Coarse Grained Dynamics for Generalized Recombination

Christopher R. Stephens
Department of Computer Science
University of Essex, Wivenhoe, CO4 3SQ, UK

Instituto de Ciencias Nucleares, UNAM
A. Postal 70-543, México, D.F. 04510

Riccardo Poli
Department of Computer Science
University of Essex, Wivenhoe, CO4 3SQ, UK
rpoli@essex.ac.uk

Department of Computer Science
University of Essex
Technical Report CSM-432
ISSN: 1744-8050
June 2005

Abstract

An exact microscopic model for the dynamics of a genetic algorithm with generalized recombination is presented. Generalized recombination is a new form of exchange of genetic material from parents to offspring that generalizes and subsumes standard operators, such as homolgous crossover, inversion and duplication, and in which a particular gene in the offspring may originate from *any* parental gene. It is shown that the dynamics naturally coarse grains, the appropriate effective degrees of freedom being schemata that act as building blocks. It is shown that the schema dynamics has the same functional form as that of strings and derive a corresponding Exact Schema theorem. To exhibit the qualitatively new phenomena that can occur in the presence of generalized recombination, and to understand the biases of the operator, we derive a complete, exact solution for a two-locus model without selection, showing how the dynamical behaviour is radically different to that of homologous crossover. Inversion is shown to potentially introduce oscillations in the dynamics, while gene duplication leads to an asymmetry between homogeneous and heterogeneous strings. All non-homologous operators lead to allele "diffusion" along the chromosome. We discuss how inferences from the two-locus results extend to the case of a recombinative GA with selection and more than two loci providing evidence from an integration of the exact dynamical equations for more than two loci.

1 Introduction

Coarse-grained formulations of the dynamics of evolutionary algorithms (EAs) offer many advantages relative to "microscopic" ones based purely on the string/chromosome degrees of freedom. These benefits have been exhibited in both the standard genetic algorithm (GA) [27], and in variable-length GAs/linear Genetic Programming (GP) and GP itself [16]. The main advantage is the simpler and deeper understanding of the role of homologous recombination they provide [27, 16], wherein the most appropriate effective degrees of freedom for describing recombination are not strings/chromosomes, but coarse-grained "building blocks", with which the EA builds optimal solutions. The form that these building blocks takes depends on the representation used. For

instance, in GAs they are a particular subset of schemata that form an alternative and more appropriate basis - the Building Block Basis (BBB) [3]. In the case of variable-length strings and trees, they are generalisations of those found in fixed length GAs - Building Block Hyperschemata [16].

Additionally, coarse grained formulations have shown a theoretical commonality between GAs and GP that was not previously apparent, thus leading to a unification of the theoretical underpinnings of both areas. This unification has been chiefly achieved by recognizing that, in the presence of recombination, the underlying dynamical effective degrees of freedom are not the microscopic ones - strings or program trees - but rather schemata or hyperschemata and, in particular, not arbitrary schemata, but a priviliged subset - Building Block schemata - that the EA processes, forming higher order Building Block schemata by recombining corresponding lower order ones. In this sense, coarse grained formulations have led to a much deeper understanding of approximate Schema theorems, such as Holland's [10], and the Building Block Hypothesis [9]. As coarse grained formulations are also exact they have also served as a bridge between this latter work and dynamical systems models [30].

Coarse grained formulations up to now have been studied for GAs and GP with both homologous and subtree crossover and, for GAs, in the presence of standard point mutation. In nature however, there are many more ways of combining parental genetic material into an offspring than homologous crossover, many of which have been used in EAs and been found to be useful by pracititioners. Gene duplication, for example, has been studied in biology [4] as well as in the context of GAs [21] and GP [11], while inversion was one of the operators used by Holland [10] in the original formulation of the GA. Additionally, there is little to no theoretical analysis in the EC literature concerning inversion and duplication, at least not based on an underlying exact dynamical model. The very fact that these operators exist in nature makes manifest their importance in the evolutionary process and, hence, their potential utility in evolutionary computation.

In this paper[1] the notion of generalized recombination is introduced, which can account for *any* redistribution of parental genes into the offspring. This requires the generalisation of the concept of a crossover mask to that of a Generalised Crossover Mask (GCM), with an associated Generalised Recombination Distribution (GRD). Generalised recombination generalises and subsumes many other common genetic operators, including homologous crossover and inversion, as well as gene duplication and deletion. Thus, by studying this more general operator theoretically, we are simulataneously developing a framework within which several different familiar and used operators can be studied. Generalized recombination can also lead to qualititatively new phenomena that are not present in the case of homologous crossover, such as periodic oscillations in the dynamics of strings and schemata in the presence of inversion, and a preference for homogeneous strings and schemata over heterogeneous ones in the presence of duplication, e.g. for two loci a potential preference for 11 over 10 even if they are equally fit.

We study generalised recombination first in the context of a variable-length representation with mutation, deriving an exact dynamical systems type model for describing it. More importantly however, we present a detailed analysis of how the dynamics of generalised recombination, as in the case of homologous crossover, is much more naturally represented in terms of Building Block schemata, as opposed to strings. The difference in this case is that, due to the presence of operators other than homologous recombination, a richer diversity of building blocks enters into the dynamics. We show that the dynamics is form invariant under a coarse graining when passing from strings to schemata, and hence derive an Exact Schema theorem for a fixed-length GA evolving in the presence of mutation and generalised recombination. Interpreted in terms of the concept of *effective* fitness [26, 24, 25, 23], this Exact Schema theorem states that, as in the case of homologous crossover, those schemata which are more effectively fit propagate preferentially. This coarse grained formulation of generalised recombination, as in previous analyses, offers a further theoretical unification for GAs, showing that building blocks naturally appear also in the context of a GA where a gene of the offspring is derived from any of the parental genes.

After introducing the Exact Schema theorem for generalised recombination and mutation, we focus our attention on a detailed analysis of a two-locus model without selection in the infinite

[1]This paper is an extension of earlier, preliminary work [19, 29] presented at CEC 2005.

population limit. Of course, one might question to what extent a two-locus model can illuminate the more complicated multi-locus case. It is wise to remember however, that in population biology, such models have played a crucial role, permitting the qualitative, and sometimes quantitative, analysis of a host of important phenomena (see for instance, [1] and references therein). Even in EC, such models have made important appearances, such as in the deceptive two-bit problem [6], and in previous analyses of the effects of recominbation and mutation [22]. The model we will present has the advantage of being exactly soluble, while at the same time being quite transparent. Additionally, all the interesting phenomena observed in this two-locus model have been shown to be also present in the case of multi-locus models with selection, as we explicitly demonstrate by considering some three-locus results with different fitness landscapes. One may also question the relevance of an infinite population model. The relevance of the infinite population model for finite population dynamics has been discussed extensively in the context of the canonical GA [30]. The same arguments apply here. In particular, for a population of size n_p, one expects finite population effects to be $O(n^{-1/2})$ and, hence, small in the large population limit. Additionally, many of the elements we discuss here also enter into a formulation of the dynamics in terms of a Markov chain, which is the appropriate general framework for the finite population model.

2 Generalised Recombination

In nature there are a multitude of ways of distributing genetic material from "parents" to "offspring". Some involve two parental chromosomes, such as "homologous" recombination and translocation - the latter being the breakage and removal of a large segment of DNA from one chromosome, followed by the segment's attachment to a different chromosome. Others involve only one parental chromosome, such as inversion, duplication and deletion. "Homologous" in biology can have different meanings. In the context of meiosis [12] in diploids, it refers to the recombination that takes place between "homologous" pairs of chromosomes, an homologous pair being such that the ith locus in each member of the pair codes for the same gene, even though the particular allele might be different, e.g. green eyes versus blue eyes as opposed to green eyes and brown hair. It can also refer, however, to the fact that a subset of genes or nucleotides in a pair have the same structure and hence might serve as a preferred point at which exchange of genetic material can take place. Translocation is of this type, and is often termed "unequal" crossing over in evolutionary computation in order to distinguish it from the more familiar homologous crossover, where parental chromosomes are first "aligned" so that homologous genes are in corresponding positions. Almost by definition, i.e. they exist in nature, all these operators have played an important role in evolution, though naturally their relative importance depends on the type of organism. For instance, homologous recombination seems to be more important in diploids than in simpler organisms such as bacteria and viruses [12].

The most well known operator for transferring genetic material in GAs, in the context of a fixed-length representation, is homologous recombination, $\mathcal{H}$, modelled on meiosis [12], where alleles at a particular locus have their origin in the corresponding genetic loci of the parental chromosomes. n-point and uniform crossover are of this type. Such recombination can be succinctly modelled using the concept of a recombination mask, $\mathbf{m}$, which, for strings of length ℓ, can be represented by an ℓ-dimensional vector $\mathbf{m} = (m_1, m_2, \ldots, m_\ell)$, where $m_i = 0,\ 1$ indicates from which parent the ith allele is taken - 0 meaning take it from the ith locus of the first parent, and 1, from the ith locus of the second parent. The total number of possible masks is 2^ℓ. Associated with them is a recombination distribution, denoted by $p_c(\mathbf{m})$. If the probability to perform crossover is p_{xo}, then $p_c(\mathbf{m})$ is the conditional probability for choosing the mask $\mathbf{m}$ *given* that crossover was implemented. Hence, $\sum_{\mathbf{m}} p_c(\mathbf{m}) = 1$ and $p_{xo} \times p_c(\mathbf{m})$ is the probability to crossover using the mask $\mathbf{m}$. It is the choice of $p_c(\mathbf{m})$ that specifies if one is considering one-point, two-point, uniform crossover etc..

2.1 Generalised Recombination Masks and Distributions

Although, binary masks are sufficient to model homologous genetic operators in a fixed length setting, to describe more general ones, where an allele in a particular locus of the offspring comes from a different locus in either one of the parents, a new level of generality is required. To represent this, the analog of a crossover mask is required that explicitly specifies which alleles from the parents are to form the offspring and in which order.

To specify the arbitrary redistribution of genetic material from parents to offspring a generalisation of the concept of crossover mask is needed. We will term such a generalisation - a *Generalised Crossover Mask* (GCM). The associated probability distribution over the GCMs generalises the concept of a recombination distribution and will be termed a *Generalised Recombination Distribution* (GRD). A GCM can be specified mathematically using several, equivalent representations. If we consider the more general case of variable-length strings, in order to be able to account for unequal crossing over, as well as homologous crossover, a GCM has to specify how ℓ alleles, for an offspring of length ℓ, are obtained from two parents of lengths ℓ_1 and ℓ_2 respectively.

The closest representation of a GCM to a standard crossover mask we call a recombination vector, $\mathbf{v}$. As the ith allele in the offspring could come from any locus in the parents, the components of this vector can take values from the set $\mathcal{N}_{\ell_1+\ell_2} = \{1, \cdots, \ell_1 + \ell_2\}$, values from 1 to ℓ_1 denoting that the allele originated in the first parent, and values between ℓ_1+1 and $\ell_1+\ell_2$ signifying that it came from the second. Thus, in this representation we denote a GCM by $\mathbf{v}(\ell, \ell_1, \ell_2) = (v_1, \cdots, v_\ell)$, whose components $v_i \in \mathcal{N}_{\ell_1+\ell_2}$. Thus, for example, for $\ell = 3$, $\mathbf{v}(3, 2, 4) = (1, 5, 3)$ represents a GCM that takes genetic material from parents of lengths 2 and 4 respectively and recombines it into an offspring of length 3. The notation $(1, 5, 3)$ for the components of $\mathbf{v}(3, 2, 4)$ signifies that the first gene of the offspring came from the first gene of the first parent, the second gene from the third gene of the second parent and the last from the first of the second parent. As the "mask alphabet" is of cardinality $\ell_1 + \ell_2$, rather than two, as in the case of normal crossover masks, the total number of GCMs associated with $\mathbf{v}(\ell, \ell_1, \ell_2)$ is $(\ell_1 + \ell_2)^\ell$. For arbitrary string lengths, the total number of possible GCMs is, of course, infinite. The associated distribution of probabilities, $p_c(\mathbf{v}(\ell, \ell_1, \ell_2))$, then determines the GRD.

For the case of a fixed-length representation, we will drop the string length arguments, as the unique string length is specified by the dimensionality of the vector. Thus, in this case, the recombination vector $\mathbf{v} = (1, 5, 3)$, for strings of $\ell = 3$, signifies that the first gene of the offspring came from the first gene of the first parent, the second gene from the second of the second parent and the last from the last of the first parent. In this case the total number of recombination vectors is $(2\ell)^\ell$.

Another equivalent representation for GCMs uses arrays (crossover matrices) instead of bit strings (vectors) to represent recombination events. An appropriate crossover matrix for representing the formation of a length ℓ offspring from length ℓ_1 and ℓ_2 parents will have ℓ rows and $(\ell_1 + \ell_2)$ columns. The first ℓ_1 columns indicate which alleles are copied from the first parent, while columns ℓ_1+1 through $\ell_1+\ell_2$ indicate what is provided by the second. The matrix elements are either 0 or 1, where a 1 in row r and column c means that locus r in the offspring is filled with the allele from locus c in the first parent if $c \leq \ell_1$. If $c > \ell_1$ it is filled with the allele from locus $c - \ell_1$ of the second parent. Because an offspring would not be fully specified if some of its alleles were undefined, or would be overly specified if we tried to place more than one allele in a locus, in each row of a crossover matrix there must be exactly one 1 (with all other elements in the row being 0). For a fixed length representation the matrices are of constant size $\ell \times 2\ell$.

Finally, another useful representation is that of a *recombination pair*, $\mathbf{r}(\ell, \ell_1, \ell_2) \equiv (\mathbf{m}, \mathbf{v})$, which is a hybrid between the notion of a standard crossover mask and a recombination vector with a reduced cardinality alphabet. Here, $\mathbf{m} = (m_1 \cdots m_\ell)$ is an ℓ-dimensional vector (i.e., $m \in \{0, 1\}^\ell$) whose components specify which parent contributes the alleles to fill each locus in the offspring. So, for example, $m_i = 0$ means locus i will be filled with an allele from parent 1, while $m_i = 1$ means parent 2 will contribute the allele instead. Having specified from which parent a particular allele originates, one must specify from which locus. This is achieved by specifying $v(\ell, \ell_1, \ell_2) = (v_1, \cdots, v_\ell)$, an ℓ-component vector of integers with components $v_i \in \{1, \cdots, \ell_1\}$ if

$m_i = 0$ (i.e., $v \in \mathcal{N}_{\ell_1}$) and $v_i \in \{1, \cdots, \ell_2\}$ if $m_i = 1$. The elements of $\mathbf{v}(\ell, \ell_1, \ell_2)$ specify which alleles from a particular parent will be transferred to the offspring. In the case of a recombination pair, for variable-length strings, the cardinality of the alphabet from which the v_i are taken depends on the corresponding m_i. Hence, this representation is less convenient than that of a recombination vector for variable-length strings. For fixed-length strings however, it is perfectly natural, and actually presents several advantages relative to the recombination vector notation. In the recombination pair notation homologous recombination for fixed-length strings can be represented by pairs of the form $\mathbf{r} = (m, (1, 2, \cdots, \ell))$ where, effectively, m can be seen as a standard crossover mask.

As an example of how the different representations of a GCM work consider standard one-point crossover for $\ell = 3$. The associated crossover matrices are

$$\begin{pmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1 \end{pmatrix} \quad \begin{pmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \end{pmatrix}$$

each invoked with probability $\frac{1}{2}$. These are equivalent to the recombination vectors $v_1 = (1, 5, 6)$ and $v_2 = (1, 2, 6)$, or to the recombination pairs $r_1 = (011, (1, 2, 3))$ and $r_2 = (001, (1, 2, 3))$, or to the more traditional crossover masks 011 and 001.

To see the large variety of ways in which parental genetic material can be distributed among the offspring consider the case of $\ell = 2$, where the $(2 \times 2)^2 = 16$ recombination pairs are

(00,(1,1))	(00,(1,2))	(00,(2,1))	(00,(2,2))
(01,(1,1))	(01,(1,2))	(01,(2,1))	(01,(2,2))
(10,(1,1))	(10,(1,2))	(10,(2,1))	(10,(2,2))
(11,(1,1))	(11,(1,2))	(11,(2,1))	(11,(2,2))

If the associated GRD is such that each is invoked with probability $p_c(m, v) = \frac{1}{16}$, this would represent a recombination operator where each locus in the offspring is filled with a randomly chosen allele from the parents. Clearly this operator could not be represented with standard crossover masks. As a final example, the following GRD represents a single-parent inversion operator in the case of a three-locus system:

$$\begin{aligned} p_c(111, (2, 1, 3)) &= p_c(111, (1, 3, 2)) \\ &= p_c(111, (3, 2, 1)) = \frac{1}{3} \end{aligned}$$

Although, for formal manipulations we will use the more powerful recombination pair notation, as we will treat the two-locus case extensively in this paper, we will represent the GRD there using the recombination vectors notation. The GRD then is a vector with components p_{ab}, where the indices a and b take values from one to four, one and two corresponding to the first and second loci of the first parent and three and four the corresponding loci of the second parent. Thus, for example p_{13} gives the probability for applying the GCM associated with finding the first locus of the offspring from the first locus of the first parent and the second locus from the first locus of the second parent.

2.2 Genetic Operators as GCMs

So, how are different genetic operators represented by GCMs? We will here discuss this question in the context of fixed-length strings, there being an analogous extension to the case of variable-length strings. In generalised recombination a genetic locus in the offspring can originate from any locus of the parents. Does this mean that generalised recombination should be considered as representing a completely new type of genetic operator? Or can it be represented by other more familiar and basic operators? This can best be analysed in terms of the recombination pair representation of a GCM.

There are three principal components involved in the generalised recombination operator $\mathcal{G}$: homologous crossover $\mathcal{H}$, permutations $\mathcal{P}$ and duplications $\mathcal{D}$. The operator $\mathcal{G} : \Omega \times \Omega \to \Omega$, where Ω is the ℓ-dimensional space of string configurations. Similarly, $\mathcal{H} : \Omega \times \Omega \to \Omega$, while $\mathcal{P} : \Omega \to \Omega$ and $\mathcal{D} : \Omega \to \Omega$. To understand the relation between $\mathcal{G}$ and $\mathcal{H}$, $\mathcal{P}$ and $\mathcal{D}$ further, we introduce the following notation: a superscript ℓ will refer to strings of length ℓ. Thus, $I^\ell \in \Omega$ will refer to the genotype of a string of length ℓ. I^ℓ can also be thought of as a multi-index, or a vector - $I^\ell = (I^\ell_1, I^\ell_2, \ldots, I^\ell_\ell)$, referring to the alleles of the ℓ loci of a string of length ℓ.

Now, $\mathcal{G}$ and $\mathcal{H}$ are maps from string pairs (J, K) to a target string, i.e. $I^{\mathcal{G}_\alpha} = \mathcal{G}_\alpha(J, K)$ and $I^{\mathcal{H}_\alpha} = \mathcal{H}_\alpha(J, K)$, where α indexes the particular map chosen, there being one map for every mask in the case of homologous crossover, or GCM for generalised recombination. The actual output string depends on the map chosen. For homologous crossover there are 2^ℓ such maps, while for generalised recombination there are $(2\ell)^\ell$. We take the maps to be deterministic, i.e. for a given α $\mathcal{G}$ returns I, given parents (J, K), with probability one. However, the probability for applying the map, $p(\alpha)$ is determined by the recombination distribution or GRD. The actions of the different operators can be related to the particular properties of the possible output strings, which in their turn depend on the α. For homologous recombination α is just the mask m. In order to give a uniform treatment we can also represent this standard mask in the more general recombination pair notation $\mathbf{r}_\mathcal{H} = (m, v)$, where m is the standard mask and v is an ordered permutation list, i.e. $v_1 < v_2 < \cdots < v_\ell$ and $v_i \neq v_j \ \forall i, \ j$, i.e. $v = (1, 2, \cdots, \ell)$. Introducing the following function

$$a \bullet b = \begin{array}{ll} b & \text{if } a = 1 \\ * & \text{otherwise,} \end{array} \tag{1}$$

where $*$ is the standard "wildcard" symbol used for defining schemata, we can represent the components I_i of the output string that arises due to homologous recombination as

$$I_i^{\mathcal{H}_m} = (\bar{m}_i \bullet J_i) \cap (m_i \bullet K_i) \tag{2}$$

where $\bar{m}_i = (1 - m_i)$. Equation (2) states that: if $m_i = 0$, then the ith locus of the offspring string, $I^{\mathcal{H}_m}$, is taken from the ith locus of the first parent, while, if $m_i = 1$, it is taken from the ith locus of the second parent. Thus, we see how the offspring gene $I_i^{\mathcal{H}_m}$ is inherited from the parents.

