tag{78}$$

and is equal to $R_{(12)}$. Substituting in (67), we get a somewhat uninspiring dynamical equation for P_{12},

$$\begin{aligned} P_{(12)}(t+1) = (1-p_c)P_{12} + \frac{1}{2}p_c(&P_{02}P_{10} + P_{02}P_{11} + P_{00}P_{12} + P_{01}P_{12} \\ &+ 2P_{02}P_{12} + 2P_{10}P_{12} + 2P_{11}P_{12} + 2P_{12}^2 + 2P_{12}P_{22} \\ &+ P_{12}P_{20} + P_{12}P_{21} + P_{10}P_{22} + P_{11}P_{22}). \end{aligned} \tag{79}$$

□

7.4.2 K-Alphabet Recombination in the Monomial Basis

To avoid having to invent too many new symbols, we retain the values $(0, 1, \ldots, K-1)$ of the indices. In particular, the index 0 will now play the role of the "wildcard" $*$. In the binary case, the 1-bit Λ-matrix, $\tilde{\Lambda} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, offered no middle option between maintaining the identity of the basis element (second row), and completely erasing it (first row). In the K-alphabet case, all monomial basis elements, apart from the first and last one, represent intermediate degrees of coarse graining, that interpolate between the above two. This might be anticipated, but what comes perhaps as a surprise, is the particular form of the coefficients in the "partial sums" provided by, e.g., $\tilde{\Lambda}_{1,3}$ of Equation (43), or, $\tilde{\Lambda}_{1,4}$ below,

$$\tilde{\Lambda}_{1,4} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 2 & 6 \\ 0 & 0 & 0 & 6 \end{pmatrix} . \tag{80}$$

One might, naively, have expected all non-zero entries above to be given by 1's (a true "partial sum"), but duality to the Taylor basis implies this strange-looking averaging procedure. Two features of the $\tilde{\Lambda}$'s are particularly important: triangularity, and the first row's string of 1's (notice that both are preserved by tensor products). The former guarantees a hierarchical ordering of the monomial basis elements, according to their degree of averaging, resulting in a partial decoupling of the dynamical equations, since no P_J can appear in the r.h.s. of the equation for P_I, if J is a "finer" variable than I, i.e., if J corresponds to a higher order schema than I. This fact has been thought to lie at the root of the simplification provided by the monomial basis in the study of recombination. We will prove shortly that it is actually quite irrelevant. The whole magic of the monomial basis stems from the inclusion of the unit function in the basis, thanks to $\tilde{\Lambda}_{1,k}$'s first row — we explain its role in detail below.

Example 9 $N = 2$, *3-alphabet recombination in the monomial basis*

Continuing Example 8, we compute $\tilde{\lambda}_{12}$. We transform λ^{δ} as a rank-3 tensor, using $\tilde{\Lambda}_{2,3}$ of Equations (43), (44) — the result is

$$\tilde{\lambda}_{12} = \frac{1}{2} \begin{pmatrix} 0 & D & 0 \\ D & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \qquad \text{where} \quad D = \frac{1}{2} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} , \tag{81}$$

and is equal to $\tilde{R}_{(12)}$. Substituting in (67), and using $\tilde{P}_{00} = 1$, we get the dynamical equation for $\tilde{P}_{12}$,

$$\tilde{P}_{12}(t+1) = (1 - \frac{p_c}{2})\tilde{P}_{12} + \frac{p_c}{2}\tilde{P}_{10}\tilde{P}_{02} , \tag{82}$$

arguably an improvement over (79). □

As can be appreciated, only complementary schemata enter in the r.h.s. of (82) — notice, in particular, the absence of terms like $\tilde{P}_{11}\tilde{P}_{12}$ that might have been expected. This represents maximal skew-diagonalization, and can be traced to the particular form of the first row of $\tilde{\Lambda}$, as we show, in all generality, in Section 7.5. The practical implication is that, quite generally, in the dynamical equation for $\tilde{P}_I$, only terms $\tilde{P}_J\tilde{P}_K$ enter, such that J, K can "give birth" to I, *with all "unused" Kits equal to zero*.

