Theoretical Analysis of Generalised Recombination

Riccardo Poli
Department of Computer Science
University of Essex
UK
rpoli@essex.ac.uk

Christopher R. Stephens
Department of Computer Science
University of Essex
UK
csteph@essex.ac.uk

Abstract- In this paper we propose, model theoretically and study a general notion of recombination for fixed-length strings where homologous crossover, inversion, gene duplication, gene deletion, diploidy and more are just special cases. The analysis of the model reveals similarities and differences between genetic systems based on these operations. It also reveals that the notion of schema emerges naturally from the model's equations even for the strangest of recombination operations. The study provides a variety of fixed points for the case where recombination is used alone, which generalise Geiringer's manifold.

1 Introduction

An important objective in evolutionary computation (EC) is to exactly model classes of evolutionary algorithms (EAs) and, further, to be able to draw inferences from these models that enhance theoretical understanding and, hopefully, aid "practitioners" in finding more competent EAs. Early models for GAs, proposed by Holland, Goldberg, Whitley and others in the seventies and eighties were either approximate or not easily scalable [4, 3, 28, 29]. Exact probabilistic models have been developed, such as the dynamical systems model of Vose and collaborators [27, 20]. More recently, an alternative exact approach, based on a coarse graining of the dynamics and directly involving schemata, has been introduced, leading to a spate of both new theoretical results [26, 24, 25, 11, 13, 14] and practical recipes for implementation [7, 12].

These models are important in that they allow for the mathematical investigation of the intrinsic dynamics of genetic systems, thereby nicely complementing, corroborating and, occasionally, disproving the findings of empirical studies. However, the vast majority of theoretical work in EAs, at least for classical fixed-length binary and real-valued representations, has been centred on the "canonical" genetic algorithm (GA) with selection, mutation and "homologous" recombination (where a locus in the offspring can by filled only by using alleles coming from the same locus in one of the parents). In nature, though, there are many more ways of combining parental genetic material into an offspring than just homologous crossover, many of which have been used in EAs. Gene duplication, for example, has been studied in biology [1] as well as in the context of GAs [21] and GP [5], while inversion was one of the operators used by Holland [4] in the original formulation of the GA.

In this paper we introduce an exact probabilistic model for fixed length strings, that extends current models by implementing a more general notion of recombination, that can account for *any* distribution of the parental genes to the offspring, including as special cases, among others – fixed-length versions of gene duplication and deletion, as well as inversion and homologous crossover. We show that, as in the case of homologous crossover, a coarse graining naturally appears, revealing that the notion of schemata as building blocks emerges from the model's equations, even for the strangest of recombination operations. The analysis of the model reveals interesting similarities and differences between the various genetic operators present.

2 Generalised Recombination

Crossover masks are normally used to indicate from which parent to take an allele for each available locus. They are sufficient to model a crossover operator when only alleles at the same locus can be exchanged, i.e. homologous crossover. However, if we want to cope with other ways of redistributing genetic material, such as inversion, gene duplication, gene deletion, and, more generally, unequal crossing over, we need to allow for the possibility that the allele in one particular locus of the offspring comes from a different locus of a parent.

This new level of generality can be represented mathematically in several equivalent ways. One is to use arrays (crossover matrices) instead of bit strings to represent crossover events. Crossover matrices are a generalisation of the notion of crossover mask. A crossover matrix will have as many rows as the number of loci in the offspring, say ℓ, and twice as many columns. The first ℓ columns indicate which alleles are copied from the first parent, while columns $\ell + 1$ through to 2ℓ indicate what is provided by the second parent. The elements of the matrix are either 0 or 1. 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 $c \leq \ell$. If $c > \ell$ it is filled with the allele from locus $c - \ell$ 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 this reason we can also represent a recombination matrix as a vector $v = (v_1 \cdots v_\ell)$ with elements from $\mathcal{N}_{2\ell} = \{1, \cdots, 2\ell\}$, where v_i represents the position of the 1 in the i-th row. We will denote either the matrix or vector representation a Generalised Crossover Mask (GCM). The total number of

GCMs is $(2\ell)^\ell$, many more than the 2^ℓ masks for homologous recombination. The action of a GCM, v, is then fully determined when the probability $p_c(v)$ of choosing any particular crossover matrix, or its equivalent crossover vector, is given. This is a generalisation of the notion of recombination distribution – the Generalised Recombination Distribution (GRD).