A permutation can then be applied to $I^{\mathcal{H}_m}$. The permutation $\mathcal{P}_\alpha$ is associated with only the v part of the recombination pair $\mathbf{r}$, α denoting the explicit permutation map applied. Its action on v is: $v \to v^p$. For instance, for $\ell = 3$, if $v = (1, 2, 3)$ and $\mathcal{P}_\alpha$ represents an inversion of the entire string then $v^p_1 = v_3$, $v^p_2 = v_2$ and $v^p_3 = v_1$. At the level of the string, the components of an offspring $I^{\mathcal{P}_p}$ arise from those of a parent J by mapping the components: $I_i^{\mathcal{P}_p} = J_{v_i^p}$. For example, for $v^p = (3, 2, 1)$, $I_1^{\mathcal{P}_p} = J_3$, $I_2^{\mathcal{P}_p} = J_2$ and $I_3^{\mathcal{P}_p} = J_1$.

The composite action of homologous recombination and then permutation can be represented as $I^{\mathcal{P}_p\mathcal{H}_\alpha} = \mathcal{P}_p\mathcal{G}_\alpha(J, K)$. Explicitly, in terms of components,

$$I_i^{\mathcal{P}_p\mathcal{H}_m} = (\bar{m}_i \bullet J_{v_i^p}) \cap (m_i \bullet K_{v_i^p}) \tag{3}$$

The difference with (2) is that J_{v_i} and K_{v_i} did not necessarily originate at the ith locus of the parents. Note that v^p remains a permutation list, i.e. $v_i \neq v_j \ \forall i, j$. This is no longer true when duplication is involved. Like permutations, duplications affect only the v part of $\mathbf{r}$. The action of $\mathcal{D}$ on v is $v \to v^d$ where $v^d = (v^d_1, v^d_2, \cdots, v^d_\ell)$ is such that one can now have $v^d_i = v^d_j$ for some values of i and j. Equation (3) serves to cover this situation too, the difference now being that some parental loci J_{v_i} or K_{v_i} may appear more than once in the offspring. By the actions of the three operators: $\mathcal{H}$, $\mathcal{P}$ and $\mathcal{D}$ it is possible to represent any redistribution of loci from parents to offspring. Thus, the generalised recombination operator can be written in the form $\mathcal{G}_\mathbf{r} = \mathcal{D}_{v^d}\mathcal{P}_{v^p}\mathcal{H}_m$ which, acting on strings gives $I^{\mathcal{G}_\mathbf{r}} = \mathcal{D}_d\mathcal{P}_p\mathcal{G}_\alpha(J, K)$ whose explicit action component-wise is given by (3). Thus, we see how generalised recombination acts at the level of strings viewed as points or vectors in the configuration space or as multi-indices. Equation (3)

shows how an offspring gene $I_i^{\mathcal{G}_\alpha}$ is inherited from one of the parental genes J_{v_i} or K_{v_i}. Note that the inheritance of another offspring gene $I_j^{\mathcal{G}_\alpha}$ is statistically independent of the inheritance of $I_i^{\mathcal{G}_\alpha}$. This is a consequence of the fact that recombination is implemented using a mask m and a vector v, that are exogeneously specified, and are independent of the allelic content of both the parental and offspring strings.

Inversions, being a subset of permutations, can be realised by permutations of some or all of the elements of the string ordering $v = (v_1, \ldots, v_\ell)$. For example, for $\ell = 3$, $(001, (2, 1, 3))$ represents a GCM where the first gene of the offspring came from the second gene of the first parent, the second gene from the first gene of the first parent and the last from the last gene of the second parent. Thus the first and second offspring genes are now in inverted order. There are two forms of duplication: one parent duplication - duplication from the same locus in the same parent - and two-parent duplication - duplication from the same locus but in different parents. The former is manifest in the corresponding crossover pair by the repetition of an element of v, e.g. $(001, (1, 1, 3))$ gives an offspring where both the first and second genes came from the first locus of the first parent. Duplication from different parents can be seen in the recombination pair $(011, (1, 1, 3))$, where the second and third genes of the offspring now come from the first gene of the first parent and the first gene of the second parent respectively.

3 Evolution Equation in the String Basis

We first write down the exact equations for a finite population of variable-length strings in the string basis, with selection, mutation and generalised recombination. $P_{I^\ell}(t)$ will refer to the proportion in the population of genotypes of type I in strings of length ℓ at generation t. With this notation in hand the equation for $E(P_{I^\ell}(t+1))$, the *expected* proportion of genotype I in strings of length ℓ in generation $t+1$ given $P_{I^\ell}(t)$ for all ℓ, is

$$\begin{aligned} E(P_{I^\ell}(t+1)) = & \sum_{J^\ell} {M_{I^\ell}}^{J^\ell} [(1-p_{xo}) P'_{J^\ell}(t) \\ + \quad & p_{xo} \sum_{\ell_1 \ell_2} \sum_{\mathbf{r}(\ell,\ell_1,\ell_2)} p_c(\mathbf{r}(\ell,\ell_1,\ell_2)) \sum_{K^{\ell_1}} \sum_{L^{\ell_2}} {\lambda_{J^\ell}}^{K^{\ell_1} L^{\ell_2}}(\mathbf{r}(\ell,\ell_1,\ell_2)) P'_{K^{\ell_1}}(t) P'_{L^{\ell_2}}(t)] \end{aligned} \tag{4}$$

where P'_{I^ℓ} is the probability to select I^ℓ, and ${\lambda_{J^\ell}}^{K^{\ell_1} L^{\ell_2}}(\mathbf{r}(\ell,\ell_1,\ell_2))$ is the conditional probability that offspring I^ℓ is formed, given parents K^{ℓ_1} and L^{ℓ_2} and a GCM $\mathbf{r}(\ell,\ell_1,\ell_2)$. ${\lambda_{J^\ell}}^{K^{\ell_1} L^{\ell_2}}(\mathbf{r}(\ell,\ell_1,\ell_2)) = 0,\ 1$, as either the offspring is formed or it isn't. Finally, mutation is considered to act here in the standard point-wise fashion wherein the mutation matrix has elements ${M_{I^\ell}}^{J^\ell} = p_m^{d_{I^\ell J^\ell}} (1-p_m)^{\ell - d_{I^\ell J^\ell}}$, where $d_{I^\ell J^\ell}$ is the Hamming distance between I^ℓ and J^ℓ and p_m is the mutation rate. This represents the probability that the string J^ℓ mutates into a string I^ℓ. The notation P'_{I^ℓ} accounts for any selection mechanism. However, the relation between P'_{I^ℓ} and P_{I^ℓ} depends on the specifics of the selection operator. For instance, for proportional selection $P'_{I^\ell}(t) = (f_I/\bar{f}(t)) P_{I^\ell}(t)$, where $\bar{f}(t)$ is the average population fitness. The first term in (4) arises from the cloning of J^ℓ and its subsequent mutation, while the second term represents all the ways in which J^ℓ may be constructed from other strings via generalised recombination, including construction from strings of different lengths, and subsequently mutated.

In the case of fixed-length strings ${\lambda_{J^\ell}}^{K^{\ell_1} L^{\ell_2}}(\mathbf{r}(\ell,\ell_1,\ell_2)) \propto {\delta_\ell}^{\ell_1} {\delta_\ell}^{\ell_2}$, where δ is the Kronecker-delta with the property ${\delta_i}^j = 1$ for $i = j$ and 0 otherwise, and so the sums over ℓ_1 and ℓ_2 disappear. Hence, (4) simplifies to[2]

$$E(P_I(t+1)) = \sum_J {M_I}^J [(1-p_{xo}) P'_J(t) + p_{xo} \sum_{\mathbf{r}} p_c(\mathbf{r}) \sum_K \sum_L {\lambda_J}^{KL}(\mathbf{r}) P'_K(t) P'_L(t)] \tag{5}$$

where we have now dropped the redundant ℓ superscript.

[2]Note that the arrangement of indices I, J, K and L leave (5) in "covariant" form [3] wherein its content is the same in any coordinate basis be it string, Walsh or Building Block basis.

Equations (4) and (5) describe the dynamics of genetic systems that recombine genetic material from two parents in a most general way, ranging from recombination that respects genetic locus, i.e. J_i comes from K_i or L_i, such as is the case for a standard homologous recombination mask, to a repeated ℓ-fold duplication of one particular allele associated with a particular genetic locus in a particular parent. Equation (4) extends known results for exact evolution equations for the canonical GA [30], and for variable-length GAs with either homologous [18, 16] or subtree-type [17] crossover, to the more general class of variable or fixed-length strings and generalised recombination.

The most complicated elements of equation (4) are the ${\lambda_{J^\ell}}^{K^{\ell_1}L^{\ell_2}}$. Although we have used above the recombination pair notation, $\mathbf{r}$, for the GCMs, the representation is at this point arbitrary, recombination vectors or matrices being equally valid. When we come to write an explicit representation for the ${\lambda_{J^\ell}}^{K^{\ell_1}L^{\ell_2}}$ however, the particular form it takes depends on how one represents the GCMs.

As mentioned in section 2.2, inheritance via recombination can be understood by considering it gene by gene, how gene J_i is inherited via recombination being independent of how another gene J_j is inherited. Thus, one may write

$$\lambda_{J^\ell}{}^{K^{\ell_1}L^{\ell_2}}(\mathbf{r}(\ell,\ell_1,\ell_2)) = \prod_{i=1}^{\ell} \lambda_{J_i^\ell}{}^{K_i^{\ell_1}L_i^{\ell_2}}(r_i(\ell,\ell_1,\ell_2)) \tag{6}$$

The meaning of this is that: the conditional probability that generalised recombination, using a recombination pair $\mathbf{r}$ and acting on two parents K^{ℓ_1} and L^{ℓ_2}, gives rise to an offspring J^ℓ is a product of probabilities, i.e. each gene assignment is independent of the rest. Each component in this product is also a conditional probability, being the conditional probability that the locus, i, in the offspring arises from the parental locus determined by $r_i = (m_i, v_i)$, in the recombination pair representation, or v_i, in the recombination vector representation. The explicit form of the ${\lambda_{J_i^\ell}}^{K^{\ell_1}L^{\ell_2}}(r_i(\ell,\ell_1,\ell_2))$ depends on how the GCM is represented, as we will now see.

In section 2.2, (3), based on the function (1), was used to show how generalised recombination acted at the level of strings viewed as multi-indices or vectors. Here, we now have to understand how it works at the level of the dynamical evolution equations which are associated with the string proportions P_I and involve string sums $\sum_I$. In this case it is natural to represent inheritance of a particular allele, I_i^ℓ, at a given genetic locus, i, in the offspring, I^ℓ, from the parents, J^{ℓ_1} and K^{ℓ_2}, via a recombination vector $\mathbf{v}(\ell,\ell_1,\ell_2)$, using the function ${\delta_{I_i^\ell}}^{J^{\ell_1}\otimes K^{\ell_2}(v_i(\ell,\ell_1,\ell_2))}$, where ${\delta_i}^j$ is the Kronecker delta: ${\delta_i}^j = 1$ if $j = i$ and zero otherwise. The meaning of this is that the ith locus of the offspring of genotype I is inherited from the locus associated with the ith component of the recombination vector $\mathbf{v}(\ell,\ell_1,\ell_2)$, $J^{\ell_1}\otimes K^{\ell_2}$ representing the $(\ell_1+\ell_2)$-dimensional vector whose first ℓ_1 components represent the loci of the first parent, J^{ℓ_1}, and the second ℓ_2 components those of the second parent, K^{ℓ_2}. As the components of $\mathbf{v}(\ell,\ell_1,\ell_2)$ range over $\mathcal{N}_{\ell_1+\ell_2}$, $J^{\ell_1}\otimes K^{\ell_2}(v_i(\ell,\ell_1,\ell_2))$ picks out the corresponding component of $J^{\ell_1}\otimes K^{\ell_2}$. For instance, for a two-locus offspring, resulting from a crossover of a two-locus and a three-locus parent via a recombination vector $\mathbf{v}(2,2,3) = (2,3)$, then ${\delta_{I_1^2}}^{J^2\otimes K^3(v_1)} = {\delta_{I_1^2}}^{J_2^2}$, i.e. the first bit of the offspring was inherited from the second locus of the first parent, while ${\delta_{I_2^2}}^{J^2\otimes K^3(v_2)} = {\delta_{I_2^2}}^{K_1^3}$, i.e. the second bit of the offspring was inherited from the first bit of the second parent.

In the recombination pair notation, inheritance of the ith gene via a recombination pair $\mathbf{r} = (m,v)$ can be represented by the function $((1-m_i){\delta_{I_i^\ell}}^{J_{v_i}^{\ell_1}} + m_i{\delta_{I_i^\ell}}^{K_{v_i}^{\ell_2}})$. In this case, m_i specifies from which parent the ith locus will be chosen, while J_{v_i} picks out from the first parent the allele corresponding to v_i and K_{v_i} picks it from the second.

Using recombination vectors, then

$$\lambda_{J_i^\ell}{}^{K_i^{\ell_1}L_i^{\ell_2}}(v_i(\ell,\ell_1,\ell_2)) = {\delta_{J_i^\ell}}^{K^{\ell_1}\otimes L^{\ell_2}(v_i(\ell,\ell_1,\ell_2))} \tag{7}$$

whereas using recombination pairs

$$\lambda_{J_i^\ell}{}^{K_i^{\ell_1} L_i^{\ell_2}}(r_i(\ell,\ell_1,\ell_2)) = ((1-m_i)\delta_{J_i^\ell}{}^{K_{v_i}^{\ell_1}} + m_i \delta_{J_i^\ell}{}^{L_{v_i}^{\ell_2}}) \tag{8}$$

As an example, consider for fixed length strings with $\ell = 3$ the recombination vector $\mathbf{v} = (3,1,5)$, and the corresponding recombination pair $\mathbf{r} = (001(3,1,2))$. In this case

$$\begin{aligned} \lambda_J{}^{KL}((3,1,5)) &= \delta_{J_1}{}^{K_3}\delta_{J_2}{}^{K_1}\delta_{J_3}{}^{L_2} & (9) \\ \lambda_I{}^{JK}((001,(3,1,2))) &= (1\delta_{J_1}{}^{K_3} + 0\delta_{J_1}{}^{L_3})(1\delta_{J_2}{}^{K_1} + 0\delta_{J_1}{}^{L_1})(0\delta_{J_2}{}^{K_1} + 1\delta_{J_1}{}^{L_2}) \\ &= \delta_{J_1}{}^{K_3}\delta_{J_2}{}^{K_1}\delta_{J_3}{}^{L_2} & (10) \end{aligned}$$

Thus we see that the action of the projection operators is, as it should be, the same, no matter how the GCM is represented, be it by a recombination vector or a recombination pair.

Note that equation (5) is functionally identical to that for the case of standard mask-based crossover [3], the only difference being the different recombination distribution, and hence the different set of $\lambda_I{}^{JK}(\mathbf{r})$ that are non-zero. As in the standard homologous crossover case, for binary strings we have 2^ℓ coupled, first-order difference equations to solve. The chief problem however, is the fact that on the right hand side we have $2^\ell \times 2^\ell \times (2\ell)^\ell = (8\ell)^\ell$ possible contributing terms.

3.1 String Example for $\ell = 2$

For example, for $\ell = 2$ there are sixteen GCMs denoted in recombination vector notation by $\{(v_1, v_2)\}$, where v_1 and v_2 run over the values 1, 2, 3 and 4. The sums over the strings J and K run over the values 1 to $\mathcal{A}^\ell$ for an alphabet of cardinality $\mathcal{A}$. Thus, for an arbitrary GRD, even at the two bit level there are $16 \times 4 \times 4 = 256$ $\lambda_I{}^{JK}(\mathbf{r})$ to compute for a given string I. Symbolically, however, the terms are quite simple

$$\begin{aligned} \textit{cloning} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(1,2) = \delta_{I_1 J_1}\delta_{I_2 J_2} & (11) \\ \textit{inversion} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(2,1) = \delta_{I_1 J_2}\delta_{I_2 J_1} & (12) \\ \textit{crossover} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(1,4) = \delta_{I_1 J_1}\delta_{I_2 K_2} & (13) \\ \textit{cross+inv} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(4,1) = \delta_{I_1 K_2}\delta_{I_2 J_1} & (14) \\ \textit{dup 1} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(1,1) = \delta_{I_1 J_1}\delta_{I_2 J_1} & (15) \\ \textit{dup 1} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(2,2) = \delta_{I_1 K_1}\delta_{I_2 K_1} & (16) \\ \textit{dup 2} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(1,3) = \delta_{I_1 J_1}\delta_{I_2 K_1} & (17) \\ \textit{dup 2} \quad & \lambda_{I_1 I_2}{}^{J_1 J_2 K_1 K_2}(2,4) = \delta_{I_1 J_2}\delta_{I_2 K_2} & (18) \end{aligned}$$

where we use the recombination vector notation and show only those GCMs that correspond to creation of genotype I using J as the first parent. The corresponding second parent terms can be found by interchanging J and K on the right hand side of (11-18) and letting $(v_1, v_2) \to (v_1', v_2')$, where $v_i' = (v_i + 2) \bmod 2$. The meaning of these terms, as alluded to in equations (11-18), is the following: the terms represented by GCMs $(1,2)$ (cloning of first parent) and $(3,4)$ (cloning of second parent) are *cloning* terms due to the application of a trivial standard crossover mask, where both offspring alleles come from the corresponding loci of only one of the parents. The *inversion* term is represented by GCMs $(2,1)$ (inversion of first parent) and $(4,3)$ (inversion of second parent). The GCMs $(1,4)$ and $(3,2)$ represent the results of standard *one-point crossover*, while the GCMs $(4,1)$ and $(2,3)$ represent the results of standard *one-point crossover* followed by an *inversion* (or vice versa). The terms denoted by *duplication 1* - one-parent duplication - are associated with the GCMs $(1,1)$, $(2,2)$, $(3,3)$ and $(4,4)$ and represent duplication of an allele from a single locus of a single parent. Finally, the *duplication 2* - two-parent duplications - GCMs $(1,3)$, $(2,4)$, $(3,1)$ and $(4,2)$ represent gene duplication as well, but where the two genes of the offspring come from the same locus but in different parents.

Substituting these expressions in (5), computing all terms, expanding the sums $\sum_K$ and $\sum_L$, and setting $I_1 = i'$, $I_2 = j'$, $J_1 = i$ and $J_2 = j$ for conciseness, one finds

$$\begin{aligned}
E(P_{i'j'}(t+1)) = \sum_{i,j=0,1} {M_{i'j'}}^{ij} \{ & (1-p_{xo})P'_{ij} \\
+ \quad & p_{xo}[(p_{11}+p_{33}+p_{22}+p_{44})P'_{ij}\delta_{ij} \\
+ \quad & ((p_{11}+p_{33})P'_{i\bar{j}} + (p_{22}+p_{44})P'_{\bar{i}j})\delta_{ij} \\
+ \quad & (p_{12}+p_{34})P'_{ij} + (p_{21}+p_{43})P'_{ji}(t) \\
+ \quad & (p_{13}+p_{31}+p_{14}+p_{32}+p_{23} \\
+ \quad & p_{41}+p_{42}+p_{24})P'_{ii}P'_{jj} \\
+ \quad & (p_{23}+p_{41}+p_{13}+p_{31})P'_{ii}P'_{j\bar{j}} \\
+ \quad & (p_{13}+p_{31}+p_{14}+p_{32})P'_{i\bar{i}}P'_{jj} \\
+ \quad & (p_{24}+p_{42}+p_{14}+p_{32})P'_{ii}P'_{\bar{j}j} \\
+ \quad & (p_{14}+p_{32})P'_{i\bar{i}}P'_{\bar{j}j} + (p_{13}+p_{31})P'_{i\bar{i}}P'_{j\bar{j}} \\
+ \quad & (p_{24}+p_{42}+p_{23}+p_{41})P'_{\bar{i}i}P'_{jj} \\
+ \quad & (p_{41}+p_{23})P'_{\bar{i}i}P'_{j\bar{j}} + (p_{42}+p_{24})P'_{\bar{i}i}P'_{\bar{j}j}]\}
\end{aligned} \tag{19}$$

where, for simplicity, we are restricting attention to a binary alphabet and $\bar{i}$ signifies the bit complement of i. The first term on the right hand side is a cloning-mutation term due to the fact that with probability $(1-p_{xo})$ strings are copied without recombination, P'_{ij} being the probability to select the genotype ij and ${M_{i'j'}}^{ij}$ being the probability to mutate it to the genotype $i'j'$. The meaning of the different terms in (20) is inherited from the meaning of the corresponding terms of the GRD, the p_{ab} being the notation for the GCM probability associated with the GCM (a, b). The Kronecker delta, ensures that the contribution from gene duplication from a single parent is only present for homogeneous offspring, i.e. those with both allele values the same. Note that of the 256 possibilities there are only 44 non-zero terms in (20). However, in order to compute which ones are non-zero all have to be computed. By way of comparison, the canonical genetic algorithm with one-point crossover, where $p_{14} = p_{32} = 1/2$ with all other GCMs zero, has only 8 non-zero terms out of the $2 \times 4 \times 4$ possible ones.