7.5 The Tensor Product Structure of Recombination

7.5.1 Factorization of the λ Tensor

As we have seen above, the dynamics of recombination in the δ-basis is controlled by the tensor $\lambda(M)$, which contains the information about which parents may give rise to a particular child. In deciding this, one needs to perform a bit-by-bit test, the outcome for the entire string being the logical AND of the individual bit tests (see Equation (65), where AND corresponds to multiplication). The fact that the value of $\lambda(M)$ factorizes in this manner, reflects itself in that $\lambda_I(M)$, for a length-N string, is the tensor product of the λ's of its individual bits,

$$\lambda_I(M) = \lambda_{i_1}(m_1) \otimes \lambda_{i_2}(m_2) \otimes \ldots \otimes \lambda_{i_N}(m_N)\,. \tag{83}$$

A simple calculation then shows that the same is true for the mask-independent λ, i.e., in matrix notation,

$$\lambda_I = \lambda_{i_1} \otimes \ldots \otimes \lambda_{i_N}\,. \tag{84}$$

Finally, given that Λ_N is itself the N-th tensor power of the 1-bit Λ_1, we conclude that *the above statements about λ are valid in all bases*. Notice that ${R_I}^{JK}$ *does not*, in general, factorize in this manner — this is because checking for the first *or* the second child, for $N > 1$, is not a bit-wise operation.

Example 10 *Tensor product structure for a binary alphabet*

Consider $N = 1$ binary recombination in the δ-basis. We find

$$\lambda_0 = \frac{1}{2}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \lambda_1 = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}. \tag{85}$$

Transforming to the monomial basis we find

$$\tilde{\lambda}_* = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}_1 = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{86}$$

The second equation above clearly shows that $\tilde{\lambda}_{11\ldots1} = \tilde{\lambda}_1^{\otimes N}$ is skew-diagonal for all N. Notice also that the λ_{11} given in Equation (69) is just the tensor square of λ_1 given in Equation (85) above. □

Example 11 *Tensor product structure for 3-alphabet*

We calculate the 1-Kit $\lambda_I(M)$ matrices in the δ-basis, as given by the first of (65),

$$\lambda_0(0) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \lambda_0(1) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \tag{87}$$

$$\lambda_1(0) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \lambda_1(1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \tag{88}$$

$$\lambda_2(0) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \qquad \lambda_2(1) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \tag{89}$$

resulting in the mask-averaged λ_I's (with $p(M) = 1/2$)

$$\lambda_0 = \frac{1}{2}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \lambda_1 = \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \lambda_2 = \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 2 \end{pmatrix}. \quad (90)$$

All higher N λ's can be built from these by tensor multiplication. For instance, a look at (75), (77) shows that, e.g., the $\lambda_{12}(01)$ found there is given simply by $\lambda_{12}(01) = \lambda_1(0) \otimes \lambda_2(1)$. Also, the mask-averaged λ_{12} of Equation (78) is given simply by $\lambda_{12} = \lambda_1 \otimes \lambda_2$.

Transforming the above λ's according to (64), with the $\tilde{\Lambda}_{1,3}$ matrix of (43), we find for the $\tilde{\lambda}$'s,

$$\tilde{\lambda}_0(0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}_0(1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (91)$$

$$\tilde{\lambda}_1(0) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}_1(1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (92)$$

$$\tilde{\lambda}_2(0) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}_2(1) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (93)$$

giving the mask-averaged $\tilde{\lambda}_I$'s

$$\tilde{\lambda}_0 = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}_1 = \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}_2 = \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \quad (94)$$

Higher N $\tilde{\lambda}$'s are tensor products of the above. For example, $\tilde{\lambda}_{12}$ of Equation (81) is clearly given by $\tilde{\lambda}_{12} = \tilde{\lambda}_1 \otimes \tilde{\lambda}_2$. □