Another useful representation is a hybrid between the notion of crossover mask and the recombination vector. To represent a possible recombination event we use a *recombination pair* $r \equiv (m, v)$ where $m = (m_1 \cdots m_\ell)$ is an ℓ-component bit vector (i.e., $m \in \{0,1\}^\ell$) and $v = (v_1, \cdots, v_\ell)$ is a vector of integers whose components are in $\{1, \cdots, \ell\}$ (i.e., $v \in \mathcal{N}_\ell^\ell$). The semantics of this representation is very simple. The elements in m specify which parent contributes the alleles to fill each locus in the offspring, while the elements of v tell us which particular alleles in a parent will be transferred to the offspring. So, $m_i = 1$ means locus i will be filled with an allele from parent 1, $m_i = 0$ means parent 2 will contribute the allele instead. If the corresponding entry $v_i = j$ then locus i will be filled with the allele currently in position j in a parent. In this notation, traditional (homologous) crossover events can be represented with pairs of the form $r = (m, (1, 2, \cdots, \ell))$ where, effectively, m can be seen as a traditional crossover mask. Examples of how the different representations of a GCM work are provided in [16].

2.1 Mixing graph and recombination cliques

An important concept when considering redistribution of genetic material as determined by the GRD is: in which direction can one have a flow of genes? As qualitatively different behaviours are exhibited by genetic systems with different GRDs, to understand which features are important, we model the effects of the GRD through a *mixing graph*. The nodes in the graph represent different loci. The arcs are *directed* and represent causal relationships between loci. Thus, we will connect locus i with an arrow from locus j if the frequency of alleles in locus i can be influenced by the allele frequency of locus j.

The network of causal influences is completely determined by the GRD. The connection matrix $C = (c_{ij})$ for the mixing graph is given by

$$c_{ij} = \delta\big(p_c(* \cdots *, (\underbrace{*, \cdots, *}_{i-1}, j, \underbrace{*, \cdots, *}_{\ell-i})) > 0\big)$$

where $\delta(x) = 1$ if x is true, while $\delta(x) = 0$ otherwise. If there is a directed path between each pair of nodes in the mixing graph (the mixing graph is strongly connected), we define the recombination to be *order-1 mixing*.

Imagine a population of strings and focus attention on a particular allele a at a particular locus l of a particular string s. A fully-mixing generalised crossover allows for the migration of allele a to different strings. So, generalised crossover promotes a process of "diffusion" of alleles from one locus to other loci. That is, unlike the case of homologous crossover, in general, generalised crossover does not keep the alleles in their original position, i.e. allele a might migrate to loci different from l. Because of this, in repeated applications of crossover, a copy of the allele can be placed back into the original string s (which may now have a different allele composition) but at a different locus, effectively creating a sort of gene duplication (indeed unequal crossing over seems to be the mechanism of gene duplication in nature [19]). Put another way, crossover is trying to spread each allele as thinly as possible over every locus available in the population. On the other hand, for homologous crossovers, the mixing matrix is diagonal and so each node in the graph is isolated (having only a self-connection).

Naturally, many qualitatively different intermediate situations are also possible. In all intermediate cases we can divide the mixing graph into two or more *recombination cliques*. These are characterised by the fact that all pairs of nodes in a clique are mutually accessible by traversing only nodes and arcs in the clique, while none of the nodes in a clique is mutually accessible from any other node outside the clique. Formally, recombination cliques are the strong components of the recombination graph. So, each locus belongs to one and only one clique. Also, the cliques themselves form a directed acyclic graph (component graph) that we will call the *recombination clique graph*. This has one node for each recombination clique and an arc between two nodes if there is an edge between the corresponding cliques. See [16] for examples.

3 Evolution equations

3.1 Evolution equations for strings

We will now derive and study exact equations for a generational evolutionary system based on selection and generalised recombination and using a fixed-length representation of size ℓ, where alleles take values from a generic alphabet Ω of any fixed cardinality. Under these assumptions the frequency of a string $h = h_1 \cdots h_\ell \in \Omega^\ell$ is given by

$$E[\Phi(h, t+1)] = \sum_{a \in \mathrm{P}(t)} p(a,t) \sum_{b \in \mathrm{P}(t)} p(b,t) \sum_{r \in \mathcal{R}_\ell^\ell} p_c(r) \gamma(a, b, r \to h)$$

where $\Phi(h, t+1)$ is the proportion of strings of type h in the population at generation $t+1$, $\mathrm{P}(t)$ is the population at generation t, $p(a,t)$ is the probability of picking a string of type a as a parent from such a population, and $\mathcal{R}_\ell^\ell = \{0,1\}^\ell \times \mathcal{N}_\ell^\ell$ is the set of all possible crossover pairs. $p_c(r)$ is the GRD and $\gamma(a, b, r \to h)$ is the conditional probability that the offspring h is formed given the parents a and b and a GCM r. It returns value 1 if h is created from a and b using the GCM r and otherwise. Note that we can extend the string summations to cover the entire search space Ω^ℓ rather than just the population $\mathrm{P}(t)$. We are allowed to do so on the assumption that the selection probability $p(x)$ of a string a in Ω^ℓ but not in $\mathrm{P}(t)$ is zero. Note, also, that the model is written in terms of the underlying microscopic degrees of freedom – the strings themselves. Note also that the equation is functionally identical to that for the case of standard mask-based crossover [22], the only difference being the different recombination distribution, and hence the different set of $\gamma(a,b,r \to h)$ that are non-zero. As in the standard crossover case, 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, for binary strings, $2^\ell \times 2^\ell \times (2\ell)^\ell = (8\ell)^\ell$ contributing terms. For example, for two bits there are sixteen GCMs while the sums over the strings a and b run over the values 1 to $|\Omega^\ell|$. Thus, for an arbitrary GRD, even at the two bit level there are $16 \times 4 \times 4 = 256$ $\gamma(a,b,r \to h)$ to compute for a given string h. What is more, for a given h and r, there are potentially many different parental pairs a and b that can yield as offspring h.