4 Coarse Grained Evolution Equations

4.1 Strings

For both homologous and generalised recombination it is clear that there is a great deal of redundancy in the string representation. In the case of homologous recombination it has been found that a coarse grained representation in terms of schemata makes the dynamics much more transparent, partly due to the fact that the number of terms on the right hand side of (5) reduces to 2^ℓ for a binary alphabet in the case of general homologous crossover which, when compared to 8^ℓ in the string basis, is a substantial reduction in complexity. For a particular type of recombination distribution, such as one-point crossover, where there are only $(\ell - 1)$ non-zero masks, the simplification is even greater, from 8^ℓ to $(\ell - 1)$. One is naturally inclined to ask whether an appropriate simplification can be effected in this more general case.

We first note again that the question of whether or not a particular locus, i, in the offspring comes from a particular locus, j, in a particular parent, is independent of the other offspring loci. In other words, using the recombination pair representation of the GRD, equation (6) and restricting (8) to fixed length, gives

$${\lambda_J}^{KL}(\mathbf{r}) = \prod_{i=1}^{\ell} \left((1-m_i){\delta_{I_i}}^{K_{v_i}} + m_i {\delta_{I_i}}^{L_{v_i}} \right) \tag{20}$$

Here, for a given recombination pair $\mathbf{r} = (m, v)$, m_i and $(1 - m_i)$ are projection operators that determine from which parent a particular offspring locus comes, while v_i determines from where in the parental chromosome it originates, K_{v_i} being the locus picked out of the first parent by v_i and L_{v_i} the locus picked out of the second parent. The string construction terms in equation (5) for a given recombination pair $\mathbf{r}$ can then be written as

$$\begin{aligned}
&\sum_K \sum_L \lambda_J{}^{KL}(\mathbf{r}) P'_K(t) P'_L(t) \\
&\quad = \sum_{K_1 \cdots K_\ell} \sum_{L_1 \cdots L_\ell} \prod_{i=1}^{\ell} \left((1 - m_i) \delta_{I_i}{}^{K_{v_i}} + m_i \delta_{I_i}{}^{L_{v_i}} \right) P'_{K_1 \cdots K_\ell} P'_{L_1 \cdots L_\ell}
\end{aligned} \tag{21}$$

The Kronecker deltas in (21) in conjunction with the string sums act as projection operators. To understand this, before passing to the general case, we will examine what happens for one particular locus i of the offspring J, after generalised recombination and before mutation. This locus has its origin in a parental locus v_i. Let us assume that $m_i = 0$, then this locus comes from the first parent and we have $J_i = K_{v_i}$ from the action of $\delta_{I_i}{}^{K_{v_i}}$. Explicitly

$$\begin{aligned}
&\sum_{K_{v_i}} \sum_{L_{v_i}} \left((1 - m_i) \delta_{J_i}{}^{K_{v_i}} + m_i \delta_{J_i}{}^{L_{v_i}} \right) P'_{K_1 \cdots K_{v_i} \cdots K_\ell} P'_{L_1 \cdots L_{v_i} \cdots L_\ell} \\
&\quad = \left(\sum_{K_{v_i}} \delta_{J_i}{}^{K_{v_i}} P'_{K_1 \cdots K_{v_i}} \right) \left(\sum_{L_{v_i}} P'_{L_1 \cdots L_{v_i} \cdots L_\ell} \right) \\
&\quad = P'_{K_1 \cdots J_{i_{v_i}} \cdots K_\ell} P'_{L_1 \cdots *_{v_i} \cdots L_\ell}
\end{aligned} \tag{22}$$

where $J_{i_{v_i}}$ denotes that the allele corresponding to J_i in the offspring is the same as that in the v_ith locus of the parent and $*_{v_i}$ is the standard wildcard symbol signifying that the allele values at the locus L_{v_i} have been summed over, thus leaving the marginal probability $P'_{L_1 \cdots *_{v_i} \cdots L_\ell}$. For example, $\sum_{j=0,1} P'_{ij} = P'_{i0} + P'_{i1} = P'_{i*}$. One can also think of $\sum_{K_{v_i}} \delta_{J_i}{}^{K_{v_i}}$ as restricting the sum over K_{v_i} to only one value.

Thus, we see that, as in the case of homologous recombination, the notion of schemata naturally emerges, i.e. that coarse grained objects as opposed to the microscopic strings themselves naturally appear. In distinction to the homologous case however, the locus v_i that gives rise to the ith locus of the offspring is not necessarily the ith locus of the first parent. Similarly, for $m_i = 1$, then $J_i = L_{v_i}$, and we have

$$\begin{aligned}
&\sum_{K_{v_i}} \sum_{L_{v_i}} \left((1 - m_i) \delta_{J_i}{}^{K_{v_i}} + m_i \delta_{J_i}{}^{L_{v_i}} \right) P'_{K_1 \cdots K_{v_i} \cdots K_\ell} P'_{L_1 \cdots L_{v_i} \cdots L_\ell} \\
&\quad = \left(\sum_{K_{v_i}} P'_{K_1 \cdots K_{v_i}} \right) \left(\sum_{L_{v_i}} \delta_{J_i}{}^{L_{v_i}} P'_{L_1 \cdots L_{v_i} \cdots L_\ell} \right) \\
&\quad = P'_{K_1 \cdots *_{v_i} \cdots K_\ell} P'_{L_1 \cdots J_{i_{v_i}} \cdots L_\ell}
\end{aligned} \tag{23}$$

This coarse graining that appears via projection can be repeated every time a projection operator, i.e. a Kronecker-delta and a corresponding string sum in (21), appears. For example, consider the ith and the jth offspring loci that originate from the parental loci v_i and v_j. In this case, for

$m_i = 0$ and $m_j = 0$, for instance, we have

$$\begin{aligned}
&\sum_{K_{v_j}}\sum_{L_{v_j}}\sum_{K_{v_i}}\sum_{L_{v_i}}((1-m_j)\delta_{J_j}{}^{K_{v_j}} + m_j\delta_{J_j}{}^{L_{v_j}})((1-m_i)\delta_{J_i}{}^{K_{v_i}} + m_i\delta_{J_i}{}^{L_{v_i}})P'_{K_1\cdots K_\ell}P'_{L_1\cdots L_\ell} \\
&= \left(\sum_{K_{v_j}}\sum_{K_{v_i}}\delta_{J_j}{}^{K_{v_j}}\delta_{J_i}{}^{K_{v_i}}P'_{K_1\cdots K_{v_i}\cdots K_{v_j}\cdots K_\ell}\right)\left(\sum_{L_{v_j}}\sum_{L_{v_i}}P'_{L_1\cdots *_{v_i}\cdots L_{v_j}\cdots L_\ell}\right) \\
&= \left(\sum_{K_{v_j}}\delta_{J_i}{}^{K_{v_j}}P'_{K_1\cdots J_{i_{v_i}}\cdots K_{v_j}\cdots K_\ell}\right)\left(\sum_{L_{v_i}}P'_{L_1\cdots *_{v_i}\cdots L_{v_j}\cdots L_\ell}\right) \\
&= P'_{K_1\cdots J_{i_{v_i}}\cdots J_{j_{v_j}}\cdots K_\ell}P'_{L_1\cdots *_{v_i}\cdots *_{v_j}\cdots L_\ell} \qquad (24)
\end{aligned}$$

with analogous terms for $m_i = 0$, $m_j = 1$; $m_i = 1$, $m_j = 0$; and $m_i = 1$, $m_j = 1$. Note that we are here showing in $(K_1 \cdots J_{v_i} \cdots J_{v_j} \cdots K_\ell)$, J_{v_i} to the left of J_{v_j}. If $v_j < v_i$ however, it would be to the right.

As we are allowing for the possibility of duplication it may be that in the components of v, $v_i = v_j$, and hence both the ith and jth loci in the offspring originate from the same locus in one of the parents, the duplication being from one parent if $m_i = m_j$ and from two parents if $m_i \neq m_j$. In the case of one parent duplication $m_i = m_j = 0$ and if $v_i = v_j$ then K_{v_i} and K_{v_j} refer to the same locus and there is then only one sum over K_{v_i} in (24) which gives

$$\sum_{K_{v_i}}\delta_{J_j}{}^{K_{v_i}}\delta_{J_i}{}^{K_{v_i}}P'_{K_1\cdots K_\ell} = \delta_{J_j}{}^{J_i}P'_{K_1\cdots J_{i_{v_i}}\cdots K_\ell} \qquad (25)$$

It should be noted though that due to the duplication there will be a locus, j, that does not appear in v and this locus will then correspond to a free sum when considering $\sum_{K_1\cdots K_\ell}$. This can be seen through a simple example with $\mathbf{r} = (0010, (1, 2, 3, 1))$. In this case

$$\begin{aligned}
&\sum_{K_1}\sum_{K_2}\sum_{K_3}\sum_{K_4}\delta_{J_1}{}^{K_{v_1}}\delta_{J_2}{}^{K_{v_2}}\delta_{J_4}{}^{K_{v_4}}P'_{K_1K_2K_3K_4} \\
&= \left(\sum_{K_{v_1}}\delta_{J_1}{}^{K_{v_1}}\right)\left(\sum_{K_{v_2}}\delta_{J_2}{}^{K_{v_2}}\right)\left(\sum_{K_{v_3}}\right)\left(\sum_{K_4}\delta_{J_4}{}^{K_{v_4}}\right)P'_{K_1K_2K_3K_4} \qquad (26) \\
&= \left(\sum_{K_1}\delta_{J_1}{}^{K_1}\right)\left(\sum_{K_2}\delta_{J_2}{}^{K_2}\right)\left(\sum_{K_3}\right)\left(\sum_{K_4}\delta_{J_4}{}^{K_1}\right)P'_{K_1K_2K_3K_4} \qquad (27) \\
&= \sum_{K_4}P'_{J_1J_2*K_4}\delta_{J_1}{}^{J_4} \\
&= P'_{J_1J_2**}\delta_{J_1}{}^{J_4} \qquad (28)
\end{aligned}$$

In (26) we are simply reordering the $\sum_{K_i}$ according to v. However, as v is no longer a permutation list, then there is one K_i, K_4 in fact, that cannot be reordered, as there is no $v_i = 4$ entry in v. In (27) the explicit values for (v_1, v_2, v_3, v_4) have been substituted. Note that the effect of the duplication is to leave a free sum over K_4, which leads to an extra coarse graining - a wildcard appears at the fourth locus - and a constraint, manifest in $\delta_{J_1}{}^{J_4}$, that the first and fourth offspring alleles be the same. The pattern of how the projection works at the level of each locus should now be clear from (22-24) and the subsequent discussion of the extra subtlety of what happens when single-parent duplication is present.

Above we have seen how a coarse graining naturally appears due to the action of projection operators acting on the vectors P_J and P'_J. One can also determine the action of the coarse graining at the level of the strings themselves[3] by viewing strings or schemata as *sets*. Given a

[3]How the coarse graining acts at the level of the strings is also, of course, a consequence of how the projection operators work; as can be seen in (22-24), where it is the index (subscript) of the string proportion or selection probability. For example, in equation (28) the schema $J_1J_2 * *$ emerges from the coarse graining process.

string J and a GCM $\mathbf{r}$, then using equation (1), the set

$$J^r = \bigcap_{i=1}^{\ell} (*^{v_i-1}(\bar{m}_i \bullet J_i) \; *^{\ell-v_i}) \tag{29}$$

where $*^n$ signifies $*$ repeated n times, denotes the alleles of the offspring J inherited from the first parent, and

$$J^{\bar{r}} = \bigcap_{i=1}^{\ell} (*^{v_i-1}(m_i \bullet J_i) \; *^{\ell-v_i}) \tag{30}$$

denote the alleles of the offspring J inherited from the second. Note how the alleles of J appear in *either* J^r or $J^{\bar{r}}$. Thus, for example, for $\ell = 4$, if $r = (0010, (4, 3, 4, 1))$, which represents two parent duplication then

$$\begin{aligned} J^r &= (*^{v_1-1}J_1*^{4-v_1}) \cap (*^{v_2-1}J_2*^{4-v_2}) \cap (*^4) \cap (*^{v_4-1}J_4*^{4-v_4}) \\ &= (*^3 J_1) \cap (*^2 J_2 *) \cap (*^4) \cap (J_4 *^3) \\ &= J_4 * J_2 J_1 \end{aligned}$$

and

$$\begin{aligned} J^{\bar{r}} &= (*^4) \cap (*^4) \cap (*^{v_3-1}J_3*^{4-v_3}) \cap (*^4) \\ &= (*^4) \cap (*^4) \cap (*^3 J_3) \cap (*^4) \\ &= * * * J_3 \end{aligned}$$

Similarly, for our previous example with $\mathbf{r} = (0010, (1, 2, 3, 1))$, which exhibits one parent duplication, we have

$$\begin{aligned} J^r &= (*^{v_1-1}J_1*^{4-v_1}) \cap (*^{v_2-1}J_2*^{4-v_2}) \cap (*^4) \cap (*^{v_4-1}J_4*^{4-v_4}) \\ &= (J_1 *^3) \cap (* J_2 *^2) \cap (*^4) \cap (J_4 *^3) \\ &= J_1 J_2 * * \cap J_4 J_2 * * \end{aligned}$$

which is only non-empty when $J_1 = J_4$, which is the same constraint as seen in (28).

As we have now understood how generalised recombination acts in terms of projection operators on a locus by locus basis, and we also understand how it operates at the level of strings and schemata viewed as sets, we can now return to the general case in (21) to understand further how coarse graining naturally appears there. As m_i and $(1 - m_i)$ satisfy

$$(1 - m_i)^2 = (1 - m_i) \qquad m_i^2 = m_i \qquad m_i(1 - m_i) = 0 \tag{31}$$

using (31) we can write

$$\begin{aligned} &\prod_{i=1}^{\ell} \left((1 - m_i)\delta_{J_i}^{\;K_{v_i}} + m_i \delta_{J_i}^{\;L_{v_i}} \right) \\ &\quad = \prod_{i=1}^{\ell} \left((1 - m_i)\delta_{J_i}^{\;K_{v_i}} + m_i \right) \prod_{i=1}^{\ell} \left((1 - m_i) + m_i \delta_{I_i}^{\;L_{v_i}} \right) \end{aligned} \tag{32}$$

We now associate the two products on the right hand side of (32) with the string sums in (21) and consider the two terms

$$\sum_{K_1 \cdots K_\ell} \prod_{i=1}^{\ell} \left((1 - m_i)\delta_{J_i}^{\;K_{v_i}} + m_i \right) P'_{K_1 \cdots K_\ell} \tag{33}$$

$$\sum_{L_1 \cdots L_\ell} \prod_{i=1}^{\ell} \left((1 - m_i) + m_i \delta_{J_i}^{\;L_{v_i}} \right) P'_{L_1 \cdots L_\ell} \tag{34}$$

which consist of ℓ projection operators of the type $\sum_{K_i}((1-m_i)\delta_{J_i}^{\ K_{v_i}}+m_i)$ acting on the string selection probabilities $P'_{K_1\cdots K_\ell}$. We have seen in (22) the action of one of these projectors on $P'_{K_1\cdots K_{v_i}\cdots K_\ell}$, replacing K_{v_i} by J_i if $m_i=0$, as $\sum_{K_i}((1-m_i)\delta_{J_i}^{\ K_{v_i}}P'_{K_1\cdots K_{v_i}\cdots K_\ell}=P'_{K_1\cdots J_i\cdots K_\ell}$, and by $*$ if $m_i=1$, as $\sum_{K_i}P'_{K_1\cdots K_{v_i}\cdots K_\ell}=P'_{K_1\cdots *\cdots K_\ell}$. The repeated action is just to project out those elements - $m_i=0$, $J_i=K_{v_i}$ - of the parent K that contribute to the offspring J and to turn the other components corresponding to $m_i=1$ into wildcards. Thus we may identify (33) and (34) as the selection probabilities for two schemata. As the action of the projection operators is to replace an allele K_{v_i} with J_i and L_{v_i} with $*$ if $m_i=0$ and to replace L_{v_i} with J_i and K_{v_i} with $*$ if $m_i=1$, which has exactly the same effect as that seen in (29) and (30) we see that the two schemata of relevance are just those of (29) and (30). Thus, we can identify the selection probabilities of these schemata as

$$P'_{J^r}=\sum_{K_1}\cdots\sum_{K_\ell}\prod_{i=1}^{\ell}((1-m_i)\delta_{J_i}^{\ K_{v_i}}+m_i)P'_{K_1\cdots K_\ell} \tag{35}$$

and

$$P'_{J^{\bar{r}}}=\sum_{L_1}\cdots\sum_{L_\ell}\prod_{i=1}^{\ell}((1-m_i)+m_i\delta_{J_i}^{\ L_{v_i}})P'_{L_1\cdots L_\ell} \tag{36}$$

Hence, substituting into (21) we have

$$\sum_K\sum_L\lambda_J^{\ KL}(\mathbf{r})P'_K(t)P'_L(t)=P'_{J^r}(t)P'_{J^{\bar{r}}}(t) \tag{37}$$

The important observation here is that for each $\mathbf{r}$ the corresponding string sums have disappeared. This means that instead of having to consider 4^ℓ terms for every recombination pair one only has to consider one.[4] The schemata J^r and $J^{\bar{r}}$ are just the Building Blocks for the string J.

It is useful at this point to compare with previously used notation [19], where a Building Block was represented as

$$\Gamma(h,I_r)=\bigcap_{k=1}^{|I_r|}(*^{v_{i_k}-1}h_{i_k}*^{\ell-v_{i_k}})$$

with a corresponding conjugate Building Block $\Gamma(h,\bar{I}_r)$, where $I_r=\{i:m_i=1\}$ represents the genes picked out from the first parent by $\mathbf{r}$ that go to form part of the offspring h, and $\bar{I}_r=\{i:m_i=0\}$ is the complementary set picked out from the second parent. Therefore

$$\begin{aligned}&\sum_{a\in\Omega}p(a,t)\prod_{i\in I_r}\delta(h_i=a_{v_i})\\&\quad=\sum_{a\in\Gamma(h,I_r)}p(a,t)=p(\Gamma(h,I_r),t)\end{aligned}$$

where $p(a,t)$ is the selection probability for the string a. The $\sum_{a\in\Omega}\prod_{i\in I_r}\delta(h_i=a_{v_i})$ here is a projection operator analogous to (35), whose action can be thought of as restricting the sum $\sum_{a\in\Omega}$ to a subset corresponding to the schema $\Gamma(h,I_r)$ which is identical to the schema J^r in this paper.

Finally then, we arrive at the following for fixed-length strings:

Coarse-grained string evolution equation

The expected frequency of a string I at the next generation in a generational GA with arbitrary selection with replacement, mutation and generalised recombination is

$$E(P_I(t+1))=\sum_J M_I^{\ J}[(1-p_{xo})P'_J(t)+p_{xo}\sum_{\mathbf{r}}p_c(\mathbf{r})P'_{J^r}(t)P'_{J^{\bar{r}}}(t)] \tag{38}$$

[4] More generally, for cardinality $\mathcal{A}$ the saving is $\mathcal{A}^{2\ell}$.

where $J^r = \bigcap_{i=1}^{\ell}(*^{v_{i_k}-1}(\bar{m}_i \bullet J_i)\ *^{\ell-v_{i_k}})$ *and* $J^{\bar{r}} = \bigcap_{i=1}^{\ell}(*^{v_{i_k}-1}(m_i \bullet J_i)\ *^{\ell-v_{i_k}})$ *are the Building Block Schemata for the string* J *with respect to the GCM* $\mathbf{r} = (m, v)$.

In the limit $n \to \infty$, $E(P_I(t)) \to P_I(t)$ the probability to find a genotype I at time t and so one obtains a deterministic equation for the evolution of the $P_I(t)$. Equation (38) extends previous results obtained for homologous crossover to the case of generalised recombination. It is striking that the functional form of this dynamical equation is identical to that found for homologous crossover [26, 27], even though the range of genetic operators covered by the equation goes far beyond simple homologous crossover, encompassing different types of duplication and deletion, as well as a generalisation of the inversion operator to the larger and more general class of permutations. The key difference between generalised and homologous recombination then lies not in whether or not Building Blocks are used, but only in the nature of those Building Blocks, generalised recombination leading to a much richer set of them.