For simplicity, we have mainly illustrated the form of the mask averaged λ_I using a recombination distribution associated with equal mask probabilities. The case of uniform crossover also leads to quite simple mask averaged λ_I. Here, in the binary case, the first child gets the i-th bit from the first parent with probability p_i, resulting in the mask probability distribution

$$p(M) = \prod_{i \in M_0} p_i \prod_{j \in M_1} (1 - p_j), \quad (95)$$

where M_α is the subset of binary indices in M with value α. For example, for $N = 2$, we get

$$p_{00} = p_1 p_2, \quad p_{01} = p_1(1 - p_2), \quad p_{10} = (1 - p_1)p_2, \quad p_{11} = (1 - p_1)(1 - p_2).$$

Then the average λ still factorizes, with the string basis 1-bit factors

$$\lambda_0 = \begin{pmatrix} 1 & p_i \\ 1 - p_i & 0 \end{pmatrix}, \qquad \lambda_1 = \begin{pmatrix} 0 & 1 - p_i \\ p_i & 1 \end{pmatrix}, \quad (96)$$

and their monomial basis counterparts

$$\tilde{\lambda}_* = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}_1 = \begin{pmatrix} 0 & 1-p_i \\ p_i & 0 \end{pmatrix}, \tag{97}$$

where i denotes the position of the bit. We see again that $\tilde{\lambda}_{11\ldots1}$ is skew-diagonal. It is easy to see that this generalizes to *any* probability distribution $p(M)$ since $\tilde{\lambda}_{11\ldots1}(M)$ itself is skew-diagonal (a fact that does not depend on $p(M)$) and ${\lambda_I}^{JK}$ is a sum over such matrices. It is equally easy to see that the above results also hold for K-alphabets.

7.5.2 The Origin of Skew-Diagonalization

We have seen above that in the monomial (Building Block) basis the quadratic part of the dynamical equation for recombination only involves pairs of conjugate schemata (see (67), (73), (82)). We examine, in this section, the algebraic origin of this simplification. In particular, we address the following problem: given the form of the recombination operator R_I in the δ-basis, Equations (65), (66), find properties of an invertible transformation matrix $\Lambda_N = \Lambda_1^{\otimes N}$, guaranteeing that the transformed recombination operator $(\Lambda_N^{-1})^T R_I \Lambda_N^{-1}$ assumes the above simplified form. The proposition that follows establishes a sufficient condition: the first row of Λ_1 consists of only 1's.

PROPOSITION: Given $\lambda(M)$, defined as in the first of (65), and an invertible matrix Λ_N, such that $\Lambda_N = \Lambda_1^{\otimes N}$ and ${(\Lambda_1)_0}^i = 1$, $0 \le i \le K-1$. Then in the basis $\mathbf{P'} = \Lambda_N \mathbf{P}$, the recombination equations take the *building-block reduced* (BBR) form

$$P'_I(t+1) = (1-p_c)P_I + \frac{p_c}{2}\sum_M \big(p(M) + p(\bar{M})\big) P_{M_I^{(1)}} P_{M_I^{(2)}}, \tag{98}$$

where $M_I^{(\alpha)} = \big(m_1^{(\alpha)}(I), \ldots, m_N^{(\alpha)}(I)\big)$, $\alpha = 1, 2$, and

$$m_r^{(1)}(I) = \begin{cases} i_r & \text{if } m_r = 0 \\ 0 & \text{if } m_r = 1 \end{cases}, \qquad m_r^{(2)}(I) = \begin{cases} 0 & \text{if } m_r = 0 \\ i_r & \text{if } m_r = 1 \end{cases}. \tag{99}$$

PROOF: From the factorization of the $\lambda(M)$ tensor, Equation (83), we get for the mask-averaged λ,

$${\lambda_U}^{ST} = \sum_M p(M) \prod_{r=1}^{N} \big(\bar{m}_r {\delta_{u_r}}^{s_r} + m_r {\delta_{u_r}}^{t_r}\big). \tag{100}$$