In the case of homologous crossover these defects can be circumvented by coarse graining the dynamical equations and passing to a description in terms of Building Block Schemata rather than strings.One is naturally led to enquire as to whether similar benefits may be accrued in this more complex scenario.

The offspring $h = h_1 \cdots h_\ell$, produced by parents $a = a_1 \cdots a_\ell$ and $b = b_1 \cdots b_\ell$, with GRM $r = (m, v)$, can be represented very simply: $h_i = m_i a_{v_i} + (1 - m_i) b_{v_i}$, where a_{v_i} is the allele from the first parent picked out by the crossover pair r, and similarly for b_{v_i} from the second. Then

$$\gamma(a,b,r \to h) = \prod_{i \in I_r} \delta(h_i = a_{v_i}) \prod_{j \in \bar{I}_r} \delta(h_j = b_{v_j})$$

where $I_r = \{i : m_i = 1\}$ represents the genes picked out from the first parent by 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. As the full genetic composition of h has to come from the parents we have $I_r \cup \bar{I}_r = \{1, 2, \cdots, \ell\}$. By substituting this result into the evolution equation for h and reordering terms, we obtain

$$\begin{aligned} &E[\Phi(h, t+1)] \\ &= \sum_{r \in \mathcal{R}_\ell^\ell} p_c(r) \sum_{a \in \Omega^\ell} p(a,t) \prod_{i \in I_r} \delta(h_i = a_{v_i}) \\ &\quad \sum_{b \in \Omega^\ell} p(b,t) \prod_{j \in \bar{I}_r} \delta(h_j = b_{v_j}). \end{aligned}$$

The effect of terms of the form $\prod_{i \in I_r} \delta(h_i = a_{v_i})$ in this equation is simply to limit the summations to subsets of Ω^ℓ. If we denote the elements of I_r with i_k (and the elements of $\bar{I}_r$ with j_k) and we use the standard computer science notation x^y to indicate pattern x repeated y times, these subsets are

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

and the corresponding $\Gamma(h, \bar{I}_r)$. Therefore

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

Thus, we see that the action of the GCM r is to induce a coarse graining on the string sums. The benefit of this is immediately apparent, in that the string sum $\sum_{a \in \Omega^\ell}$ has disappeared. Thus, $p(\Gamma(h, I_r), t)$ denotes the probability for selecting the Building Block schema $\Gamma(h, I_r)$ which forms part of the offspring. In essence, this is identical to the case of homologous crossover. What is more complex here, in the presence of generalised recombination, is the form that the Building Block can take. For example, for $\ell = 4$ and $r = (1101, (4, 3, 4, 1))$, then $I_r = \{1, 2, 4\}$ and, so,

$$\begin{aligned} &\Gamma(h_1 h_2 h_3 h_4, \{1, 2, 4\}) = \\ &\quad = (*^{v_{i_1}-1} h_{i_1} *^{4-v_{i_1}}) \cap (*^{v_{i_2}-1} h_{i_2} *^{4-v_{i_2}}) \cap \\ &\quad\quad (*^{v_{i_3}-1} h_{i_3} *^{4-v_{i_3}}) \\ &\quad = (*^3 h_1) \cap (*^2 h_2 *) \cap (h_4 *^3) = h_4 * h_2 h_1 \end{aligned}$$

hence the first Building Block for the string $h_1 h_2 h_3 h_4$ for the above GRD is $h_4 * h_2 h_1$. The second Building Block is $\Gamma(h, \bar{I}_r) = * * * h_3$. Note that, unlike for the homologous case, in general $h \neq \Gamma(h, I_r) \cap \Gamma(h, I_r) = h_4 * h_2 h_1 \cap * * * h_3$. This new notation based on schemata and the previous calculations lead us to the following

Theorem 1 (Coarse-grained string evolution equation) *The expected frequency of a string h at the next generation in a generational GA with any type of selection with replacement and generalised recombination is given by*

$$E[\Phi(h, t+1)] = \sum_{r \in \mathcal{R}_\ell^\ell} p_c(r) p(\Gamma(h, I_r), t) p(\Gamma(h, \bar{I}_r), t), \qquad (1)$$

where $\Gamma(h, I_r) = \bigcap_{i \in I_r} H_{v_i}^{h_i}$, H_s^a is the order 1 schema $*^{s-1} a *^{\ell - s}$, and $\bar{I}_r = \{1, \cdots, \ell\} \setminus I_r$.