The differences can be simply illustrated considering $\ell = 2$. To see how one is led to different Building Blocks, consider the GRD where only p_{41} and p_{23} are non-zero (for simplicity we put $p_{xo} = 1$ and $p_m = 0$). From equation (5)

$$\begin{aligned} E(P_{I_1 I_2}(t+1)) &= p_{41} \sum_{J_1 J_2} \sum_{K_1 K_2} \delta_{I_1 K_2}\delta_{I_2 J_1} P'_{J_1 J_2} P'_{K_1 K_2} + p_{23} \sum_{J_1 J_2} \sum_{K_1 K_2} \delta_{I_1 J_2}\delta_{I_2 K_1} P'_{J_1 J_2} P'_{K_1 K_2} \\ &= p_{41} P'_{I_2 *} P'_{*I_1} + p_{23} P'_{*I_1} P'_{I_2 *} \end{aligned} \tag{39}$$

where we have taken the opportunity in the first line to also illustrate how the coarse graining that corresponds to this GRD appears, by starting off with the string representation of the equation, rather than the coarse grained one. The Building Blocks in this case are I_2* and $*I_1$, whereas for homologous 1-point crossover, where the corresponding GRD is p_{14} and p_{32} and the rest zero, the Building Blocks would be I_1* and $*I_2$. Thus, we see how generalised recombination may use Building Blocks that are not available in homologous crossover.

From equation (38), as introduced in [26, 24, 25, 23],[5] one defines the *effective* fitness, $f_I^{\text{eff}}(t)$, of the string I to be, in the case of proportional selection,

$$E(P_I(t+1)) = \frac{f_I^{\text{eff}}(t)}{\bar{f}(t)} P_I(t) \tag{40}$$

Thus, in the case at hand, we have

$$f_I^{\text{eff}} = \frac{\bar{f}(t)}{P_I(t)} \sum_J {M_I}^J \left[(1-p_{xo}) \frac{f_J}{\bar{f}(t)} P_J(t) + p_{xo} \sum_{\mathbf{r}} p_c(\mathbf{r}) \frac{f_{J^r}(t)}{\bar{f}(t)} P_{J^r}(t) \frac{f_{J^{\bar{r}}}(t)}{\bar{f}(t)} P_{J^{\bar{r}}}(t) \right]$$

where $f_{J^r}(t) = \sum_{J \in J^r} f_J P_J(t) / \sum_{J \in J^r} P_J(t)$ is the fitness of the schema J^r. Those strings for which $f_I^{\text{eff}}(t) > \bar{f}(t)$ increase in number, while those with $f_I^{\text{eff}}(t) < \bar{f}(t)$ decrease. The effective fitness depends on all the genetic operators not just selection, hence a string, f_I, may increase in frequency relative to another even if $f_I < f_J$ if it is more favoured by mutation or crossover. The latter is governed by the selection weighted linkage disequilibrium coefficient

$$\Delta_I^r(t) = (P'_I(t) - P'_{I^r}(t) P'_{I^{\bar{r}}}(t)) \tag{41}$$

If $\Delta_I^r(t) < 0$ for a given GCM then generalised recombination increases the proportion of genotype I in the next generation, relative to the proportion one would have with $p_{xo} = 0$; whereas, if $\Delta_I^r(t) > 0$ the proportion is decreased relative to the no-recombination limit.

4.2 Schemata

One lesson that previous work on coarse grained formulations has taught us is that schemata naturally emerge in any study of homologous recombination. We see from equation (38) that this

[5] This is a generalisation of that of [14, 15] and [7, 8] where only the destructive effect of crossover is considered.

is also true for generalised recombination. The fact that schemata have emerged in the dynamics of strings evolving under the action of generalised recombination is manifest at several levels. At the level of the strings themselves, viewed as sets, it is manifest in (29) and (30), which are schemata corresponding to hyperplanes of Ω; while at the level of the string proportions and selection probabilities it is manifest in (35) and (36), which are probabilities associated with hyperplanes of Ω. The form of (38) also tells us that in order to describe the dynamics of strings one must simultaneously understand the dynamics of schemata and, in particular, Building Block Schemata, as the string frequencies at $t+1$ depend on the Building Block frequencies at t.

For homologous crossover, one of the most remarkable features of the analog of equation (38) is its form invariance under a further coarse graining [26, 27], i.e. that the functional form of the equations for a schema is identical to that of the equations for the strings themselves. This means that Building Blocks for a string are composed, in their turn, by other more coarse grained (lower order - less defining bits/more wildcards) Building Blocks, which in their turn etc., the whole hierarchy terminating at 1-schemata, i.e. schemata with only one defining bit and $(\ell-1)$ wildcards, as the latter cannot be composed of more elementary objects. It is precisely the existence of this form invariance and the hierarchical nature of the relationship between the different Building Blocks that has led to so many new results using the coarse grained formulation. We are thus led to consider whether for generalised recombination the same features appear, which can then be further exploited to gain a better theoretical understanding and derive new practical results.

Thus, we begin by considering what happens when we act with a coarse graining operator $\mathcal{C}^n$ on the string I to obtain a schema I^n, i.e. $I^n = \mathcal{C}^n I$. We can implement this by once again appealing to the concept of a standard crossover mask $n = (n_1, \cdots, n_\ell)$, such that, if $n_i = 0$, the locus i will not be coarse grained, while, if $n_i = 1$, it will, i.e. $I_i \to *$. One may implement this coarse graining at the level of strings using (1) or, as done in defining the Building Block schemata, J^r and $J^{\bar{r}}$, using sets. The corresponding action at the level of schema frequencies and selection probabilities will be implemented using projection operators. For the string I itself, using (1) one finds

$$I^n = (\bar{n}_1 \bullet I_1, \cdots, \bar{n}_\ell \bullet I_\ell \tag{42}$$

whereas in terms of I and I^n viewed as sets

$$I^n = \bigcap_{i=1}^{\ell} \left(*^{i-1} (\bar{n}_i \bullet I_i) \; *^{\ell - i} \right) \tag{43}$$

As an example, for $\ell = 4$ and $n = 0101$, we have $I^{0101} = (\bar{n}_1 \bullet I_1, \bar{n}_2 \bullet I_2, \bar{n}_3 \bullet I_3, \bar{n}_4 \bullet I_4) = (I_1, *, I_3, *)$ in one representation, and $I^{0101} = I_1 \, *^3 \bigcap *^4 \bigcap *^2 I_3 * \bigcap *^4 = I_1 * I_3 *$ in the other. Note that the mask n in (43) is not that associated with the recombination pair $\mathbf{r} = (m, v)$. The composition of two coarse grainings, implemented by masks n and n', is given by

$$I^{nn'} = ((\bar{n}_1 \bullet (\bar{n}'_1 \bullet I_1)), \cdots, (\bar{n}_\ell \bullet (\bar{n}'_\ell \bullet I_i)) \tag{44}$$

or

$$I^{nn'} = \bigcap_{i=1}^{\ell} \left(*^{i-1} (\bar{n}_i \bullet (\bar{n}'_i \bullet I_i)) \; *^{\ell - i} \right) \tag{45}$$

We may now ask how a coarse graining $\mathcal{C}^n$ composes with the coarse graining that takes us to the Building Block schemata (29) and (30). In this case one arrives at coarse grained Building Blocks, i.e. Building Blocks of the schema I^n as opposed to the string I, of the form

$$I^{rn} = \bigcap_{i=1}^{\ell} (*^{v_i - 1} (\bar{n}_i \bullet (\bar{m}_i \bullet I_i)) \; *^{\ell - v_i}) \tag{46}$$

and

$$I^{\bar{r}n} = \bigcap_{i=1}^{\ell} (*^{v_i-1}(\bar{n}_i \bullet (m_i \bullet I_i)) \; *^{\ell - v_i}) \tag{47}$$

The content of (46) is that the allele I_i appears if and only if $m_i = n_i = 0$, otherwise a $*$ appears in the corresponding position, while for (47) the allele I_i appears if and only if $n_i = 0$ and $m_i = 1$. For example, if $n = 0101$ and $\mathbf{r} = (0010, (4, 3, 4, 1))$, then $I^{rn} = *^3 I_1 \bigcap *^4 \bigcap *^4 \bigcap *^4 = * * * I_1$ and $I^{\bar{r}n} = *^4 \bigcap *^4 \bigcap *^3 I_3 \bigcap *^4 = * * * I_4$. To compare with homologous crossover we only need replace $v = (4, 3, 4, 1) \rightarrow (1, 2, 3, 4)$ to obtain $I^{rn} = I_1 \bigcap *^4 \bigcap *^4 \bigcap *^4 = I_1 * **$ and $I^{\bar{r}n} = *^4 \bigcap *^4 \bigcap *^2 I_3 * \bigcap *^4 = ** I_3 *$. In both cases the "offspring" I^n that results from recombining the "parents" I^{rn} and $I^{\bar{r}n}$ is $I_1 * I_3 *$.

Having determined the action of the coarse graining $\mathcal{C}^n$ at the level of strings and schemata defined as sets, we now wish to see how it acts on the evolution equation. We represent the coarse graining now by $\mathcal{C}^n_I$, where

$$\mathcal{C}^n_I = \sum_{I'_1 \cdots I'_\ell} \prod_{i=1}^{\ell} \left((1 - n_i) \delta_{I_i}^{\; I'_i} + n_i \right) \tag{48}$$

which will project out from any vector defined on the configuration space, Ω, a coarse grained vector that naturally lives on a subspace, Ω^n, of Ω. As an example, consider

$$\begin{aligned} \mathcal{C}^{010}_I P_{I'} &= \sum_{I'_1=0,1} \sum_{I'_2=0,1} \sum_{I'_3=0,1} (1\delta_{I_1}^{\; I'_1} + 0)(0\delta_{I_2}^{\; I'_2} + 1)(1\delta_{I_3}^{\; I'_3} + 0) P_{I'_1 I'_2 I'_3} \\ &= P_{I_1 * I_3} \end{aligned} \tag{49}$$

We now wish to coarse grain equation (38) using equation (48). Acting with $\mathcal{C}^n_I$ on the left hand side one immediately obtains

$$\mathcal{C}^n_I E(P_{I'}(t+1)) = E\left(\sum_{I'_1} \cdots \sum_{I'_i} \cdots \sum_{I'_\ell} \prod_{i=1}^{\ell} \left((1 - n_i) \delta_{I_i}^{\; I'_i} + n_i \right) P_{I'_1 \cdots I'_i \cdots I'_\ell}(t+1) \right) = E(P_{I^n}(t+1)) \tag{50}$$

where I^n is defined in (43). Acting with $\mathcal{C}^n_I$ on the right hand side it can be shown that

$$\mathcal{C}^n_I \sum_J M_I{}^J P_J(t) = \sum_{J^n} M_{I^n}{}^{J_n} P_{J^n}(t) \tag{51}$$

where $M_{I^n}{}^{J_n}$ is the mutation matrix projected down onto the subspace Ω^n defined by $\mathcal{C}^n_I$. Ω^n is defined, for a schema of order q, to be the 2^q-dimensional subspace of Ω projected out by the action of $\mathcal{C}^n$. For instance, for $\ell = 3$ and $n = 010$, $q = 2$. Ω in this case is spanned by the eight strings $111, \cdots, 000$ while Ω^n is spanned by the four schemata $1 * 1$, $1 * 0$, $0 * 1$ and $0 * 0$. The explicit action of $\mathcal{C}^n_I$ on the mutation and cloning term for $\ell = 2$ with the coarse graining mask $n = 10$ is

$$\begin{aligned} \mathcal{C}^{10}_I \sum_{J_1=0,1} \sum_{J_2=0,1} M_{I'_1 I'_2}{}^{J_1 J_2} P_{J_1 J_2}(t) &= \sum_{I'_1=0,1} \sum_{I'_2=0,1} \delta_{I_2}^{\; I'_2} \sum_{J_1=0,1} \sum_{J_2=0,1} M_{I_1 I_2}{}^{J_1 J_2} P_{J_1 J_2}(t) \\ &= pP_{00}(t) + (1-p)P_{01}(t) + pP_{10}(t) + (1-p)P_{11}(t) \\ &= (1-p)P_{*1}(t) + pP_{*0}(t) \\ &\equiv \sum_{J_2=0,1} M_{I_2}{}^{J_2} P_{*J_2}(t) \end{aligned} \tag{52}$$

To see how $\mathcal{C}^n$ operates on the construction term of (38), for transparency we put $p_m = 0$ first, and consider the action of $\mathcal{C}^n_J$ for one recombination pair $\mathbf{r}$, then

$$\mathcal{C}^n_J P'_{J^{rr}} P'_{J^{\bar{r}\bar{r}}} = \sum_{J'_1 \cdots J'_\ell} \prod_{i=1}^{\ell} ((1 - n_i) \delta_{J_i}^{\; J'_i} + n_i) P'_{J'^r_1 \cdots J'^r_\ell} P'_{J'^{\bar{r}}_1 \cdots J'^{\bar{r}}_\ell} \tag{53}$$

where J^r and $J^{\bar{r}}$ are given by (29) and (30) respectively, the components of which have to be determined from these equations for a given $\mathbf{r} = (m, v)$. To determine the action of $\mathcal{C}_J^n$ we divide $\prod_{i=1}^{\ell}$ into $\prod_{i\in M_0}\prod_{i\in M_1}$, where M_0 and M_1 are the sets that correspond to the 0s and 1s respectively in the mask m of $\mathbf{r} = (m, v)$. $|M_0|$ is then the number of 0s and $|M_1|$ the number of 1s, where $\ell = |M_0| + |M_1|$. If $m_j = 0$ then the allele J_j appears in the first parent, in a position determined by v_j, and if $m_j = 1$ it appears in the second parent. Then, if $n_i = 0$ and $m_i = 0$ $\mathcal{C}_J^n$ picks out the allele J_i from the first parent, while if $n_i = 0$ and $m_i = 1$ it picks it out from the second. On the other hand, if $n_i = 1$ and $m_i = 0$ then $J_i \to *$ in the first parent and, if $n_i = 1$ and $m_i = 1$ then $J_i \to *$ in the second parent. We can illustrate this by considering only one locus, i: there are four possibilities for the pair (n_i, m_i) - $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. Then, considering $A = \sum_{J_i'}\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right) P'_{J_1'^r\cdots J_\ell'^r} P'_{J_1'^{\bar{r}}\cdots J_\ell'^{\bar{r}}}$; for $(0,0)$, $A \to P'_{J_1'^r\cdots J_i\cdots J_\ell'^r} P'_{J_1'^{\bar{r}}\cdots *_i\cdots J_\ell'^{\bar{r}}}$, where J_i and $*_i$ are in the slot determined by v_i. Similarly, for $(0,1)$ $A \to P'_{J_1'^r\cdots *_i\cdots J_\ell'^r} P'_{J_1'^{\bar{r}}\cdots J_i\cdots J_\ell'^{\bar{r}}}$; while for $(1,0)$ and $(1,1)$, $A \to P'_{J_1'^r\cdots *_i\cdots J_\ell'^r} P'_{J_1'^{\bar{r}}\cdots *_i\cdots J_\ell'^{\bar{r}}}$. This projection can be repeated to obtain

$$\mathcal{C}_J^n P'_{J'^r} P'_{J'^{\bar{r}}} = P'_{J'^{rn}} P'_{J'^{\bar{r}n}} \tag{54}$$

where J'^{rn} and $J'^{\bar{r}n}$ are as given in (46) and (47).

We can see this property emerging more explicitly by considering the action of the projection, or coarse graining, operators. Beginning at the level of the strings, the construction term can be coarse grained using a mask n in the following form

$$\begin{aligned}
&\mathcal{C}_J^n \sum_K \sum_L \lambda_J^{\ KL}(\mathbf{r}) P'_K(t) P'_L(t) \\
&\quad = \sum_{J_1'\cdots J_\ell'} \sum_{K_1\cdots K_\ell} \sum_{L_1\cdots L_\ell} P'_{K_1\cdots K_\ell} P'_{L_1\cdots L_\ell} \prod_{i=1}^{\ell}\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right) \lambda_{J_1'\cdots J_\ell'}^{\ \ K_1\cdots K_\ell L_1\cdots L_\ell}(\mathbf{r})
\end{aligned}$$

where

$$\lambda_{J_1'\cdots J_\ell'}^{\ \ K_1\cdots K_\ell L_1\cdots L_\ell}((m,v)) = \prod_{i=1}^{\ell}\left((1-m_i)\delta_{J_i'}^{\ K_{v_i}} + m_i\right)\prod_{i=1}^{\ell}\left((1-m_i) + m_i\delta_{J_i'}^{\ L_{v_i}}\right) \tag{55}$$

Writing

$$\prod_{i=1}^{\ell}\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right) = \prod_{i\in M_0}\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right)\prod_{i\in M_1}\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right) \tag{56}$$

then we can see that

$$\left(\sum_{J_i'(M_0)}\prod_{i\in M_0}\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right)\right)\left(\sum_{K_1\cdots K_\ell}\prod_{i=1}^{\ell}\left((1-m_i)\delta_{J_i'}^{\ K_{v_i}} + m_i\right)\right)$$

where $\sum_{J_i'(M_0)}$ is a sum over those loci that correspond to 0s in m, is such that the second product projects onto the components of J' that originate in the first parent, while the first product further projects on that subset for which $n_i = 0$, i.e.

$$\begin{aligned}
P'_{J^{rn}} &= \sum_{J_i'(M_0)}\sum_{K_1\cdots K_\ell}\prod_{i\in M_0}\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right)\prod_{i=1}^{\ell}\left((1-m_i)\delta_{J_i'}^{\ K_{v_i}} + m_i\right) P'_{K_1\cdots K_\ell} \\
&= \sum_{J_1'\cdots J_\ell'}\sum_{K_1\cdots K_\ell}\prod_{i=1}^{\ell}\left((1-m_i)\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right) + m_i\right)\left((1-m_i)\delta_{J_i'}^{\ K_{v_i}} + m_i\right) P'_{K_1\cdots K_\ell} \\
&= \sum_{J_1'\cdots J_\ell'}\prod_{i=1}^{\ell}\left((1-m_i)\left((1-n_i)\delta_{J_i}^{\ J_i'} + n_i\right) + m_i\right) P'_{J'^r}
\end{aligned} \tag{57}$$

where J^{rn} is given by (46) and the introduction of $(1-m_i)$ and m_i in (57) allows us to extend the range of the first product to ℓ. Similarly, one finds

$$\begin{aligned} P'_{J^{\bar{r}n}} &= \sum_{J'_i(M_1)} \sum_{L_1\cdots L_\ell} \prod_{i\in M_1} \left((1-n_i)\delta_{J_i}^{\ J'_i} + n_i\right) \prod_{i=1}^{\ell} \left((1-m_i)\delta_{J'_i}^{\ L_{v_i}} + m_i\right) P'_{L_1\cdots L_\ell} \\ &= \sum_{J'_1\cdots J'_\ell} \sum_{L_1\cdots L_\ell} \prod_{i=1}^{\ell} \left(m_i\left((1-n_i)\delta_{J_i}^{\ J'_i} + n_i\right) + (1-m_i)\right)\left((1-m_i) + m_i\delta_{J'_i}^{\ L_{v_i}}\right) P'_{L_1\cdots L_\ell} \quad (58) \\ &= \sum_{J'_1\cdots J'_\ell} \prod_{i=1}^{\ell} \left(m_i\left((1-n_i)\delta_{J_i}^{\ J'_i} + n_i\right) + (1-m_i)\right) P'_{J'^{\bar{r}}} \quad (59) \end{aligned}$$

where now, $J^{\bar{r}n}$ is given by (47) and, once again, the introduction of m_i and $(1-m_i)$ in (58) allows us to extend the range of the first product to ℓ.

We are now in a position to write the fundamental evolution equation for an arbitrary schema I^n

Coarse-grained schema evolution equation

The expected frequency of a schema I^n at the next generation in a generational GA with arbitrary selection with replacement, mutation and generalised recombination is

$$E(P_{I^n}(t+1)) = \sum_{J^n} M_{I^n}^{\ J^n}[(1-p_{xo})P'_{J^n}(t) + p_{xo}\sum_{\mathbf{r}} p_c(\mathbf{r})P'_{J^{rn}}(t)P'_{J^{\bar{r}n}}(t)] \quad (60)$$

where $J^{rn} = \bigcap_{i=1}^{\ell}(*^{v_i-1}(\bar{n}_i \bullet (\bar{m}_i \bullet J_i)) *^{\ell-v_i})$ and $J^{\bar{r}n} = \bigcap_{i=1}^{\ell}(*^{v_i-1}(\bar{n}_i \bullet (m_i \bullet I_i)) *^{\ell-v_i})$ are the Building Block Schemata for the schema J^n with respect to the GCM $\mathbf{r} = (m, v)$.