Transforming to the primed basis, as in Equation (64), and using the factorization of the Λ_N matrix, gives

$${(\lambda')_I}^{JK} = \sum_M p(M) \prod_{r=1}^{N} \sum_{u_r, s_r, t_r} \Big(\big(\bar{m}_r {\delta_{u_r}}^{s_r} + m_r {\delta_{u_r}}^{t_r}\big) {(\Lambda_1)_{i_r}}^{u_r} {(\Lambda_1^{-1})_{s_r}}^{j_r} {(\Lambda_1^{-1})_{t_r}}^{k_r} \Big) \tag{101}$$

$$= \sum_M p(M) \prod_{r=1}^{N} \left(\bar{m}_r {\delta_{i_r}}^{j_r} \sum_{t_r} {(\Lambda_1^{-1})_{t_r}}^{k_r} + m_r {\delta_{i_r}}^{k_r} \sum_{s_r} {(\Lambda_1^{-1})_{s_r}}^{j_r} \right). \tag{102}$$

However, for the sum of the elements of a column of Λ_1^{-1} we find

$${\delta_0}^v = \sum_l {(\Lambda_1)_0}^l {(\Lambda_1^{-1})_l}^v = \sum_l {(\Lambda_1^{-1})_l}^v, \tag{103}$$

since $(\Lambda_1)_0{}^l = 1$ for all l. Substituting in (102) gives

$$(\lambda')_I{}^{JK} = \sum_M p(M) \prod_{r=1}^{N} \left(\bar{m}_r \delta_{i_r}{}^{j_r} \delta_0{}^{k_r} + m_r \delta_{i_r}{}^{k_r} \delta_0{}^{j_r} \right). \tag{104}$$

Consider now the factor, in the above product, corresponding to a particular value of r — this is just the 1-Kit transformed $\lambda(m)$ tensor, $(\lambda'(m_r))_{i_r}{}^{j_r k_r}$. Contracting its two upper indices with arbitrary vectors x, y, one gets

$$(\lambda'(m_r))_{i_r}{}^{j_r k_r} x_{j_r} y_{k_r} = \left(\bar{m}_r \delta_{i_r}{}^{j_r} \delta_0{}^{k_r} + m_r \delta_{i_r}{}^{k_r} \delta_0{}^{j_r} \right) x_{j_r} y_{k_r} \tag{105}$$

$$= \bar{m}_r x_{i_r} y_0 + m_r x_0 y_{i_r} \tag{106}$$

$$= \begin{cases} x_{i_r} y_0 & \text{if } m_r = 0 \\ x_0 y_{i_r} & \text{if } m_r = 1 \end{cases}. \tag{107}$$

It is clear then that the terms that appear in the quadratic form $(\lambda')_I{}^{JK} P_J P_K$ contain the indices $(i_1, \ldots, i_N)$ distributed between the two P's, and whenever one P receives an i_r as an index, the other gets a 0, i.e.,

$$(\lambda')_I{}^{JK} P_J P_K = \sum_M p(M) P_{M_I^{(1)}} P_{M_I^{(2)}}, \tag{108}$$

in the notation of (99). Adding the contribution of the transpose of λ, in order to form R_I, one gets the extra term proportional to $p(\bar{M})$ in (98). ■

Notice that the summand in (98) is invariant under $M \to \bar{M}$, so that the $1/2$ factor in front of the sum can be omitted, if the latter is restricted to only half the masks, omitting their conjugates. We find it rather remarkable that the final form of λ', Equation (104), only depends on the fact that $(\Lambda_1)_0{}^i = 1$, and the invertibility of Λ_1, but is otherwise insensitive to the structure of Λ_1. Since the $\tilde{\Lambda}_1$ matrix connecting the string and monomial bases satisfies the above conditions (see Equation (39)), the result (104) holds, in particular, for $\tilde{\lambda}$.