Thus, as in the case of homologous crossover, we see that evolution proceeds by building a string from its component Building Block schemata. Of course, to make further progress, one would then need to have the equations that govern these schemata. We will do this in the next section. Before we do that, however, we would like to discuss the differences between strings and schemata for describing the evolution. Firstly, note that there are an exponentially large number of ways of reshuffling genetic material from parents to offspring. To emphasise once again, there are $(2\ell)^\ell$ GRMs irrespective of whether strings or schemata are used to describe the dynamics. Of course, it may well be that only a small subset of these masks have non-zero probability. For instance, for homologous crossover there are 2^ℓ possible masks. However, for one-point crossover only $\ell - 1$ of these masks have non-zero probability. For a given GRM, there remains the question of how many combinations of strings or schemata can lead to a particular offspring. This is where the advantage of Building Block schemata plays a crucial role as for a given GRM there is uniquely only *one* relevant pair of schemata and, therefore, correspondingly only one term in the r.h.s. of Equation 1. For strings, however, there are an exponential number of terms to consider. Also, even if there are an exponential number of GRMs, as we will show later, we can study schema equations formally (i.e., for any ℓ and without having to compute the actual terms in the equations) to infer general properties of genetic systems.

3.2 Coarse-grained evolution equations

For homologous crossover, one of the most remarkable features of the coarse grained exact schema equations is their form invariance under a further coarse graining [26], i.e. that the functional form of the equations for a Building Block 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) building blocks, which in their turn etc., the whole hierarchy terminating at the 1-schemata. 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.

This is indeed the case as shown by the following

Theorem 2 (Schema evolution equation) *Equation 1 is applicable to both strings and schemata of any order.* (The proof is available in [16].)

3.3 A more explicit notation

When, for a given recombination pair $r = (m, v) \in \mathcal{R}_\ell^\ell$, v is a permutation of the vector $(1, 2, \cdots, \ell)$, then $\Gamma(h, I_r) = \bigcap_{k=1}^{|I_v|} H_{v_{i_k}}^{h_{i_k}}$ is an ordinary schema. In order to be able to express exactly which schema this is we need to order the sets $I_r = \{i_1, i_2, \cdots, i_{|I_r|}\}$ and $\bar{I}_r = \{j_1, j_2, \cdots, j_{|\bar{I}_r|}\}$ on the basis of the corresponding entries in the vector v. That is, the elements i_k of I_r are ordered in such a way that $v_{i_k} \leq v_{i_{k+1}}$ for any k, and the same is true for $\bar{I}_r$.

For example, if $\ell = 4$ and $r = (m, v) = (1101, (4, 3, 4, 1))$, then I_r is obtained as follows. As before, first we collect the indices of the elements of m that are 1 in a set (in this example, $\{1, 2, 4\}$). Then we sort the elements of this set based on the values of the corresponding elements in v. So, because $v_4 \leq v_2 \leq v_1$, $I_r = \{4, 2, 1\}$. Naturally, $\bar{I}_r = \{3\}$.

With this ordering, when v is a permutation, then $v_{i_k} < v_{i_{k+1}}$ for all k. Therefore

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

where we used the convention that $v_{i_0} = 0$, that the $\prod$ operator means *concatenation* when applied to strings of symbols and that $*^0$ is the empty symbol (i.e. $*^0$ can be safely edited out from any sequence of characters).

We can interpret $\Gamma(h, I_r)$ as a schema also when $v_{i_k} = v_{i_{k-1}}$ for some k, as long as $h_{i_k} = h_{i_{k-1}}$. If this is not the case, then $\Gamma(h, I_r)$ is the empty set $\emptyset$ (naturally $p(\emptyset, t) = 0$).

Therefore, in general we can write

$$\begin{aligned} & p\left(\Gamma(h, I_r), t\right) \\ & \quad = \quad p\Big(\prod_{\substack{1 \leq k \leq |I_r| \\ i_k \neq i_{k-1}}} \left(*^{v_{i_k} - v_{i_{k-1}} - 1} h_{i_k} \right) *^{\ell - v_{i_{|I_r|}}}, t \Big) \\ & \qquad \times \prod_{\substack{1 \leq k \leq |I_r| \\ i_k = i_{k-1}}} \delta(h_{i_k} = h_{i_{k-1}}) \end{aligned} \tag{2}$$

3.4 Examples

As an example, let us write the evolution equations for a generic string of length $\ell = 2$ from Equation 1 with the more explicit "δ notation" introduced in Section 3.3:

$$\begin{aligned} & E[\Phi(ab, t+1)] \\ & \quad = \quad p_{11} p(a*) \delta(a = b) + p_{12} p(ab) + p_{13} p(a*) p(b*) \\ & \quad + \quad p_{14} p(a*) p(*b) + p_{21} p(ba) + p_{22} p(*a) \delta(a = b) \\ & \quad + \quad p_{23} p(*a) p(b*) + p_{24} p(*a) p(*b) + p_{31} p(b*) p(a*) \\ & \quad + \quad p_{32} p(*b) p(a*) + p_{33} p(a*) \delta(a = b) + p_{34} p(ab) \\ & \quad + \quad p_{41} p(b*) p(*a) + p_{42} p(*b) p(*a) + p_{43} p(ba) \\ & \quad + \quad p_{44} p(*a) \delta(a = b) \end{aligned}$$

where for simplicity we omitted time from the selection probabilities and we used p_{ij} as a shorthand notation for the GRD $p_c(v)$, $v = (i, j) \in \mathcal{N}_4^2$ being a recombination vector (see Section 2).