Comparing (38) and (60), we see that the functional form of the dynamics for strings and schemata is the same, and hence the dynamics is covariant under a coarse graining. This covariance, or functional "self-similarity", of the equations can be observed nicely by noting that

$$\begin{aligned} \sum_{J'_1\cdots J'_\ell} \prod_{i=1}^{\ell} \left((1-m_i)\left((1-n_i)\delta_{J_i}^{\ J'_i} + n_i\right) + m_i\right) \sum_{K_1\cdots K_\ell} \prod_{i=1}^{\ell} \left((1-m_i)\delta_{J'_i}^{\ K_{v_i}} + m_i\right) P'_{K_1\cdots K_\ell} \\ = \sum_{K_1\cdots K_\ell} \prod_{i=1}^{\ell} \left((1-m'_i)\delta_{J_i}^{\ K_{v_i}} + m'_i\right) P'_{K_1\cdots K_\ell} \end{aligned}$$

where $(1-m'_i) = (1-n_i)(1-m_i)$ and $m'_i = n_i(1-m_i) + m_i$, and we have once again used the projection operator properties of (31). Note that $(1-m'_i) = 1$ if and only if $m_i = 0$ *and* $n_i = 0$, while $m'_i = 1$ if and only if $m_i = 1$ *or* $n_i = 1$ and $m_i = 0$ which manifests the constraint that one cannot coarse grain a locus that has already been coarse grained. The content of this equation is that the coarse graining onto the schema Building Block J^{rn} can be achieved in two equivalent ways: one involves a composition of coarse grainings, wherein strings, K, can be coarse grained via a recombination pair $\mathbf{r}$, that involves a mask m, to yield a Building Block J^r of a string J, which can be further coarse grained using a mask n, to give a Building Block, J^{rn}, of the schema J^n. The second, equivalent, way of achieving the coarse graining, is to implement it using a composite mask, m' that involves both m and n and whose elements are as above.

The coarse graining operators (48) form a semi-group, and are actually just an explicit representation of the renormalization group [28, 2]. Thus,

$$\mathcal{C}_I^{nm} = \mathcal{C}_I^n \mathcal{C}_I^m \quad (61)$$

which, in terms of the projection operators, is

$$\left(\sum_{I_1\cdots I_\ell}\prod_{i=1}^{\ell}\left((1-m_i')\delta_{I_i}^{\ I_i'}+m_i'\right)\right)$$
$$=\left(\sum_{I_1'\cdots I_\ell'}\prod_{i=1}^{\ell}\left((1-n_i)\delta_{I_i}^{\ I_i'}+n_i\right)\right)\left(\sum_{I_1''\cdots I_\ell''}\prod_{i=1}^{\ell}\left((1-m_i)\delta_{I_i'}^{\ I_i''}+m_i\right)\right) \tag{62}$$

where, once again, $(1-m_i')=(1-n_i)(1-m_i)$ and $m_i'=n_i(1-m_i)+m_i$.

In analogy with (63) one may define the effective fitness of the schema I^n via

$$E(P_{I^n}(t+1))=\frac{f_{I^n}^{\rm eff}(t)}{\bar{f}(t)}P_{I^n}(t) \tag{63}$$

Thus,

$$f_{I^n}^{\rm eff}=\frac{\bar{f}}{P_{I^n}}\sum_{J^n}M_{I^n}^{\ J^n}\left[(1-p_{xo})\frac{f_{J^n}}{\bar{f}}P_{J^n}+p_{xo}\sum_{\mathbf{r}}p_c(\mathbf{r})\frac{f_{J^{rn}}}{\bar{f}}P_{J^{rn}}\frac{f_{J^{\bar{r}n}}}{\bar{f}}P_{J^{\bar{r}n}}\right] \tag{64}$$

from which we can enunciate a generalisation of the Exact Schema Theorem for the canonical genetic algorithm. Equation (60) is, in fact, just this schema theorem which, formulated in terms of effective fitness, states that: a genetic algorithm evolving under the action of the operators selection, mutation and generalised recombination gives an increasing number of trials to *effectively* fit schemata. Equation (60), and indeed, its equivalent (38), both clearly show that the population evolves in a analogous fashion to that with purely homologous crossover - by combining lower order Building Blocks into higher order ones which in their turn form yet higher order blocks etc. The end points of this process are the strings themselves, the highest order objects possible, and one-schemata, the lowest order objects possible.

5 Examples

To illustrate some of the general features of the coarse grained equations we will first consider some general properties of the equations for $\ell=2$ and $\ell=3$ for an arbitrary fitness landscape. We will subsequently give a solution to the two-locus case for no selection and an infinite population. As our main interest in this paper is evolution under generalised recombination we will set the mutation rate $p_m=0$ in the rest of the paper. We will also for simplicity in this section set $p_{xo}=1$.

5.1 $\ell=2$

The evolution equations for a generic string of length $\ell=2$ from equation (38) are,

$$\begin{aligned}E(P_{ij}(t+1))&=p_{11}P'_{i*}\delta_{ij}+p_{12}P'_{ij}+p_{13}P'_{i*}P'_{j*}\\&+p_{14}P'_{i*}P'_{*j}+p_{21}P'_{ji}+p_{22}P'_{*i}\delta_{ij}\\&+p_{23}P'_{*i}P'_{j*}+p_{24}P'_{*i}P'_{*j}+p_{31}P'_{j*}P'_{i*}\\&+p_{32}P'_{*j}P'_{i*}+p_{33}P'_{i*}\delta_{ij}+p_{34}P'_{ij}\\&+p_{41}P'_{j*}P'_{*i}+p_{42}P'_{*j}P'_{*i}+p_{43}P'_{ji}+p_{44}P'_{*i}\delta_{ij}\end{aligned} \tag{65}$$

If one replaces the generic, arbitrary alleles i and j with particular values from Ω the Kronecker deltas are zero or one according to whether one is considering the heterogeneous case $i\neq j$, or the homogeneous one $i=j$ then the equations simplify further. For example, if $i=j=1$ and all GCMs have equal probability ($p_{ij}=1/16$), we obtain

$$E(P_{11}(t+1))=\frac{1}{8}\left(P_{1*}'^{\,2}+P'_{1*}+P'_{1*}P'_{*1}+P_{*1}'^{\,2}+P'_{11}+P'_{*1}\right) \tag{66}$$

Notice that in order to solve for the dynamics of the strings here we need to solve first for the dynamics of the Building Blocks `i*`, `*i`, `j*` and `*j`. Of course, only two of these one-schemata are independent.

As an example, the evolution equation for the schema `i*` (a Building Block for `ij`) is

$$E(P_{\mathtt{i}*}(t+1)) = (p_{1*} + p_{3*})P'_{\mathtt{i}*} + (p_{2*} + p_{4*})P'_{*\mathtt{i}} \tag{67}$$

where $p_{a*} = \sum_b p_{ab}$ represents a coarse graining of the GCMs themselves. We immediately notice an important difference with the case of homologous crossover, where in (67) $p_{2*} = p_{4*} = 0$, and hence there is no contribution from the schema $*\mathtt{i}$ and therefore $E(P_{\mathtt{i}*}(t+1)) = P'_{\mathtt{i}*}(t)$. Thus, we see that the dynamics of the most elementary and fundamental objects under recombination - one-schemata - are quite different in the presence of generalised recombination.

5.2 $\ell = 3$

Turning now to the case of $\ell = 3$: The general form of the evolution equations for the generic string `ijk` is given in Figure 1. This includes 216 terms – a number that, although quite big, is only a tiny fraction of the 13,824 terms one would get in the absence of coarse graining! We have explicitly included this somewhat unwieldy looking equation, as a thorough study of its content is most instructive as far as understanding the differences between homologous and generalised recombination is concerned. In fact, all the most important phenomena that can occur as a result of generalised recombination are already present at this level.

$$
\begin{aligned}
&E(P_{\mathtt{ijk}}(t+1)) \\
&= p_{111}P'_{\mathtt{i}**}\delta_{\mathtt{ij}}\delta_{\mathtt{ik}} + p_{112}P'_{\mathtt{ik}*}\delta_{\mathtt{ij}} + p_{113}P'_{\mathtt{i}*\mathtt{k}}\delta_{\mathtt{ij}} + p_{114}P'_{\mathtt{i}**}P'_{\mathtt{k}**}\delta_{\mathtt{ij}} \\
&+ p_{115}P'_{\mathtt{i}**}P'_{*\mathtt{k}*}\delta_{\mathtt{ij}} + p_{116}P'_{\mathtt{i}**}P'_{**\mathtt{k}}\delta_{\mathtt{ij}} + p_{121}P'_{\mathtt{ij}*}\delta_{\mathtt{ik}} + p_{122}P'_{\mathtt{ij}*}\delta_{\mathtt{jk}} \\
&+ p_{123}P'_{\mathtt{ijk}} + p_{124}P'_{\mathtt{ij}*}P'_{\mathtt{k}**} + p_{125}P'_{\mathtt{ij}*}P'_{*\mathtt{k}*} + p_{126}P'_{\mathtt{ij}*}P'_{**\mathtt{k}} \\
&+ p_{131}P'_{\mathtt{i}*\mathtt{j}}\delta_{\mathtt{ik}} + p_{132}P'_{\mathtt{ikj}} + p_{133}P'_{\mathtt{i}*\mathtt{j}}\delta_{\mathtt{jk}} \\
&+ p_{134}P'_{\mathtt{i}*\mathtt{j}}P'_{\mathtt{k}**} + p_{135}P'_{\mathtt{i}*\mathtt{j}}P'_{*\mathtt{k}*} + p_{136}P'_{\mathtt{i}*\mathtt{j}}P'_{**\mathtt{k}} + p_{141}P'_{\mathtt{i}**}P'_{\mathtt{j}**}\delta_{\mathtt{ik}} \\
&+ p_{142}P'_{\mathtt{ik}*}P'_{\mathtt{j}**} + p_{143}P'_{\mathtt{i}*\mathtt{k}}P'_{\mathtt{j}**} + p_{144}P'_{\mathtt{i}**}P'_{\mathtt{j}**}\delta_{\mathtt{jk}} + p_{145}P'_{\mathtt{i}**}P'_{\mathtt{jk}*} \\
&+ p_{146}P'_{\mathtt{i}**}P'_{\mathtt{j}*\mathtt{k}} + p_{151}P'_{\mathtt{i}**}P'_{*\mathtt{j}*}\delta_{\mathtt{ik}} + p_{152}P'_{\mathtt{ik}*}P'_{*\mathtt{j}*} + p_{153}P'_{\mathtt{i}*\mathtt{k}}P'_{*\mathtt{j}*} \\
&+ p_{154}P'_{\mathtt{i}**}P'_{\mathtt{kj}*} + p_{155}P'_{\mathtt{i}**}P'_{*\mathtt{j}*}\delta_{\mathtt{jk}} + p_{156}P'_{\mathtt{i}**}P'_{*\mathtt{jk}} + p_{161}P'_{\mathtt{i}**}P'_{**\mathtt{j}}\delta_{\mathtt{ik}} \\
&+ p_{162}P'_{\mathtt{ik}*}P'_{**\mathtt{j}} + p_{163}P'_{\mathtt{i}*\mathtt{k}}P'_{**\mathtt{j}} + p_{164}P'_{\mathtt{i}**}P'_{\mathtt{k}*\mathtt{j}} + p_{165}P'_{\mathtt{i}**}P'_{*\mathtt{kj}} \\
&+ p_{166}P'_{\mathtt{i}**}P'_{**\mathtt{j}}\delta_{\mathtt{jk}} + p_{211}P'_{\mathtt{ji}*}\delta_{\mathtt{jk}} + p_{212}P'_{\mathtt{ji}*}\delta_{\mathtt{ik}} + p_{213}P'_{\mathtt{jik}} \\
&+ p_{214}P'_{\mathtt{ji}*}P'_{\mathtt{k}**} + p_{215}P'_{\mathtt{ji}*}P'_{*\mathtt{k}*} + p_{216}P'_{\mathtt{ji}*}P'_{**\mathtt{k}} + p_{221}P'_{\mathtt{ki}*}\delta_{\mathtt{ij}} \\
&+ p_{222}P'_{*\mathtt{i}*}\delta_{\mathtt{ij}}\delta_{\mathtt{ik}} + p_{223}P'_{*\mathtt{ik}}\delta_{\mathtt{ij}} + p_{224}P'_{*\mathtt{i}*}P'_{\mathtt{k}**}\delta_{\mathtt{ij}} + p_{225}P'_{*\mathtt{i}*}P'_{*\mathtt{k}*}\delta_{\mathtt{ij}} \\
&+ p_{226}P'_{*\mathtt{i}*}P'_{**\mathtt{k}}\delta_{\mathtt{ij}} + p_{231}P'_{\mathtt{kij}} + p_{232}P'_{*\mathtt{ij}}\delta_{\mathtt{ik}} + p_{233}P'_{*\mathtt{ij}}\delta_{\mathtt{jk}} \\
&+ p_{234}P'_{*\mathtt{ij}}P'_{\mathtt{k}**} + p_{235}P'_{*\mathtt{ij}}P'_{*\mathtt{k}*} + p_{236}P'_{*\mathtt{ij}}P'_{**\mathtt{k}} + p_{241}P'_{\mathtt{ki}*}P'_{\mathtt{j}**} \\
&+ p_{242}P'_{*\mathtt{i}*}P'_{\mathtt{j}**}\delta_{\mathtt{ik}} + p_{243}P'_{*\mathtt{ik}}P'_{\mathtt{j}**} + p_{244}P'_{*\mathtt{i}*}P'_{\mathtt{j}**}\delta_{\mathtt{jk}} + p_{245}P'_{*\mathtt{i}*}P'_{\mathtt{jk}*} \\
&+ p_{246}P'_{*\mathtt{i}*}P'_{\mathtt{j}*\mathtt{k}} + p_{251}P'_{\mathtt{ki}*}P'_{*\mathtt{j}*} + p_{252}P'_{*\mathtt{i}*}P'_{*\mathtt{j}*}\delta_{\mathtt{ik}} + p_{253}P'_{*\mathtt{ik}}P'_{*\mathtt{j}*} \\
&+ p_{254}P'_{*\mathtt{i}*}P'_{\mathtt{kj}*} + p_{255}P'_{*\mathtt{i}*}P'_{*\mathtt{j}*}\delta_{\mathtt{jk}} + p_{256}P'_{*\mathtt{i}*}P'_{*\mathtt{jk}} + p_{261}P'_{\mathtt{ki}*}P'_{**\mathtt{j}} \\
&+ p_{262}P'_{*\mathtt{i}*}P'_{**\mathtt{j}}\delta_{\mathtt{ik}} + p_{263}P'_{*\mathtt{ik}}P'_{**\mathtt{j}} + p_{264}P'_{*\mathtt{i}*}P'_{\mathtt{k}*\mathtt{j}} + p_{265}P'_{*\mathtt{i}*}P'_{*\mathtt{kj}} \\
&+ p_{266}P'_{*\mathtt{i}*}P'_{**\mathtt{j}}\delta_{\mathtt{jk}} + p_{311}P'_{\mathtt{j}*\mathtt{i}}\delta_{\mathtt{jk}} + p_{312}P'_{\mathtt{jki}} + p_{313}P'_{\mathtt{j}*\mathtt{i}}\delta_{\mathtt{ik}} \\
&+ p_{314}P'_{\mathtt{j}*\mathtt{i}}P'_{\mathtt{k}**} + p_{315}P'_{\mathtt{j}*\mathtt{i}}P'_{*\mathtt{k}*} + p_{316}P'_{\mathtt{j}*\mathtt{i}}P'_{**\mathtt{k}} + p_{321}P'_{\mathtt{kji}} \\
&+ p_{322}P'_{*\mathtt{ji}}\delta_{\mathtt{jk}} + p_{323}P'_{*\mathtt{ji}}\delta_{\mathtt{ik}} + p_{324}P'_{*\mathtt{ji}}P'_{\mathtt{k}**} + p_{325}P'_{*\mathtt{ji}}P'_{*\mathtt{k}*} \\
&+ p_{326}P'_{*\mathtt{ji}}P'_{**\mathtt{k}} + p_{331}P'_{\mathtt{k}*\mathtt{i}}\delta_{\mathtt{ij}} + p_{332}P'_{*\mathtt{ki}}\delta_{\mathtt{ij}} + p_{333}P'_{**\mathtt{i}}\delta_{\mathtt{ij}}\delta_{\mathtt{ik}} \\
&+ p_{334}P'_{**\mathtt{i}}P'_{\mathtt{k}**}\delta_{\mathtt{ij}} + p_{335}P'_{**\mathtt{i}}P'_{*\mathtt{k}*}\delta_{\mathtt{ij}} + p_{336}P'_{**\mathtt{i}}P'_{**\mathtt{k}}\delta_{\mathtt{ij}} + p_{341}P'_{\mathtt{k}*\mathtt{i}}P'_{\mathtt{j}**} \\
&+ p_{342}P'_{*\mathtt{ki}}P'_{\mathtt{j}**} + p_{343}P'_{**\mathtt{i}}P'_{\mathtt{j}**}\delta_{\mathtt{ik}} + p_{344}P'_{**\mathtt{i}}P'_{\mathtt{j}**}\delta_{\mathtt{jk}} + p_{345}P'_{**\mathtt{i}}P'_{\mathtt{jk}*} \\
&+ p_{346}P'_{**\mathtt{i}}P'_{\mathtt{j}*\mathtt{k}} + p_{351}P'_{\mathtt{k}*\mathtt{i}}P'_{*\mathtt{j}*} + p_{352}P'_{*\mathtt{ki}}P'_{*\mathtt{j}*} + p_{353}P'_{**\mathtt{i}}P'_{*\mathtt{j}*}\delta_{\mathtt{ik}} \\
&+ p_{354}P'_{**\mathtt{i}}P'_{\mathtt{kj}*} + p_{355}P'_{**\mathtt{i}}P'_{*\mathtt{j}*}\delta_{\mathtt{jk}} + p_{356}P'_{**\mathtt{i}}P'_{*\mathtt{jk}} + p_{361}P'_{\mathtt{k}*\mathtt{i}}P'_{**\mathtt{j}} \\
&+ p_{362}P'_{*\mathtt{ki}}P'_{**\mathtt{j}} + p_{363}P'_{**\mathtt{i}}P'_{**\mathtt{j}}\delta_{\mathtt{ik}} + p_{364}P'_{**\mathtt{i}}P'_{\mathtt{k}*\mathtt{j}} + p_{365}P'_{**\mathtt{i}}P'_{*\mathtt{kj}} \\
&+ p_{366}P'_{**\mathtt{i}}P'_{**\mathtt{j}}\delta_{\mathtt{jk}} + p_{411}P'_{\mathtt{j}**}P'_{\mathtt{i}**}\delta_{\mathtt{jk}} + p_{412}P'_{\mathtt{jk}*}P'_{\mathtt{i}**} + p_{413}P'_{\mathtt{j}*\mathtt{k}}P'_{\mathtt{i}**} \\
&+ p_{414}P'_{\mathtt{j}**}P'_{\mathtt{i}**}\delta_{\mathtt{ik}} + p_{415}P'_{\mathtt{j}**}P'_{\mathtt{ik}*} + p_{416}P'_{\mathtt{j}**}P'_{\mathtt{i}*\mathtt{k}} + p_{421}P'_{\mathtt{kj}*}P'_{\mathtt{i}**} \\
&+ p_{422}P'_{*\mathtt{j}*}P'_{\mathtt{i}**}\delta_{\mathtt{jk}} + p_{423}P'_{*\mathtt{jk}}P'_{\mathtt{i}**} + p_{424}P'_{*\mathtt{j}*}P'_{\mathtt{i}**}\delta_{\mathtt{ik}} + p_{425}P'_{*\mathtt{j}*}P'_{\mathtt{ik}*} \\
&+ p_{426}P'_{*\mathtt{j}*}P'_{\mathtt{i}*\mathtt{k}} + p_{431}P'_{\mathtt{k}*\mathtt{j}}P'_{\mathtt{i}**} + p_{432}P'_{*\mathtt{kj}}P'_{\mathtt{i}**} + p_{433}P'_{**\mathtt{j}}P'_{\mathtt{i}**}\delta_{\mathtt{jk}} \\
&+ p_{434}P'_{**\mathtt{j}}P'_{\mathtt{i}**}\delta_{\mathtt{ik}} + p_{435}P'_{**\mathtt{j}}P'_{\mathtt{ik}*} + p_{436}P'_{**\mathtt{j}}P'_{\mathtt{i}*\mathtt{k}} + p_{441}P'_{\mathtt{k}**}P'_{\mathtt{i}**}\delta_{\mathtt{ij}} \\
&+ p_{442}P'_{*\mathtt{k}*}P'_{\mathtt{i}**}\delta_{\mathtt{ij}} + p_{443}P'_{**\mathtt{k}}P'_{\mathtt{i}**}\delta_{\mathtt{ij}} + p_{444}P'_{\mathtt{i}**}\delta_{\mathtt{ij}}\delta_{\mathtt{ik}} + p_{445}P'_{\mathtt{ik}*}\delta_{\mathtt{ij}} \\
&+ p_{446}P'_{\mathtt{i}*\mathtt{k}}\delta_{\mathtt{ij}} + p_{451}P'_{\mathtt{k}**}P'_{\mathtt{ij}*} + p_{452}P'_{*\mathtt{k}*}P'_{\mathtt{ij}*} + p_{453}P'_{**\mathtt{k}}P'_{\mathtt{ij}*} \\
&+ p_{454}P'_{\mathtt{ij}*}\delta_{\mathtt{ik}} + p_{455}P'_{\mathtt{ij}*}\delta_{\mathtt{jk}} + p_{456}P'_{\mathtt{ijk}} + p_{461}P'_{\mathtt{k}**}P'_{\mathtt{i}*\mathtt{j}} \\
&+ p_{462}P'_{*\mathtt{k}*}P'_{\mathtt{i}*\mathtt{j}} + p_{463}P'_{**\mathtt{k}}P'_{\mathtt{i}*\mathtt{j}} + p_{464}P'_{\mathtt{i}*\mathtt{j}}\delta_{\mathtt{ik}} + p_{465}P'_{\mathtt{ikj}} \\
&+ p_{466}P'_{\mathtt{i}*\mathtt{j}}\delta_{\mathtt{jk}} + p_{511}P'_{\mathtt{j}**}P'_{*\mathtt{i}*}\delta_{\mathtt{jk}} + p_{512}P'_{\mathtt{jk}*}P'_{*\mathtt{i}*} + p_{513}P'_{\mathtt{j}*\mathtt{k}}P'_{*\mathtt{i}*} \\
&+ p_{514}P'_{\mathtt{j}**}P'_{\mathtt{ki}*} + p_{515}P'_{\mathtt{j}**}P'_{*\mathtt{i}*}\delta_{\mathtt{ik}} + p_{516}P'_{\mathtt{j}**}P'_{*\mathtt{ik}} + p_{521}P'_{\mathtt{kj}*}P'_{*\mathtt{i}*} \\
&+ p_{522}P'_{*\mathtt{j}*}P'_{*\mathtt{i}*}\delta_{\mathtt{jk}} + p_{523}P'_{*\mathtt{jk}}P'_{*\mathtt{i}*} + p_{524}P'_{*\mathtt{j}*}P'_{\mathtt{ki}*} + p_{525}P'_{*\mathtt{j}*}P'_{*\mathtt{i}*}\delta_{\mathtt{ik}} \\
&+ p_{526}P'_{*\mathtt{j}*}P'_{*\mathtt{ik}} + p_{531}P'_{\mathtt{k}*\mathtt{j}}P'_{*\mathtt{i}*} + p_{532}P'_{*\mathtt{kj}}P'_{*\mathtt{i}*} + p_{533}P'_{**\mathtt{j}}P'_{*\mathtt{i}*}\delta_{\mathtt{jk}} \\
&+ p_{534}P'_{**\mathtt{j}}P'_{\mathtt{ki}*} + p_{535}P'_{**\mathtt{j}}P'_{*\mathtt{i}*}\delta_{\mathtt{ik}} + p_{536}P'_{**\mathtt{j}}P'_{*\mathtt{ik}} + p_{541}P'_{\mathtt{k}**}P'_{\mathtt{ji}*} \\
&+ p_{542}P'_{*\mathtt{k}*}P'_{\mathtt{ji}*} + p_{543}P'_{**\mathtt{k}}P'_{\mathtt{ji}*} + p_{544}P'_{\mathtt{ji}*}\delta_{\mathtt{jk}} + p_{545}P'_{\mathtt{ji}*}\delta_{\mathtt{ik}} \\
&+ p_{546}P'_{\mathtt{jik}} + p_{551}P'_{\mathtt{k}**}P'_{*\mathtt{i}*}\delta_{\mathtt{ij}} + p_{552}P'_{*\mathtt{k}*}P'_{*\mathtt{i}*}\delta_{\mathtt{ij}} + p_{553}P'_{**\mathtt{k}}P'_{*\mathtt{i}*}\delta_{\mathtt{ij}} \\
&+ p_{554}P'_{\mathtt{ki}*}\delta_{\mathtt{ij}} + p_{555}P'_{*\mathtt{i}*}\delta_{\mathtt{ij}}\delta_{\mathtt{ik}} + p_{556}P'_{*\mathtt{ik}}\delta_{\mathtt{ij}} + p_{561}P'_{\mathtt{k}**}P'_{*\mathtt{ij}} \\
&+ p_{562}P'_{*\mathtt{k}*}P'_{*\mathtt{ij}} + p_{563}P'_{**\mathtt{k}}P'_{*\mathtt{ij}} + p_{564}P'_{\mathtt{kij}} + p_{565}P'_{*\mathtt{ij}}\delta_{\mathtt{ik}} \\
&+ p_{566}P'_{*\mathtt{ij}}\delta_{\mathtt{jk}} + p_{611}P'_{\mathtt{j}**}P'_{**\mathtt{i}}\delta_{\mathtt{jk}} + p_{612}P'_{\mathtt{jk}*}P'_{**\mathtt{i}} + p_{613}P'_{\mathtt{j}*\mathtt{k}}P'_{**\mathtt{i}} \\
&+ p_{614}P'_{\mathtt{j}**}P'_{\mathtt{k}*\mathtt{i}} + p_{615}P'_{\mathtt{j}**}P'_{*\mathtt{ki}} + p_{616}P'_{\mathtt{j}**}P'_{**\mathtt{i}}\delta_{\mathtt{ik}} + p_{621}P'_{\mathtt{kj}*}P'_{**\mathtt{i}} \\
&+ p_{622}P'_{*\mathtt{j}*}P'_{**\mathtt{i}}\delta_{\mathtt{jk}} + p_{623}P'_{*\mathtt{jk}}P'_{**\mathtt{i}} + p_{624}P'_{*\mathtt{j}*}P'_{\mathtt{k}*\mathtt{i}} + p_{625}P'_{*\mathtt{j}*}P'_{*\mathtt{ki}} \\
&+ p_{626}P'_{*\mathtt{j}*}P'_{**\mathtt{i}}\delta_{\mathtt{ik}} + p_{631}P'_{\mathtt{k}*\mathtt{j}}P'_{**\mathtt{i}} + p_{632}P'_{*\mathtt{kj}}P'_{**\mathtt{i}} + p_{633}P'_{**\mathtt{j}}P'_{**\mathtt{i}}\delta_{\mathtt{jk}} \\
&+ p_{634}P'_{**\mathtt{j}}P'_{\mathtt{k}*\mathtt{i}} + p_{635}P'_{**\mathtt{j}}P'_{*\mathtt{ki}} + p_{636}P'_{**\mathtt{j}}P'_{**\mathtt{i}}\delta_{\mathtt{ik}} + p_{641}P'_{\mathtt{k}**}P'_{\mathtt{j}*\mathtt{i}} \\
&+ p_{642}P'_{*\mathtt{k}*}P'_{\mathtt{j}*\mathtt{i}} + p_{643}P'_{**\mathtt{k}}P'_{\mathtt{j}*\mathtt{i}} + p_{644}P'_{\mathtt{j}*\mathtt{i}}\delta_{\mathtt{jk}} + p_{645}P'_{\mathtt{jki}} \\
&+ p_{646}P'_{\mathtt{j}*\mathtt{i}}\delta_{\mathtt{ik}} + p_{651}P'_{\mathtt{k}**}P'_{*\mathtt{ji}} + p_{652}P'_{*\mathtt{k}*}P'_{*\mathtt{ji}} + p_{653}P'_{**\mathtt{k}}P'_{*\mathtt{ji}} \\
&+ p_{654}P'_{\mathtt{kji}} + p_{655}P'_{*\mathtt{ji}}\delta_{\mathtt{jk}} + p_{656}P'_{*\mathtt{ji}}\delta_{\mathtt{ik}} + p_{661}P'_{\mathtt{k}**}P'_{**\mathtt{i}}\delta_{\mathtt{ij}} + p_{662}P'_{*\mathtt{k}*}P'_{**\mathtt{i}}\delta_{\mathtt{ij}} \\
&+ p_{663}P'_{**\mathtt{k}}P'_{**\mathtt{i}}\delta_{\mathtt{ij}} + p_{664}P'_{\mathtt{k}*\mathtt{i}}\delta_{\mathtt{ij}} + p_{665}P'_{*\mathtt{ki}}\delta_{\mathtt{ij}} + p_{666}P'_{**\mathtt{i}}\delta_{\mathtt{ij}}\delta_{\mathtt{ik}}
\end{aligned}
$$