In Section 5.2 we defined the monomial basis for Kit strings as the dual of the Taylor basis. Exploiting the liberty afforded by the above observation about $(\Lambda_1)_0{}^i = 1$, we can define an alternative basis whose elements are purely strings or schemata. This is done using the $K \times K$ transformation matrix, for $N = 1$,

$$\Lambda_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{109}$$

Λ_1^{-1} has the sign of off-diagonal elements flipped. The corresponding transformation matrix for arbitrary N, Λ_N, is found, as before, by just taking the N-fold tensor product of Λ_1 — the corresponding basis elements are purely strings or schemata.

7.5.3 "Zapping"

As mentioned earlier, given the dynamical equation for recombination of a particular string in the δ-basis, a similar equation for any other string can be obtained simply by

renaming of indices. One may wonder how in the BB basis, given the equation for a particular string, one generates equations for the other elements of the basis. As these other basis elements represent schemata one must consider how to coarse grain the string equation to obtain equations for schemata. This coarse graining was carried out explicitly in (Stephens and Waelbroeck 1997; 1999; Stephens, 2001) by directly summing over the allele values of those loci which were not in the corresponding schema partition. The resulting equations were shown to be form invariant. Later, Vose and collaborators (Vose 1999; Vose and Wright 2001) generalized this result to arbitrary cardinality GAs, while in (Vose, Wright and Rowe 2004) it was generalized to a more general class of crossover operators. In the latter works the projection to schemata from strings was achieved by a mask representation that directly mapped from, for example binary strings, C^N to C^{N_M}, where N_M is the subspace picked out by the mask M.

Here we will show how one can pass from strings to schemata without projecting onto a lower dimensional space so that the transformation can be implemented using a square matrix that, importantly, can be written as a bit-wise tensor product. This has the important ramification that all our tensor machinery can still be brought to bear.

In the BB basis, consider schemata I, J, where I can be obtained by coarse graining J (turning non-$*$ indices of J into $*$'s). Then the dynamical equation for I can be obtained from the one for J by an operation which we call "zapping". This consists in summing over all values of a non-$*$ index to turn it into a $*$. We remark here that the process can be formalized further, taking into account the tensor product structure of the matrices involved. Indeed, restricting for the moment to the binary case, the dynamical equation for recombination is governed essentially by the 1-bit λ's, Equation (86), or, more generally, (97). It is now a simple matter to verify that

$$Z^T \tilde{\lambda}_1 Z = \tilde{\lambda}_* \,, \qquad\qquad Z = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} . \tag{110}$$

This fact can be used to obtain $\tilde{\lambda}_I$ from $\tilde{\lambda}_J$ as follows

$$\tilde{\lambda}_I = Z_{IJ}^T \tilde{\lambda}_J Z_{IJ} \,, \qquad\qquad Z_{IJ} = A(i_1, j_1) \otimes \ldots \otimes A(i_N, j_N) \,, \tag{111}$$

where

$$A(i_r, j_r) = \begin{cases} I_2 & \text{if } i_r = j_r \\ Z & \text{if } i_r \neq j_r \end{cases} . \tag{112}$$

In words, two schemas I, J as above, will differ in that some of the 1's of J are replaced by $*$'s in I. This means that, in the corresponding λ-matrices, $\tilde{\lambda}_I$, $\tilde{\lambda}_J$, each of which is a tensor product of $\tilde{\lambda}_1$, $\tilde{\lambda}_*$ factors, the difference lies in that some of the $\tilde{\lambda}_1$ factors in $\tilde{\lambda}_J$ are replaced by $\tilde{\lambda}_*$ factors in $\tilde{\lambda}_I$. The matrix Z_{IJ} defined above implements this substitution by conjugation by a Z factor in just the right places (and a unit matrix factor in the rest). R_I then is obtained by $R_I = Z_{IJ}^T R_J Z_{IJ}$.