If one replaces a and b with some values from Ω, all of the δ's turn either into 1's or 0's, and so it is possible to further simplify the equation. For example, if $a = b = 1$ and all GCMs have equal probability ($p_{ij} = 1/16$), we obtain

$$\begin{aligned} & E[\Phi(11, t+1)] \\ & \quad = \quad 0.125 p(1*)^2 + 0.125 p(1*) + 0.25 p(1*) p(*1) \\ & \quad + \quad 0.125 p(*1)^2 + 0.25 p(11) + 0.125 p(*1) \end{aligned}$$

Notice that in order to solve for the dynamics of the strings we need to have a solution for the building blocks a*, *a, b* and *b.

As an example, the evolution equation for the schema a* (a building block for ab) is

$$\begin{aligned} E[\Phi(\mathtt{a}*, t+1)] & = (p_{1*} + p_{3*}) p(\mathtt{a}*) \\ & + (p_{2*} + p_{4*}) p(*\mathtt{a}) \end{aligned}$$

where $p_{x*} = \sum_y p_{xy}$.

A much deeper analysis of the $\ell = 2$ case is provided in [23], where a complete, exact solution, is derived, showing how the dynamical behaviour is radically different to that of homologous crossover. Even in such a simple case new qualitatively different behaviour is observed. For example, inversion is shown to potentially introduce oscillations in the dynamics, while gene duplication leads to an asymmetry between homogeneous and heterogeneous strings. Also, all non-homologous operators lead to allele "diffusion" along the chromosome. The case $\ell = 3$ is exemplified in [16].

3.5 General case

These examples show that all schema/string evolution equations have the same structure with a linear part which depends on the selection probabilities of schemata of the same order as the schema on the left-hand side of the equation, and a non-linear forcing term which depends on lower-order schemata. The only exception to this is order one objects, in which case there is no forcing term. These objects, therefore, evolve independently but contribute to all higher-order schemata. So, *order one schemata act as pacemakers for a genetic system evolving under generalised recombination.* For these reasons we will analyse the evolution equations for such a case in more detail in the next section.

4 Equations for order 1 schemata

Let us focus on the order 1 schemata $H_s^a = *^{s-1}a*^{\ell-s}$ where only one allele is specified. By coarse-graining on the recombination distribution, the schema evolution equations for these schemata transform into:

$$\begin{aligned} &E[\Phi(H_s^a, t+1)] \\ &\quad= \sum_{(m_s,v_s)\in\mathcal{R}_\ell} p_c(*^{s-1}m_s*^{\ell-s}, *^{s-1}v_s*^{\ell-s})p\left(H_{v_s}^a, t\right) \\ &\quad= \sum_{k=1}^{\ell} p_c(*\cdots*, *^{s-1}k*^{\ell-s})p\left(H_k^a, t\right). \end{aligned}$$

That is, the evolution of order 1 schemata is governed by systems of ℓ linear equations. There are as many such systems as the arity of the alphabet adopted for strings. In the binary case $a \in \{0, 1\}$ and so there are two such systems.

So, in general, unlike the case for homologous crossovers, with generalised recombination, order 1 schemata may evolve even on a flat landscape (where $p(H, t) = \Phi(H, t)$ for any schema H). The flat landscape case is interesting as its analysis unveils the biases of genetic operators [8, 17, 18, 6]. These biases become very important whenever selection is not dominating, as, for example, towards the end of a run or when the algorithm is exploring an area rich in neutral networks.

Let us consider the case of an infinitely large population and a flat landscape. Infinitely large populations are a standard mathematical tool in the theory of evolutionary algorithms. They are used because they remove the stochasticity present in EAs. This can be very useful, for example, to aid the analysis of the intrinsic biases of the search operators. It is, however, possible to use exact schema evolution equations to study for finite population evolution. Indeed, it is easy to construct a Vose-like Markov chain model for generalised recombination by using Equation 1 to provide the success probabilities for the multinomial distribution which gives the entries of the transition matrix of the chain (see [15] for an example). Seen as a stochastic process, a GA has an enormous number of possible states. For the case of binary strings of length ℓ, a GA with a population of M individuals can be in any of $N = (M+2^\ell-1)!/M!(2^\ell-1)!$ different states [9]. So, a Markov chain for a GA requires an immense ($N \times N$) transition matrix, implying computations that are much worse than exponential.