Figure 1: Evolution equation for a 3-locus strings with selection and generalised recombination.

Note that the expected frequency of $\mathtt{ijk}$ at generation $t+1$ depends not only on its frequency at generation t but is a linear function of the selection probabilities of not only that string but all its permutations, as well as a (generally) quadratic function of the selection probabilities of the lower order schemata (Building Blocks) that compose it, i.e.

$$E(P_{\mathtt{ijk}}(t+1)) = p_{123}P'_{\mathtt{ijk}} + p_{132}P'_{\mathtt{ikj}} + p_{213}P'_{\mathtt{jik}} + p_{231}P'_{\mathtt{kij}} + p_{312}P'_{\mathtt{jki}} + p_{321}P'_{\mathtt{kji}} + b(t) \quad (68)$$

Again, in order to solve for the string dynamics we need to have the dynamics of the Building Blocks that determine the driving term $b(t)$. One of the Building Blocks, for example, is $\mathtt{ij*}$, the evolution equation of which is

$$\begin{aligned}
E(P_{\mathtt{ij}*}(t+1)) &= p_{11*}P'_{\mathtt{i}**}\delta_{\mathtt{ij}} + p_{12*}P'_{\mathtt{aj}*} \\
&+ p_{13*}P'_{\mathtt{i}*\mathtt{j}} + p_{14*}P'_{\mathtt{i}**}P'_{\mathtt{j}**} + p_{15*}P'_{\mathtt{i}**}P'_{*\mathtt{j}*} + p_{16*}P'_{\mathtt{i}**}P'_{**\mathtt{j}} \\
&+ p_{21*}P'_{\mathtt{ji}*} + p_{22*}P'_{*\mathtt{i}*}\delta_{\mathtt{ij}} + p_{23*}P'_{*\mathtt{ij}} + p_{24*}P'_{*\mathtt{i}*}P'_{\mathtt{j}**} \\
&+ p_{25*}P'_{*\mathtt{i}*}P'_{*\mathtt{j}*} + p_{26*}P'_{*\mathtt{i}*}P'_{**\mathtt{j}} + p_{31*}P'_{\mathtt{j}*\mathtt{i}} + p_{32*}P'_{*\mathtt{ji}} \\
&+ p_{33*}P'_{**\mathtt{i}}\delta_{\mathtt{ij}} + p_{34*}P'_{**\mathtt{i}}P'_{\mathtt{j}**} + p_{35*}P'_{**\mathtt{i}}P'_{*\mathtt{j}*} + p_{36*}P'_{**\mathtt{i}}P'_{**\mathtt{j}} \\
&+ p_{41*}P'_{\mathtt{j}**}P'_{\mathtt{i}**} + p_{42*}P'_{*\mathtt{j}*}P'_{\mathtt{i}**} + p_{43*}P'_{**\mathtt{j}}P'_{\mathtt{i}**} + p_{44*}P'_{\mathtt{i}**}\delta_{\mathtt{ij}} \\
&+ p_{45*}P'_{\mathtt{ij}*} + p_{46*}P'_{\mathtt{i}*\mathtt{j}} + p_{51*}P'_{\mathtt{j}**}P'_{*\mathtt{i}*} + p_{52*}P'_{*\mathtt{j}*}P'_{*\mathtt{i}*} \\
&+ p_{53*}P'_{**\mathtt{j}}P'_{*\mathtt{i}*} + p_{54*}P'_{\mathtt{ji}*} + p_{55*}P'_{*\mathtt{i}*}\delta_{\mathtt{ij}} + p_{56*}P'_{*\mathtt{ij}} \\
&+ p_{61*}P'_{\mathtt{j}**}P'_{**\mathtt{i}} + p_{62*}P'_{*\mathtt{j}*}P'_{**\mathtt{i}} + p_{63*}P'_{**\mathtt{j}}P'_{**\mathtt{i}} + p_{64*}P'_{\mathtt{j}*\mathtt{i}} \\
&+ p_{65*}P'_{*\mathtt{ji}} + p_{66*}P'_{**\mathtt{i}}\delta_{\mathtt{ij}}
\end{aligned}$$

where we have collected terms involving the same schema and where $p_{ab*} = \sum_c p_{abc}$. Notice that the Building Block $\mathtt{ij*}$, in its turn, will depend on the dynamics of its own Building Blocks, such as $\mathtt{i**}$, the equation for which is

$$E(P_{\mathtt{i}**}(t+1)) = (p_{1**} + p_{4**})P'_{\mathtt{i}**} + (p_{2**} + p_{5**})P'_{*\mathtt{i}*} + (p_{3**} + p_{6**})P'_{**\mathtt{i}} \quad (69)$$

This hierarchical structure is studied at length in [20] in order to investigate the asymptotic behaviour of the dynamics and especially its fixed points. For now though, we turn to the explicit, exact solution of the case $\ell = 2$.

6 Two-Locus Solution

In both population genetics ([1] and references therein) and EC (see for example [6] and [22] two-locus models have played an important role, leading to improved understanding in the context of potentially analytically solvable models. In this section we study generalised recombination in the context of a simple two-locus model with no mutation and no selection. Thus, we choose to initially study only the intrinsic biases of the generalised recombination operator $\mathcal{G}$. As has been seen in previous work this can lead to potentially practical recipes for practitioners [13].

In this context one would like to see if and how the dynamics of GAs, based on generalised recombination, differ from their homologous counterparts. This is naturally a quite complicated question, given that generalised recombination really captures several different basic operators, including homologous recombination, inversion (and more generally - permutations), different types of gene duplication and combinations of these different operators. Here we investigate the solutions to equation (66) in the infinite population limit in the absence of selection, i.e. $P'_{ij}(t) = P_{ij}(t)$, where $P_{ij}(t)$ is a probability, and where, for simplicity, we set $p_{xo} = 1$.

In the infinite population limit $E(P_I(t+1)) \to P_I(t+1)$ and then (5) and the equations derived from it become deterministic equations for the string and/or schema proportions and describe the corresponding dynamical system. The results derived from the infinite population model neglect the variance inherent in the dynamics due to limited sampling, an effect which is expected to vary as $N^{-1/2}$, where N is the population size. Hence, for large population sizes or short runs one would expect the analysis below to be an accurate representation of what actually occurs. For small populations one would have to consider directly the Markov chain for this model. The transition

matrix elements that enter in this case would be obtained by using a multinomial distribution with success probabilities given by the right hand side of equation (5). Similarly, one would expect our subsequent analysis to give a good qualitative description of the biases engendered by generalised recombination even if selection is included as long as the selection is weak, as will be illustrated later.

6.1 Building Block Dynamics

In order to solve (66), just as in the case of standard homologous recombination [27], we need to hierarchically solve for the dynamics of the Building Blocks of the genotype ij. These are: $i*$ and $*i$. From (66) we can determine the equations for the Building Block schemata by considering $P_{i*}(t) = \sum_{j=0,1} P_{ij}(t)$ and $P_{*j}(t) = \sum_{i=0,1} P_{ij}(t)$. The equations for the one-schemata are[6]

$$P_{i*}(t+1) = (p_{1*} + p_{3*})P_{i*}(t) + (p_{2*} + p_{4*})P_{*i}(t) \tag{70}$$

$$P_{*i}(t+1) = (p_{*1} + p_{*3})P_{i*}(t) + (p_{*2} + p_{*4})P_{*i}(t) \tag{71}$$

Equations (70) and (71) form a coupled linear system, similar to that of mutation in a one-locus system. To solve these equations we need to determine the eigenvalues and eigenvectors of the matrix

$$\mathbf{W} \equiv \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} (p_{1*} + p_{3*}) & (p_{2*} + p_{4*}) \\ (p_{*1} + p_{*3}) & (p_{*2} + p_{*4}) \end{pmatrix} \tag{72}$$

In the case of pure mask-based recombination, only p_{12}, p_{34}, p_{14} and p_{32} are non-zero. Hence, the evolution of the schema $i*$ is independent of the schema $*i$, i.e. the two equations decouple, giving as solution $P_{i*}(t) = P_{*i}(0)$ and $P_{*i}(t) = P_{*i}(0)$. More generally, the eigenvalues of matrix (72) are

$$\lambda_{\pm} = \frac{(a+d)}{2} \pm \frac{1}{2}((a-d)^2 + 4bc)^{1/2} \tag{73}$$

with corresponding normalised eigenvectors

$$\mathbf{e}_+ \equiv \begin{pmatrix} e_{+1} \\ e_{+2} \end{pmatrix} = N_+ \begin{pmatrix} b \\ (\lambda_+ - a) \end{pmatrix} \tag{74}$$

$$\mathbf{e}_- \equiv \begin{pmatrix} e_{-1} \\ e_{-2} \end{pmatrix} = N_- \begin{pmatrix} b \\ (\lambda_- - a) \end{pmatrix} \tag{75}$$

where $N_+ = ((\lambda_+ - a)^2 + b^2)^{-1/2}$ and $N_- = ((\lambda_- - a)^2 + b^2)^{-1/2}$ are normalisation constants. The transformation matrix $\mathbf{\Lambda} \equiv (\mathbf{e}_+ \mathbf{e}_-)^{-1}$, formed from the eigenvectors (74) and (75), diagonalises $\mathbf{W}$ and rotates the vector $\mathbf{P}(t) = (P_{1*} P_{*1})^T \longrightarrow (\tilde{P}_+(t)\tilde{P}_-(t))^T$ such that the diagonalized equations can be immediately integrated then rotated back to the original schema basis to find

$$P_{i*}(t) = Det(\Lambda^{-1})((e_{+1}\lambda_+^t e_{-2} - e_{-1}\lambda_-^t e_{+2})P_{i*}(0) + (-e_{+1}\lambda_+^t e_{-1} + e_{-1}\lambda_-^t e_{+1})P_{*i}(0)) \tag{76}$$

$$P_{*i}(t) = Det(\Lambda^{-1})((e_{-2}\lambda_+^t e_{+2} - e_{+2}\lambda_-^t e_{-2})P_{i*}(0) + (e_{+1}\lambda_+^t e_{-2} - e_{-1}\lambda_-^t e_{+2})P_{*i}(0)) \tag{77}$$

from which one may determine, for instance, the asymptotic behaviour as $t \to \infty$. As P_{i*} and P_{*i} are both probabilities, then $\lambda_{\pm} \leq 1$ and, in fact, one eigenvalue must be unity due to the row stochasticity of the matrix $\mathbf{W}$, as can be seen from equation (72) noting that, as $\sum_{i=1}^4 p_{i*} = \sum_{i=1}^4 p_{*i} = 1$, then $a + b = c + d = 1$. Substituting these constraints into (73) one finds

$$\lambda_+ = 1 \tag{78}$$

$$\lambda_- = \frac{1}{2}(a + d - b - c) = \frac{1}{2}[(p_{14} - p_{41}) + (p_{32} - p_{23}) + (p_{34} - p_{43})] \tag{79}$$

[6]Note that i and j are purely symbolic values so that P_{i*} and P_{*i} are sufficient to cover the 4 possibilities P_{1*}, P_{0*}, P_{*1} and P_{*0}.

with corresponding eigenvectors

$$\mathbf{e}_+ = 2^{-1/2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{80}$$

$$\mathbf{e}_- = (b^2 + c^2)^{-1/2} \begin{pmatrix} b \\ -c \end{pmatrix} \tag{81}$$

Hence, the solutions (76) and (77) simplify to

$$P_{i*}(t) = \frac{(cP_{i*}(0) + bP_{*i}(0))}{(b+c)} + \frac{b(a-c)^t}{b+c}(P_{i*}(0) - P_{*i}(0)) \tag{82}$$

$$P_{*i}(t) = \frac{(cP_{i*}(0) + bP_{*i}(0))}{(b+c)} - \frac{c(a-c)^t}{b+c}(P_{i*}(0) - P_{*i}(0)) \tag{83}$$

From this solution we can examine the fixed point. As $|a-c| \leq 1$, except for $a = 1$, $c = 0$ or $a = 0$ and $c = 1$, the time dependent term vanishes asymptotically, giving as fixed point

$$P^*_{i*} = \lim_{t\to\infty} P_{i*}(t) = \frac{(cP_{i*}(0) + bP_{*i}(0))}{(b+c)} \tag{84}$$

$$P^*_{*i} = \lim_{t\to\infty} P_{*i}(t) = \frac{(cP_{i*}(0) + bP_{*i}(0))}{(b+c)} \tag{85}$$

Interestingly, in this case the fixed point is the same for the schemata $i*$ or $*i$, though this proportion is approached from opposite directions. The behaviour of the transients on approaching the fixed point also depends sensitively on the value of $(a-c)$.[7] If $(a-c) > 0$ then the fixed point is approached monotonically. However, for $(a-c) < 0$ the factor $(-1)^t$ implies the presence of oscillations. However, as $|a-c| \leq 1$ these oscillations are damped and vanish asymptotically. We will now consider some particular cases of interest.

1. $b = c = 0$ - in this case $a = d = 1$, $\lambda_\pm = 1$ and $P^*_{i*} = P_{i*}(0)$, $P^*_{*i} = P_{*i}(0)$; thus the initial proportions are preserved. This type of recombination is homologous and preserves gene frequencies at a given locus.