Example 12 *Two-bit zapping*

We show here how to derive $\tilde{\lambda}_{1*}$ from $\tilde{\lambda}_{11}$ by zapping the latter. We have (see Equations (84),(86))

$$\tilde{\lambda}_{1*} = \tilde{\lambda}_1 \otimes \tilde{\lambda}_* = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} . \tag{113}$$

Since the schema $(1*)$ only differs from (11) in the second bit, the corresponding zapping matrix is given by

$$Z_{(1*)(11)} = I_2 \otimes Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{114}$$

It is now a matter of straightforward matrix multiplication to verify that $\tilde{\lambda}_{1*} = Z^T_{(1*)(11)} \tilde{\lambda}_{11} Z_{(1*)(11)}$, where $\tilde{\lambda}_{11}$ is given in Equation (72). □

Exactly analogous statements are easily seen to hold true in the K-alphabet case. Taking $J = \mathbb{I} \equiv (11\ldots1)$, we get for the dynamical equation of any schema I in the corresponding basis

$$\begin{aligned} P_I(t+1) &= \mathbf{P}^T Z^T_{I\mathbb{I}} R_{\mathbb{I}} Z_{I\mathbb{I}} \mathbf{P} \\ &= (1-p_c) P_I + p_c \mathbf{P}^T_{I\mathbb{I}} R_{\mathbb{I}} \mathbf{P}_{I\mathbb{I}} , \end{aligned} \tag{115}$$

where $\mathbf{P}_{I\mathbb{I}} \equiv Z_{I\mathbb{I}} \mathbf{P}$. The quadratic term in the r.h.s. of (115) can now be interpreted as the norm squared of the vector $\mathbf{P}_{I\mathbb{I}}$ in the (I-independent) metric $R_{\mathbb{I}}$. Thus, we see that we can generate all the dynamical equations for the Building Block schemata of any string by simply "zapping" the dynamical equation for that string.

Equation (115) exhibits the characteristic form invariance of the dynamical equations under a coarse graining that has been noted previously (Stephens and Waelbroeck, 1999).

8 Conclusions

We presented GA dynamics in a covariant form, showing how different existing formulations — string, Walsh mode, Building Block schemata — can be related by linear coordinate transformations. It was shown that the N-bit transformation matrices are the N-th tensor power of the corresponding 1-bit matrices. The covariance of the dynamical equations guarantees their validity in all bases — nevertheless, the analysis and its interpretation can be greatly simplified by choosing the basis best adapted to the genetic operator under study. The string basis is convenient for selection-dominated dynamics, while the Walsh basis is natural for dynamics dominated by mutation. In this paper we concentrated on the most complicated operator — recombination — showing how the BB basis offered the most natural description, the effective degrees of freedom of recombinative dynamics being Building Block schemata. Introducing a description in terms of characteristic functions in configuration space, we showed that the BB basis is dual to the standard Taylor basis. A thorough analysis of the factorizability of the various operators was given, resulting in an enormous simplification of their calculation in the different bases. Finally, we showed that by "zapping", i.e., coarse graining, the dynamical equation in building block form for an arbitrary string one may generate the dynamical equation for any Building Block schema of that string and thereby the dynamical equation for any other string.

With the unification program for EC in mind, one might wonder if a covariant formulation also exists for GP. Recently, the BB basis has been derived for variable-length GAs and GP (Poli and Stephens, 2005) hence, as more than one coordinate system is of utility there too, it would be of interest to write the dynamics of a GP system in covariant form.

Acknowledgments

CC would like to acknowledge partial financial support from DGAPA UNAM projects IN114302 and IN108103-3 and Conacyt grant 41208-F. CRS thanks the EPSRC for financial support (grant number GR/T24616/01), Conacyt project 30422-E and the UNAM Macroproyecto — "Tecnologías para la Universidad de la Información y la Computación". The research has also been partially supported by the Leverhulme Trust through a visiting professorship to CRS. CRS thanks Riccardo Poli, Jon Rowe and Alden Wright for useful discussions.

References

Akivis, M. and Goldberg, V. (1977). *An Introduction to Linear Algebra and Tensors*. Dover Publications, Mineola, NY.

Bethke, A. (1980). *Genetic Algorithms as Function Optimizers*. PhD thesis, University of Michigan.

Bürger, R. (2000). *The Mathematical Theory of Selection, Recombination, and Mutation*. Wiley, Chichester, UK.