With an infinite population, in vector notation, the system of equations becomes

$$\vec{\Phi}^a(t+1) = A\vec{\Phi}^a(t) \qquad (3)$$

where $\vec{\Phi}^a(t) = [\Phi(H_1^a, t), \cdots, \Phi(H_\ell^a, t)]^T$ and $A = (a_{sk})$ is a matrix with elements $a_{sk} = p_c(*\cdots*, *^{s-1}k*^{\ell-s})$. Since $\sum_{k=1}^{\ell} p_c(*\cdots*, *^{s-1}k*^{\ell-s}) = p_c(*\cdots*, *\cdots*) = 1$ the matrix A is row stochastic, but it is not necessarily column stochastic.

For the case $\ell = 2$ in [23] we found that, except in special conditions, a fixed point for the proportions of order 1 schemata $\Phi(H_s^a, t)$ exists. This is generally the case for any ℓ. Let us denote such a fixed point with $\Phi^*(H_s^a)$.

4.1 Fixed points

Let us look for fixed points for the dynamical system defined by Equation 3. They will have to be eigenvectors of the matrix A with an associated eigenvalue $\lambda = 1$.

Because of the row stochasticity of A, it is easy to see that $[1, \cdots, 1]^T$ is an eigenvector for the matrix. That is, for order 1 schemata, a fixed point always exists of the form

$$\Phi^*(H_s^a) = c(a)$$

for $s = 1, \cdots, \ell$, where $c(a)$ is a constant (possibly a different one for each a). Naturally the constants $c(a)$ must obey the conservation of probability for the ℓ sets of order 1 schemata partitioning the search space. That is, we require that, for all s and t,

$$\sum_a \Phi(H_s^a, t) = 1.$$

When evaluated at the fixed point, this leads to the following constraint on the values of the $c(a)$'s:

$$\sum_a c(a) = 1.$$

Generally, finding analytically other fixed points may not be simple. Also, determining whether a fixed point is a global attractor for the system is non-trivial. There are, however, some fairly general classes of generalised recombinations where we can say a bit more.

4.1.1 Fully disconnected recombination cliques

Let $Q(p_c)$ the set of recombination cliques induced by the generalised recombination distribution p_c. The elements of $Q(p_c)$ are (disjoint) sets of integers. Their union is $\{1, \cdots, \ell\}$.

Homologous crossover is a special case in which the recombination clique graph includes ℓ disconnected nodes (i.e., $|Q(p_c)| = \ell$). The order-1 mixing case is one where all nodes belong to a single clique (i.e., $|Q(p_c)| = 1$). Let us consider what happens in other cases where the loci can be grouped into a number of cliques, but where the cliques themselves are completely disconnected. In other words, we consider the case where the recombination clique DAG includes $q = |Q(p_c)|$ nodes with $1 < q < \ell$ and *no arcs*.

In this case the matrix A is block diagonal, with q blocks. So, effectively we can decompose the vector $\vec{\Phi}^a$ into q subvectors $\vec{\Phi}^a_n$ and the matrix A into q squared sub-matrices A_n (the blocks along the diagonal of A) and rewrite the evolution equations for order 1 schemata as:

$$\vec{\Phi}^a_n(t+1) = A_n \vec{\Phi}^a_n(t)$$

for $n \in Q(p_c)$. It is then easy to see that each of these smaller dynamical systems has an eigenvalue $\lambda_n = 1$ with an associated eigenvector of the form $[1, \cdots, 1]^T$. So, a fixed point exists of the form

$$\vec{\Phi}^{a^*}_n = c(n,a)[1, \cdots, 1]^T$$

for $n \in Q(p_c)$, where $c(n,a)$ are constants which depend only on the clique n and the allele a. These, again, must respect the conservation of probability and so

$$\sum_a c(n,a) = 1.$$

5 Fixed points for higher-order schemata and strings

Let us consider the case where $p_c(m,v) = 0$ for all v such that $\exists i \neq j, v_i = v_j$, that is let us assume no allele duplication can take place.

Theorem 3 (Generalised Geiringer manifold) *A fixed point distribution for the proportion of a string or a schema $h_1 h_2 \cdots h_\ell$ under generalised crossover with a duplication-free recombination distribution for an infinite population operating on a flat fitness landscape is given by*

$$\Phi^*(h_1 \cdots h_\ell) = \prod_{q \in Q(p_c)} \prod_{i \in q} c(q, h_i) \qquad (4)$$

where $c(q,*) = 1$. (The proof is available in [16].)

This result is important because *it provides a generalisation of the manifold described, for homologous crossover, by Geiringer* [2]. All points on our generalised Geiringer manifold are fixed points for a genetic system under generalised recombination. Naturally, the result also covers all the fixed points for order one schemata described in the previous section.