2. $a = d = 0$ - here, $b = c = 1$, $\lambda_\pm = 1,\ -1$ and the associated eigenvectors are $\mathbf{e}_+ = (1/\sqrt{2})(1\ 1)^T$ and $\mathbf{e}_- = (1/\sqrt{2})(1\ -1)^T$. There is now no fixed point, but rather a cycle of period two, where $P_{i*}(t) = P_{i*}(0)$ for t even and $P_{*i}(0)$ for t odd. Similarly, $P_{*i}(t) = P_{*i}(0)$ for t even and $P_{i*}(0)$ for t odd. This leads to lateral diffusion of alleles along the string from one genetic locus to another.

3. $a = b = c = d = 1/2$ - in this case $\lambda_\pm = 1,\ 0$ with the same eigenvectors as for case 2.. Now there are no oscillations ($(a-c) = 0$) and the fixed point $P^*_{i*} = P^*_{*i} = (P_{i*}(0) + P_{*i}(0))/2$ is reached after only one generation

4. $P_{i*}(0) = P_{*i}(0)$ - when this condition holds, irrespective of the generalised recombination probabilities, this remains a fixed point. This condition is satisfied both at the centre of the simplex [30] as well as at its vertices. It is equivalent to having equal proportions for heterogeneous genotypes.

We have discussed the asymptotic behaviour in terms of the four parameters a, b, c and d. However, we wish to understand the dynamics in terms of the generalised recombination probabilities, p_{ab}. For case 1. above $(p_{1*} + p_{3*}) = (p_{*2} + p_{*4}) = 1$ and $(p_{*1} + p_{*3}) = (p_{2*} + p_{4*}) = 0$, the latter being equivalent to there being no duplication or inversion or any combination that includes them. This means that there are no genetic operators that lead to lateral diffusion of alleles along the string from one genetic locus to another. The resultant fixed point for the one schemata is on

[7] Note that due to the identities $a + b = 1$ and $c + d = 1$ this is equivalent to $(b-d)$.

the Robbins/Geiringer manifold [5]. Similarly, for case 2. we have $(p_{1*} + p_{3*}) = (p_{*2} + p_{*4}) = 0$ and $(p_{*1} + p_{*3}) = (p_{2*} + p_{4*}) = 1$. Under these conditions the only non-zero terms are those associated with inversion. There is no homologous recombination or duplication. Thus, pure inversion without duplication or homologous crossover leads to periodic behaviour.

We may also investigate the biases of a particular genetic operator, investigating the solutions in the absence of the other operators. Thus, for instance, for duplication from one parent, then $p_{ii} \neq 0$ while all other generalised recombination probabilities are zero. In this case $a = c = (p_{11} + p_{33})$ and $b = d = (p_{22} + p_{44}) = (1 - (p_{11} + p_{33}))$. Additionally, we have $\sum_{i=1}^{4} p_{ii} = 1$, i.e. $b + c = a + d = 1$. Hence, there is no transient term and the fixed point is

$$P^*_{i*} = P^*_{*i} = P_{i*}(0) + (p_{22} + p_{44})(P_{*i}(0) - P_{i*}(0)) \tag{86}$$

which is reached after one generation. For cloning, p_{12} and p_{34} are the only non-zero GCMs, hence, $a = d = 1$ and $b = c = 0$. In this case the fixed point is trivially

$$P^*_{i*} = P_{i*}(0) \tag{87}$$

$$P^*_{*i} = P_{*i}(0) \tag{88}$$

For inversion, the only non-zero probabilities are p_{21} and p_{43}. In this case $a = d = 0$ and $b = c = 1$, and the asymptotic behaviour is governed by the two cycle of 2. above with

$$P^*_{i*} = \frac{1}{2}((1 + (-1)^t)P_{i*}(0) + (1 - (-1)^t)P_{*i}(0)) \tag{89}$$

$$P^*_{*i} = \frac{1}{2}((1 - (-1)^t)P_{i*}(0) + (1 + (-1)^t)P_{*i}(0)) \tag{90}$$

For two-parent duplication, the appropriate non-zero recombination probabilities are p_{13}, p_{24}, p_{31} and p_{42}. Hence, $a = c = (p_{13} + p_{31})$ and $b = d = (p_{24} + p_{42})$ with $b + c = a + d = 1$. As $a = c$ there are no transients and the fixed point

$$P^*_{i*} = P^*_{*i} = P_{i*}(0) + (p_{24} + p_{42})(P_{*i}(0) - P_{i*}(0)) \tag{91}$$

is reached after one generation just as in the case of one-parent duplication. Note that this fixed point is of the same form as that found for one parent duplication. For homologous crossover, the non-zero entries in the recombination distribution are p_{14} and p_{32}. In this case $a = d = 1$, $b = c = 0$ and the fixed point is the same as that for cloning. Finally, for crossover and inversion the corresponding recombination distribution entries are p_{41} and p_{23} which implies $a = d = 0$ and $b = c = 1$. In this case the fixed point is the same as that for inversion above.

In Figure 2, using equation (82), we see a graph of the evolution of the one-schema $1*$ for different GRDs. The direct integration of equations (70) and (71) yields exactly the same curves, as expected. The initial condition used is an asymmetric one, where $P_{11}(0) = P_{00}(0) = 0.1$, $P_{01}(0) = 0.6$ and $P_{10}(0) = 0.2$; hence, $P_{1*}(0) = 0.3$. The fixed point behaviour described in points 1. - 4. and equations (86-91) is clearly visible. For one-parent duplication the fixed point is reached after one generation at a value $P^*_{1*} = P_{1*}(0) + P_{*1}(0)/2 = 0.3 + 0.7 = 0.5$. For inversion, one sees the characteristic oscillations between the values 0.3 and 0.7 associated with $P_{1*}(0)$ and $P_{*1}(0)$. For homologous crossover the fixed point is the initial proportion $P_{1*}(0) = 0.3$, i.e. the allele frequency at a given locus is preserved. Finally, considering all GCMs with equal probability - the All curve in Figure 2 - one sees that the system reaches a fixed point in one generation.

The features and fixed points we have just delineated for $\ell = 2$, are also present for $\ell > 2$ and represent qualitatively new phenomena when compared to the normal homologous forms of crossover with which we are familiar. The lateral diffusion of alleles, relative to the homologous case, leads to a fixed point, where for a given offspring locus, the allele frequency at that locus depends not only on the allele frequency at the same parental loci but also on the allele frequencies at other genetic loci. For $\ell > 2$, instead of a pair of linear coupled equations for the one-schemata one has ℓ coupled equations whose solution can be found by solving the corresponding eigensystem.

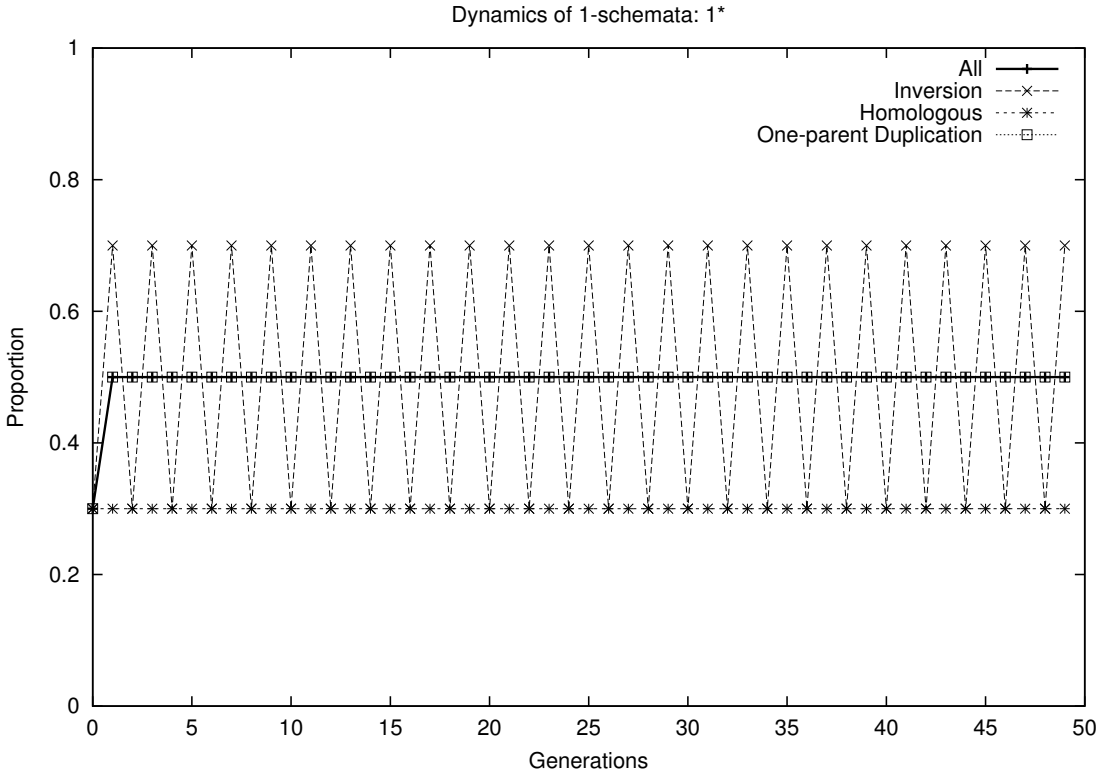

Figure 2: Dynamical evolution of the one-schema $1*$ for different GRDs and no selection

6.2 Solution for Strings

With the solutions for the one-schemata in hand we can now proceed to determine the solutions for the strings themselves from (66) by substituting in the solutions (82) and (83), which are the contributions from the Building Blocks, to yield

$$P_{ij}(t+1) = (1-p_{xo})P_{ij} + p_{xo}(p_{12}+p_{34})P_{ij}(t) + p_{xo}(p_{21}+p_{43})P_{ji}(t) + p_{xo}F_{ij}(t) \tag{92}$$

where

$$F_{ij}(t) = (C_{ij} + D_{ij}(a-c)^t + E_{ij}(a-c)^{2t}) \tag{93}$$

and the matrices C_{ij}, D_{ij} and E_{ij} are given by

$$C_{ij} = (p_{14}+p_{41}+p_{32}+p_{23}+p_{13}+p_{31}+p_{24}+p_{42})A_iA_j + (p_{11}+p_{22}+p_{33}+p_{44})A_i\delta_{ij} \tag{94}$$

$$\begin{aligned} D_{ij} \quad = \quad & [((p_{14}+p_{32})(bA_jB_i - cB_jA_i) + (p_{23}+p_{41})(bA_iB_j - cB_iA_j) \\ & +(p_{13}+p_{31})b(A_jB_i - B_jA_i) - (p_{24}+p_{42})c(A_jB_i - B_jA_i)) \\ & +((p_{11}+p_{33})b - (p_{22}+p_{44})c)B_i\delta_{ij}] \end{aligned} \tag{95}$$

$$E_{ij} = ((p_{13}+p_{31})b^2 + (p_{24}+p_{42})c^2 - (p_{14}+p_{41}+p_{32}+p_{23})bc)B_iB_j \tag{96}$$

where

$$A_i = \frac{(cP_{i*}(0) + bP_{*i}(0))}{(b+c)} \tag{97}$$

$$B_i = \frac{(P_{i*}(0) - P_{*i}(0))}{(b+c)} \tag{98}$$

To solve (92) we need to also have the equation for P_{ji}, which is just (92) with $i \leftrightarrow j$. The matrices C_{ij} and E_{ij} are both symmetric matrices, D_{ij} however is not. Hence, $P_{ji}(t)$ satisfies

$$P_{ji}(t+1) = (1-p_{xo})P_{ji} + p_{xo}(p_{12}+p_{34})P_{ji}(t) + p_{xo}(p_{21}+p_{43})P_{ij}(t) + p_{xo}F_{ji}(t) \tag{99}$$

where

$$F_{ji}(t) = (C_{ij} + D_{ji}(a-c)^t + E_{ij}(a-c)^{2t}) \tag{100}$$

Equations (92) and (99) are linear coupled inhomogeneous first order difference equations and can be solved in an analogous fashion to that of equations (70) and (71) by determining the corresponding eigensystem. Putting $p_{xo} = 1$ the relevant matrix is

$$\mathbf{W}' = \begin{pmatrix} (p_{12}+p_{34}) & (p_{21}+p_{43}) \\ (p_{21}+p_{43}) & (p_{12}+p_{34}) \end{pmatrix} \tag{101}$$

whose eigenvalues and eigenvectors are given by $\lambda_\pm = (p_{12}+p_{34}) \pm (p_{21}+p_{43})$ $\mathbf{e}_+ = 2^{-1/2}(1\ 1)^T$ and $\mathbf{e}_- = 2^{-1/2}(1\ -1)^T$. In the eigenvector basis $\mathbf{P}(t) = (P_{ij}\ P_{ji})^T \rightarrow (\tilde{P}_+(t)\ \tilde{P}_-(t))^T$ such that

$$\tilde{P}_+(t+1) = \lambda_+\tilde{P}_+(t) + \tilde{F}_+(t) \tag{102}$$
$$\tilde{P}_-(t+1) = \lambda_-\tilde{P}_-(t) + \tilde{F}_-(t) \tag{103}$$

where

$$\tilde{F}_+(t) = \frac{1}{2^{1/2}}(F_{ij}(t) + F_{ji}(t)) \tag{104}$$
$$\tilde{F}_-(t) = \frac{1}{2^{1/2}}(F_{ij}(t) - F_{ji}(t)) \tag{105}$$

which can be immediately integrated to yield

$$\tilde{P}_\pm(t) = \lambda_\pm^t\tilde{P}_\pm(0) + \sum_{n=0}^{t-1}\lambda_\pm^{t-n-1}\tilde{F}_\pm(n) \tag{106}$$

Rotating back to the original basis one finds

$$P_{ij}(t) = \frac{1}{2}(\lambda_+^t + \lambda_-^t)P_{ij}(0) + \frac{1}{2}(\lambda_+^t - \lambda_-^t)P_{ji}(0)$$
$$+\frac{1}{2}\sum_{n=0}^{t-1}(\lambda_+^{t-n-1}(F_{ij}(n)+F_{ji}(n)) + \lambda_-^{t-n-1}(F_{ij}(n)-F_{ji}(n))) \tag{107}$$

There now only remains to do the summations to obtain the final answer

$$\begin{aligned} P_{ij}(t) &= \frac{1}{2}(\lambda_+^t + \lambda_-^t)P_{ij}(0) + \frac{1}{2}(\lambda_+^t - \lambda_-^t)P_{ji}(0) \\ &+ C_{ij}\left(\frac{1-\lambda_+^t}{1-\lambda_+}\right) + E_{ij}\left(\frac{(a-c)^{2t}-\lambda_+^t}{(a-c)^2-\lambda_+}\right) \\ &+ \frac{1}{2}(D_{ij}+D_{ji})\left(\frac{(a-c)^t-\lambda_+^t}{a-c-\lambda_+}\right) + \frac{1}{2}(D_{ij}-D_{ji})\left(\frac{(a-c)^t-\lambda_-^t}{a-c-\lambda_-}\right) \end{aligned} \tag{108}$$

Note how this solution has been created - hierarchically, as in the case of homologous crossover [27]. One can solve first for the order one Building Blocks, which then serve as a "source" for construction of order 2 Building Blocks, which serve as a source for the order 3 etc. until one arrives at the strings themselves. The difference here is that inversion can couple different Building Blocks of the same order, unlike the homologous case where they are decoupled.

In the asymptotic limit $t \to \infty$, in the case where the cloning or inversion probabilities are less than one, the fixed point of (109) is

$$P^*_{ij} = \lim_{t\to\infty} P_{ij}(t) = \frac{C_{ij}}{(1-\lambda_+)} \tag{109}$$

Explicitly, in terms of the GRD

$$\begin{aligned} P^*_{ij} &= \left[\frac{(p_{14}+p_{41}+p_{32}+p_{23}+p_{13}+p_{31}+p_{24}+p_{42})}{(1-p_{12}-p_{34}-p_{21}-p_{43})}\right. \\ &\quad \times \frac{((p_{*1}+p_{*3})P_{i*}(0)+(p_{2*}+p_{4*})P_{*i}(0))}{((p_{*1}+p_{*3})+(p_{2*}+p_{4*}))} \\ &\quad \left.\times \frac{((p_{*1}+p_{*3})P_{*j}(0)+(p_{2*}+p_{4*})P_{j*}(0))}{((p_{*1}+p_{*3})+(p_{2*}+p_{4*}))}\right] \\ &+ \left[\frac{(p_{11}+p_{22}+p_{33}+p_{44})}{(1-p_{12}-p_{34}-p_{21}-p_{43})}\right. \\ &\quad \left.\times \frac{((p_{*1}+p_{*3})P_{i*}(0)+(p_{2*}+p_{4*})P_{*i}(0))}{((p_{*1}+p_{*3})+(p_{2*}+p_{4*}))}\delta_{ij}\right] \end{aligned} \tag{110}$$

To get some intuition about the nature of this fixed point we will consider some limits of interest associated with different initial populations and different recombination probability distributions. Beginning with a random initial population, where $P_{ij}(0) = 1/4$, the fixed point becomes

$$P^*_{ij} = \frac{(p_{14}+p_{41}+p_{32}+p_{23}+p_{13}+p_{31}+p_{24}+p_{42})}{4(1-p_{12}-p_{34}-p_{21}-p_{43})} + \frac{(p_{11}+p_{22}+p_{33}+p_{44})}{2(1-p_{12}-p_{34}-p_{21}-p_{43})}\delta_{ij} \tag{111}$$

Only in the case where there is no one-parent gene duplication, i.e. $p_{ii} = 0$, is the centre of the simplex a fixed point. In the presence of one-parent gene duplication homogeneous strings are favoured over their heterogeneous counterparts. For instance, for GCMs with equal probabilities of 1/16, the asymptotic proportions of homogeneous and heterogeneous strings are 1/3 and 1/6 respectively. Thus, homogeneous strings have higher *effective* fitness [27] than heterogenous ones.

In Figure 3 we see a graph of the solution (109) for the strings 11 and 01 for the same asymmetric initial conditions used for Figure 2, and for the same GCMs. Notice the presence of four different fixed points (two-cycle in the case of inversion) for each string type. This is a much richer behaviour than in the case of simple homologous crossover, where the unique Geiringer limit $P^*_{ij} = P_{i*}(0)P_{*j}(0)$ holds. The Geiringer limits for 11 and 10, with the previously stated initial conditions, are $P^*_{11} = 0.3 \times 0.7 = 0.21$ and $P^*_{01} = 0.7 \times 0.7 = 0.49$, both of which agree with the asymptotic limits seen in Figure 3. For one-parent duplication the expected fixed points for 11 and 10 are from (111) $(0.7+0.3)/2$ and 0 respectively, once again in agreement with the graph and showing the higher effective fitness associated with homogeneous strings. Note also the oscillations present in the heterogeneous string 01 and their corresponding absence in the homogeneous string 11.

7 Effects of Selection and $\ell > 2$

We have seen the appearance of novel phenomena and have been able to describe them exactly in a two-locus context in the absence of selection and for an infinite population. One is prompted to wonder whether these phenomena are peculiarities of the two-locus case or are more robust with analogs for $\ell > 2$ and when selection is present. We begin with selection for $\ell = 2$. Of course, it is not feasible to determine what happens for an arbitrary landscape and so restrict ourselves to some generic observations. We first consider in Figure 4 results found using a "Schemulator"[8], a Java-based simulator that integrates the exact coarse-grained equations for any ℓ and any fitness

[8]The programme is available from the authors.