Chryssomalakos, C. and Stephens, C. R. (2004). What basis for genetic dynamics? In *GECCO 2004*, pages 1018–1029.

Goldberg, D. E. (1989a). Genetic algorithms and Walsh functions: Part I. A gentle introduction. *Complex Systems*, 3:123–152.

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

Griffiths, R. C. and Tavaré, S. (1997). Computational methods for the coalescent. In Donnelly, P. and Taveré, S., editors, *Progress in Population Genetics and Human Evolution*, pages 165–182. Springer.

Koehler, G., Bhattarachaya, and Vose, M. (1997). General cardinality genetic algorithms. *Evol. Comp.*, 5:439–459.

Langdon, W. B. and Poli, R. (2002). *Foundations of Genetic Programming*. Springer Verlag, Berlin, New York.

McPhee, N. F. and Crane, E. (2005). A theoretical analysis of the hiff problem. In *GECCO 2005*, pages 1153–1160.

Poli, R. (2000). Exact schema theorem and effective fitness for GP with one-point crossover. In *GECCO 2000*, pages 469–476, Las Vegas. Morgan Kaufmann.

Poli, R. (2001a). Exact schema theory for genetic programming and variable-length genetic algorithms with one-point crossover. *Genetic Programming and Evolvable Machines*, 2(2):123–163.

Poli, R. (2001b). General schema theory for genetic programming with subtree-swapping crossover. In *EuroGP 2001*, LNCS. Springer-Verlag.

Poli, R. and Stephens, C. R. (2005a). The building block basis for genetic programming and variable-length genetic algorithms. *Int. Jou. Comp. Int. Res.*, 1:183–197.

Poli, R. and Stephens, C. R. (2005b). Theoretical analysis of generalised recombination. In Corne, D., editor, *Proceedings of CEC2005*, pages 411–419. IEEE.

Reeves, C. R. and Rowe, J. E. (2002). *Genetic Algorithms: Principles and Perspectives*. Kluwer, USA.

Rowe, J., Vose, M., and Wright, A. (2004). Structural search space and genetic operators. *Evol. Comp.*, 12:461–494.

Schmitt, L. (2004). Theory of genetic algorithms ii: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoretical Computer Science*, 310:181–231.

Stephens, C. (2002). The renormalization group and the dynamics of genetic systems. *Acta Phys. Slov.*, 52:515–524.

Stephens, C. R. (2001). Some exact results from a coarse grained formulation of genetic dynamics. In *GECCO 2001*, pages 631–638. Morgan Kaufmann.

Stephens, C. R. and Poli, R. (2004). E C theory - in theory: Towards a unification of evolutionary computation theory. In Menon, A., editor, *Frontiers of Evolutionary Computation*, pages 129–156. Kluwer Academic Publishers.

Stephens, C. R. and Waelbroeck, H. (1997). Effective degrees of freedom of genetic algorithms and the block hypothesis. In Bäck, T., editor, *Proceedings of ICGA97*, pages 31–41, San Francisco, CA. Morgan Kaufmann.

Stephens, C. R. and Waelbroeck, H. (1999). Schemata evolution and building blocks. *Evol. Comp.*, 7:109–124.

Stephens, C. R. and Zamora, A. (2003). EC theory: A unified viewpoint. In Paz, E. C., editor, *GECCO 2003*, pages 1394–1402, Berlin, Germany. Springer Verlag.

Toussaint, M. (2004). Notes on information geometry and evolutionary processes. *CoRR*, nlin.AO/0408040.

Vose, M. D. (1999). *The simple genetic algorithm: Foundations and theory*. MIT Press, Cambridge, MA.

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.

Vose, M. D. and Wright, A. H. (2001). Form invariance and implicit parallelism. *Evolutionary Computation*, 9(3):355–370.

Weinberger, E. (1991). Fourier and taylor series on fitness landscapes. *Biol. Cybern.*, 65:321–330.

Wright, A. H. (2000). The exact schema theorem. http://www.cs.umt.edu/u/wright/papers/schema.pdf.