It is interesting to rewrite Equation 4 is a slightly different form. If $\nu(h,n,a)$ represents the number of times symbol a appears in one of the loci in clique n of the string or schema h, and Ω represents our alphabet, then

$$\Phi^*(h) = \prod_{n \in Q(p_c)} \prod_{a \in \Omega} (c(n,a))^{\nu(h,n,a)}. \qquad (5)$$

So, for example if our alphabet is $\Omega = \{0,1,2,3\}$, if $|Q(p_c)| = 1$ and if we set $c(n,0) = c(n,1) = 1/3$ and $c(n,2) = c(n,3) = 1/6$, then $\Phi^*(0102) = (1/2)^2 \times (1/2) \times (1/3) \times (1/3)^0 = 1/24$. Interestingly, in the case of a binary alphabet, for a fixed $c(n,0)$ (note: $c(n,1) = 1 - c(n,0)$) the probability of sampling a given string is only a function of the unitation value (the number of ones) of the string.

6 Stability of fixed points

Naturally, although any choice of $c(n,a)$ will provide a formal fixed point for the evolution equations, we are only interested in choices which respect the conservation of probability constraint $\sum_a c(n,a) = 1$. Despite this constraint, we still have a huge family of potential fixed points. An important question is whether any of these fixed points would be a global attractor for the system and whether this would dependent on initial conditions and, if so, how.

In this paper we don't formally prove under which conditions the fixed point presented in the previous sections are stable. In [23] we present an exact and general solution for the dynamics for the case $\ell = 2$ and a complete analysis of the corresponding fixed points. The techniques used there can provide exact answers also for $\ell > 2$. However, the complexity of the solutions grows very quickly with ℓ. So, in this paper we prefer to present empirical evidence to corroborate our theoretical results.

7 "Schemulator" runs

In order to study the dynamics of a genetic system under selection and generalised recombination we have implemented a simulator written in Java (we call it the "*schemulator*" – a contraction of "schema simulator") which expands and then numerically integrates the string (and schema) evolution equations for any choice of recombination distribution, of fitness function and of initial conditions. The integration is performed under the standard assumption of infinite populations.

To corroborate our results we want to verify our predictions as to the existence and location of fixed points for the flat fitness landscape case. Figure 1 shows the dynamics of some schemata and strings in a population with $\ell = 3$ and a recombination distribution where $p_c(m,v) \neq 0$ for all the 48 recombination pairs where v is a permutation vector, and $p_c(m,v) = 0$ for the remaining 168 pairs. The non-zero entries of the GRD were randomly generated and then normalised so that $\sum p_c(r) = 1$. The resulting recombination distribution had only one clique, $\mathcal{N}_\ell = \{1, \cdots, \ell\}$, which includes all ℓ loci. In order to be able to distinguish between the dynamics of different schemata, we the used unequal initial proportions for strings, namely: $\Phi(\mathtt{000},0) = 0.3$, $\Phi(\mathtt{001},0) = 0.25$, $\Phi(\mathtt{010},0) = \Phi(\mathtt{011},0) = \Phi(\mathtt{100},0) = 0.1$, $\Phi(\mathtt{101},0) = 0.05$, $\Phi(\mathtt{110},0) = 0.02$ and $\Phi(\mathtt{111},0) = 0.08$.

As shown in the figure, the order 1 schemata H^1_s ($s = 1,2,3$) rapidly converge to a fixed point where $\Phi^*(1**) = \Phi^*(*1*) = \Phi^*(**1)$. This is exactly what is predicted by the fixed point provided in Equation 4. The order-one-schema fixed point proportion, 0.343333333333, suggests that $c(\mathcal{N}_\ell, 1) = 0.343333333333$ and $c(\mathcal{N}_\ell, 0) = 1 - c(\mathcal{N}_\ell, 1) = 0.656666666667$.

Order 2 schemata also converge to identical values, i.e. $\Phi^*(11*) = \Phi^*(*11) = \Phi^*(1*1)$. The fixed-point frequency is (within numerical errors) exactly $c(\mathcal{N}_\ell, 1)^2 = 0.117877777778$, which is what Equation 4 predicts.

The predictions of our generalised Geiringer mani-

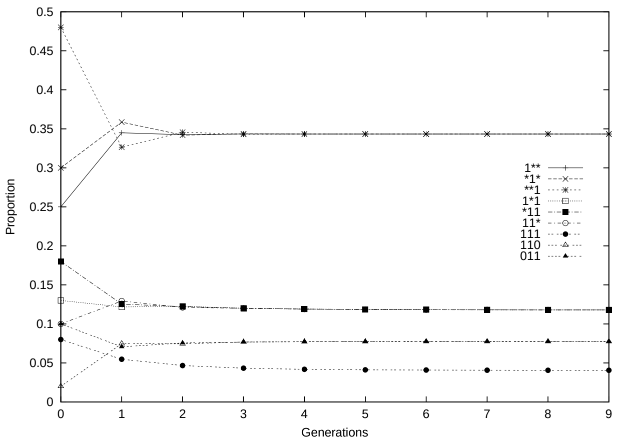

Figure 1: Dynamics of strings and schemata for $\ell = 3$ and a duplication-free, order-1 mixing, random recombination distribution.

fold theorem also hold for strings. For example, the strings 110 and 011 converge to their predicted fixed point $\Phi^*(110) = \Phi^*(011) = c(\mathcal{N}_\ell, 1)^2 c(\mathcal{N}_\ell, 0) = 0.0774064074076$ and 111 converges towards the predicted $\Phi^*(111) = c(\mathcal{N}_\ell, 1)^3 = 0.0404713703703$ within numerical errors.