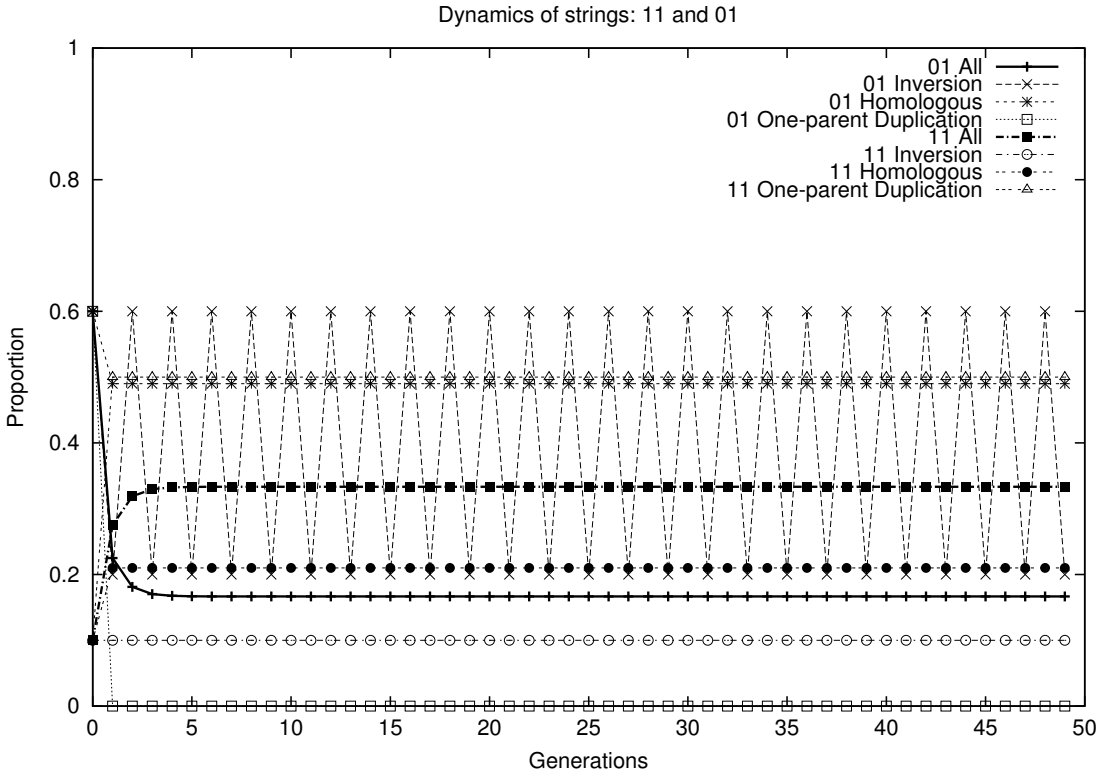

Figure 3: Dynamical evolution of the strings 11 and 01 for different GRDs and no selection

landscape. We first consider the effects for a non-epistatic landscape with fitnesses $f_{11} = 12$, $f_{10} = f_{01} = 11$ and $f_{00} = 10$. What is clearly seen is a similar bias as found from the no selection case of Figure 2, but superimposed on a selection dominated "trend". The oscillations for inversion only are clearly visible. However, here, as only heterogeneous strings can oscillate, and as the optimal string is homogeneous, the oscillations diminish in amplitude. For one-parent duplication, the proportion of 01 strings vanishes after one generation, as in the no selection case. For 01 with all GCMs present, also as in the no selection case, there is a sharp initial decrease. In distinction to that case though, in the presence of selection, the proportion continues to diminish, but at a much reduced rate. As the selection pressure diminishes, the curves of Figure 4 will imitate those of Figure 3 ever more closely, while for increasing selection pressure the phenomena due to inversion and duplication will be less and less noticeable. We see then that, although we have only exactly solved the no selection case, observed phenomena such as lateral allele diffusion, oscillations, preference for homogenous strings etc. are also present in the presence of selection.

Turning our attention now to the case $\ell = 3$, the results are shown in Figure 5, where the fitness landscape here is once again a unitation landscape where $f_{111} = 13$, while the fitness of the Hamming distance one neighbours 110, 101 and 011 is 12, that of the Hamming distance two neighbours 100, 010 and 001 is 11 and $f_{000} = 10$. We restrict ourselves to uniform homologous recombination with the corresponding $p_c(m, (1, 2, 3)) = 0.05$ and with full inversion implemented with $p_c(000, (3, 2, 1)) = p_c(111, (3, 2, 1)) = 0.3$. We also start with the initial condition $P_{110}(0) = 0$, $P_{011}(0) = 0.25$ with all other frequencies being 0.125. Note the characteristic oscillations that inversion can induce superimposed with a selective trend just as in the case of $\ell = 2$. The extra feature we wish to point out here, is the fact that the addition of inversion has added a qualitatively new feature to the search properties of the algorithm. With the chosen initial conditions and only homologous crossover it would be impossible to generate the string 110. This will also be true of Building Blocks as well as strings, as is evident in the observation that the

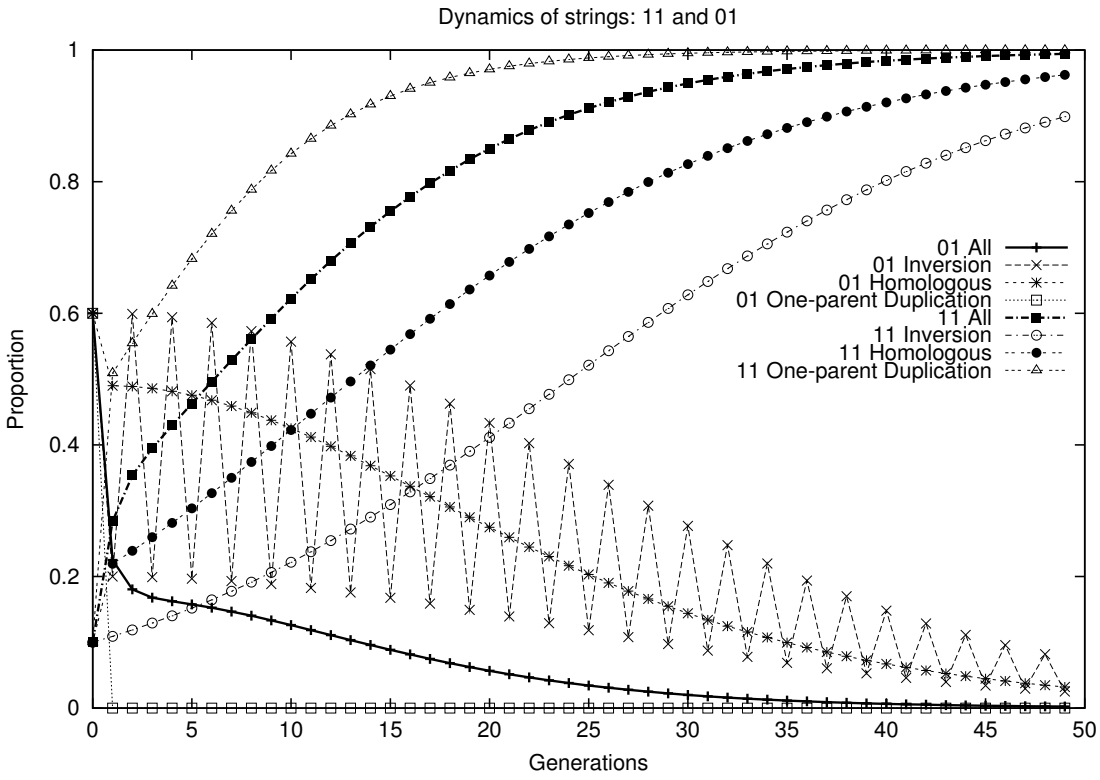

Figure 4: Dynamical evolution of the strings 11 and 01 for different GRDs in a onemax landscape

one-schemata equations are coupled for any ℓ. Thus, there exists the possibility of generating important Building Blocks that are not currently present in the search by lateral allele difusion, induced, for example, by inversion. For example, the Building Block $1*****$ could be generated via a GCM $111111, (6, 5, 4, 3, 2, 1))$ acting on the Building Block $*****1$.

As a final illustration, we consider in Figure 6 the effect of one-parent duplication for $\ell = 3$. The landscape we consider here is associated with a degenerate genotype-phenotype map wherein there are two optimal genotypes - 111 and 010 with fitness 13. The Hamming distance one neighbours of *either* of these strings has fitness 12 and the Hamming distance two neighbours 001 and 100 have fitness 11. The chosen initial population is random. The most notable feature here is the "symmetry breaking" of the genotype-phenotype map due to the presence of duplication, the homogeneous optimum being more effectively fit than its heterogeneous counterpart. In this sense even after optima are found the genetic operators are continuing the evolutionary search seeking those optima that are the most evolutionarily robust - in this case the homogeneous optimum.

So, although for $\ell > 2$ and/or including selection, exact solutions are not available, we can see that the principal derived predictions from the two-locus no-selection case - oscillations, homogeneous/heterogeneous asymmetry, lateral allele diffusion etc. - can all be observed in the more general case. This has been explicitly checked by integrating the equations for both $\ell = 3$ and $\ell = 4$ using the Schemulator. In general, all these phenomena occur simultaneously, for instance, for $\ell = 4$, one might have inversion restricted to the first two loci, leading to oscillations there, while restricting duplication to the last two loci, and having a preference for homogeneous alleles there. Because of the richness of the potential behviours that emerge from generalised recombination, to make the dynamics more transparent we have chosen to illustrate them individually.

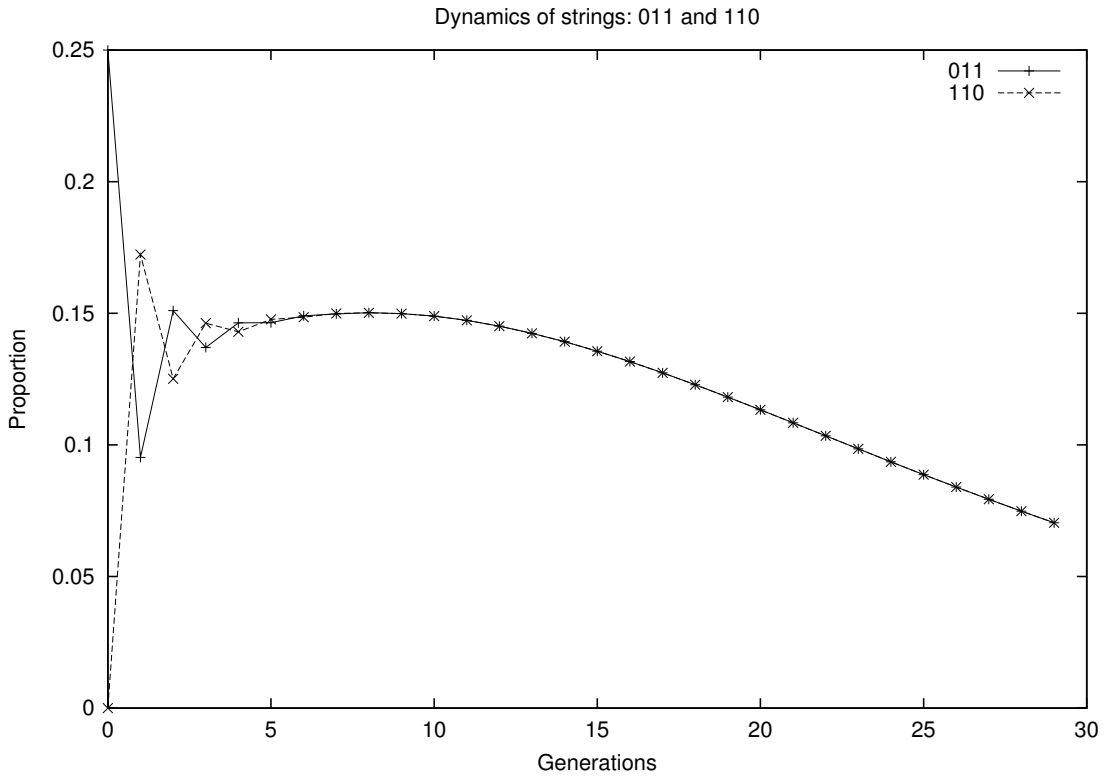

Figure 5: Dynamical evolution of the strings 011 and 110 with inversion and uniform recombination

8 Conclusions

The main results of this paper are two fold: first, the introduction and theoretical analysis of a generalised notion of exchange of genetic material - generalised recombination - which extends and subsumes many currently used genetic operators, such as homologous recombination, inversion and duplication; and secondly, the demonstration that this more general EA is most naturally treated in a coarse grained formulation, wherein the natural dynamical effective degrees of freedom are Building Block schemata not strings.

We showed that generalised recombination requires an extension of the notion of crossover mask and recombination distribution to that of a Generalised Crossover Mask and Generalised Recombination Distribution. GCMs could be explicitly represented in different ways - through a recombination vector, a crossover matrix or a recombination pair. With these representations in hand, an exact string evolution equation was derived, for both variable-length and fixed-length representations, and also including mutation. It was then shown that the dynamics was much more naturally written in terms of Building Block schemata, that emerge by the actions of projection operators that are the natural representation of the generalised recombination operator and which implement course grainings. It was then shown that the resulting string equation, written in terms of Building Block schemata, under a coarse graining, yielded a functionally identical equation for schemata, thus leading to a new Exact Schema theorem for generalised recombination. The coarse graining projection operators were shown to exhibit a semi-group structure, thus giving an explicit realisation of the renormalisation group.

Given that homologous recombination, inversion and duplication have all been found to be useful by practitioners, we do not need to justify the utility of generalised recombination, though it does remain to be seen to what extent the extra diversity, above and beyond the standard operators when considered individually, can help in evolutionary search. Having an exact theoretical

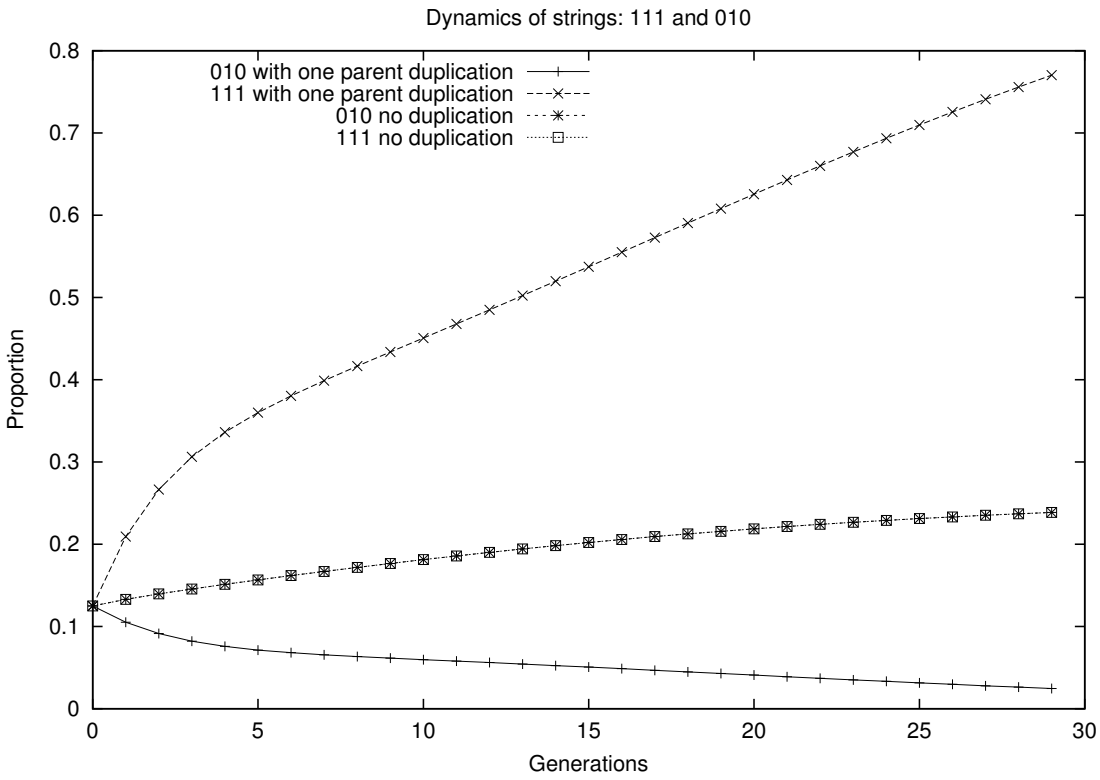

Figure 6: Dynamical evolution of the strings 111 and 010 with uniform recombination and comparing one-parent duplication and no duplication on the problem with degenerate genotype-phenotype map

framework also allows for a better understanding of how the different genetic operators work, as has been exhibited not only in the context of the exact two-locus solution, but also in the results of the Schemulator, where several interesting phenomena were observed. Among these were the appearance of oscillations in the frequencies of strings and schemata in the presence of permutations, higher effective fitness for homogeneous versus heterogenous strings and schemata in the presence of duplication, and the lateral diffusion of alleles for any non-homologous operator, the latter allowing for a more "mutation"-like effect, where an allele that did not originally exist at a particular locus can be generated by transferring it laterally from some other locus. It should be emphasised that, although we have studied these phenomena in the context of an infinite population model, they are in fact robust - appearing also in the finite size context, though, naturally, the intrinsic extra "noise" due to finite population effects can make their identification more difficult.

Acknowledgements

CRS and RP thank the ESPRC for financial support (grant number GR/T24616/01). CRS also thanks DGAPA of the UNAM for a Sabbatical Fellowship and Conacyt project 30422-E.

References

[1] Reinhard Bürger. *The Mathematical Theory of Selection, Recombination, and Mutation*. Wiley, Chichester, UK, 2000.

[2] C. Chryssomalakos Christopher R. Stephens and A. Zamora. Coarse graining in genetic dynamics: A renormalization group analysis of a simple genetic system. *Rev. Mex. Fis*, 2003.

[3] C. Chryssomalakos and C. R. Stephens. What basis for genetic dynamics? In Kalyanmoy Deb, editor, *Proceedings of GECCO 2004*, pages 1018–1029, Berlin, Germany, 2004. Springer Verlag.

[4] A.G. Clark. Invasion and maintenance of a gene duplication. *Proc. Nat. Acad. Sci.*, 91:2950–2954, 1994.

[5] Hilda Geiringer. On the probability theory of linkage in Mendelian heredity. *Annals of Mathematical Statistics*, 15(1):25–57, March 1944.

[6] D. E. Goldberg. Simple genetic algorithms and the minimal deceptive problem. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, pages 74–88, London, UK, 1987. Pitman.

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

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

[9] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.

[10] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, USA, 1975.

[11] John R. Koza. Gene duplication to enable genetic programming to concurrently evolve both the architecture and work-performing steps of a computer program. In *Proceedings of IJCAI-95*, volume 1, pages 734–740, Montreal, Canada, 1995.

[12] Lewin. *Genes VIII*. Prentice Hall, 2003.

[13] Nicholas F. McPhee and Riccardo Poli. A schema theory analysis of the evolution of size in genetic programming with linear representations. In *Genetic Programming, Proceedings of EuroGP 2001*, LNCS, Milan, 18-20 April 2001. Springer-Verlag.

[14] P. Nordin and W. Banzhaf. Complexity compression and evolution. In L. Eshelman, editor, *Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95)*, pages 310–317, Pittsburgh, PA, 1995. Morgan Kaufmann.

[15] P. Nordin, F. Francone, and W. Banzhaf. Explicitly defined introns and destructive crossover in genetic programming. In J. P. Rosca, editor, *Proceedings of the Workshop on Genetic Programming: From Theory to Real World Applications*, pages 6–22, Tahoe City, CA, 1995.

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

[17] Riccardo Poli and Nicholas Freitag McPhee. General schema theory for genetic programming with subtree-swapping crossover: Part I. *Evolutionary Computation*, 11(1):53–66, 2003.

[18] Riccardo Poli, Jon E. Rowe, and Nicholas F. McPhee. Markov chain models for GP and variable-length GAs with homologous crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.

[19] Riccardo Poli and Christopher R. Stephens. Theoretical analysis of generalised recombination. In *CEC-2005*, 2005. Accepted.

[20] Riccardo Poli and Christopher R. Stephens. Theoretical analysis of generalised recombination. Technical Report CSM-426, Department of Computer Science, University of Essex, 2005.

[21] Hidefumi Sawai and Susumu Adachi. A comparative study of gene-duplicated GAs based on pfGA and SSGA. In *Proceedings of GECCO-2000*, pages 74–81. Morgan Kaufmann, 2000.

[22] W. M. Spears. The equilibrium and transient behavior of mutation and recombination. In W. Spears and W. Martin, editors, *FOGA 6*, pages 74–88, San Francisco, USA, 2001. Morgan Kaufmann.

[23] Peter F. Stadler and Christopher R. Stephens. Landscapes and effective fitness. *Comm. Theor. Biol*, submitted to, 2003. Santa Fe Insitute Working Paper: 0210048.

[24] C. R. Stephens. "effective" fitness landscapes for evolutionary systems. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 703–714, Mayflower Hotel, Washington D.C., USA, 6-9 July 1999. IEEE Press.

[25] C. R. Stephens and J. Mora Vargas. Effective fitness as an alternative paradigm for evolutionary computation I: general formalism. *Genetic programming and evolvable machines*, 1(4):363–378, October 2000.

[26] C. R. Stephens and H. Waelbroeck. Effective degrees of freedom in genetic algorithms and the block hypothesis. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, pages 34–40, East Lansing, 1997. Morgan Kaufmann.

[27] C. R. Stephens and H. Waelbroeck. Schemata evolution and building blocks. *Evolutionary Computation*, 7(2):109–124, 1999.

[28] Christopher R. Stephens. The renormalization group and the dynamics of genetic systems. *Acta Phys. Slov.*, 52:515–524, 2003. Preprint: cond-mat/0210217.

[29] Christopher R. Stephens and Riccardo Poli. Coarse graining in an evolutionary algorithm with recombination, duplication and inversion. In *CEC-2005*, 2005. Accepted.

[30] Michael D. Vose. *The simple genetic algorithm: Foundations and theory*. MIT Press, Cambridge, MA, 1999.