8 Discussion and Conclusions

In this paper we have provided a theory that is powerful enough to model exactly genetic systems using a fixed-length representation, selection and, for the first time, a rich set of genetic operations, including gene duplication, gene deletion, inversion, homologous recombination, permutations, diploidy, etc. that are not only known to happen in nature but that have also been fruitfully used in evolutionary algorithms. This model includes as a special case previous models such as the exact schema theory in [26, 22].

We have started analysing the evolution equations provided by our model with the objective of understanding the search biases induced by such a powerful set of operators. This has allowed us to formulate a generalisation of Geiringer's manifold. As usual, we expect the study of the equations in the presence of selection to be much harder to do mathematically. However, the availability of an exact probabilistic model has allowed the implementation of an evolution equation simulator (the schemulator) with which we can numerically explore the interaction between the recombination and the selection biases for arbitrary fitness functions and potentially for any string length under the assumption of infinite populations. As we mentioned before this assumption is a standard mathematical tool to understand the dynamics of a genetic system. In practice, these simulations give very accurate results whenever sampling errors and drift are marginal phenomena. This typically is the case if one has a sufficiently large population. However, the results may be reliable also for smaller populations as long as one does not integrate the equations for too many time steps (e.g., in the case of short runs).

In future research we intend to provide a detailed general analysis of fixed-point stability, to study the evolution equations for diploid recombination distributions and to extend the results presented in this paper to the case of variable length strings, thereby, hopefully, contributing new results to theoretical population genetics as well as evolutionary computation.

From a practitioner's point of view, where could one expect to find that generalised recombination operators perform better than standard recombination operators? We have already some answers. Let us consider, for example, the effects of the lateral diffusion process typically present in generalised recombination. With this process, every time the population reaches an area of flat fitness, lateral diffusion in combination with homologous mixing will start destroying the correlations induced by selection and will effectively re-randomise the population (using unequal allele frequencies) in the neighbourhood of the best solutions found so far. This can have a very beneficial impact both in realising open ended evolutionary systems and in exploring, in an unbiased way, neutral networks. As another example, let us consider the effects of duplication. In many systems the function of an allele is not fully (in some not even partly) determined by its locus. This is the case, for example, in nature, but also in practical EAs such as certain types of linear genetic programming systems (which evolve instructions for a register based CPU in fixed length chromosomes). In these systems gene duplication may be an excellent mechanism to promote reuse of useful instructions. Naturally, generalised recombination is expected to be beneficial also in problems where solutions are expected to present a high degree of genotypic self-similarity (a trivial example is the one-max problem, which, as we empirically verified, is solved more quickly when using generalised recombination than with homologous crossover). Finally, we should note that the availability of exact schema equations for an operator (such as those for generalised recombination provided in this paper) allows one to study the interactions of multiple operators and to determine their optimal parameter settings (see [7] for an example). It is also possible to relate such equations to the sizing of populations [10].

Acknowledgements

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

Bibliography

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

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

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

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

[5] J. 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, 20-25 Aug. 1995. Morgan Kaufmann.

[6] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer-Verlag, 2002.

[7] N. F. McPhee and R. Poli. Using schema theory to explore interactions of multiple operators. In *Proceedings of GECCO 2002*, pages 853–860, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[8] N. F. McPhee, R. Poli, and J. E. Rowe. A schema theory analysis of mutation size biases in genetic programming with linear representations. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 1078–1085, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.

[9] A. E. Nix and M. D. Vose. Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88, 1992.

[10] R. Poli. Recursive conditional schema theorem, convergence and population sizing in genetic algorithms. In *Proceedings of FOGA 6*, pages 143–163, Charlottesville, VA, USA, 21–23 July 2000.

[11] R. 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, June 2001.

[12] R. Poli. A simple but theoretically-motivated method to control bloat in genetic programming. In *Proceedings of EuroGP 2003*, LNCS, pages 211–223, Essex, UK, 14-16 Apr. 2003. Springer-Verlag.

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

[14] R. Poli and N. F. McPhee. General schema theory for genetic programming with subtree-swapping crossover: Part II. *Evolutionary Computation*, 11(2), 2003.

[15] R. Poli, N. F. McPhee, and J. E. Rowe. Exact schema theory and markov chain models for genetic programming and variable-length genetic algorithms with homologous crossover. *Genetic Programming and Evolvable Machines*, 5(1):31–70, Mar. 2004.

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

[17] R. Poli, C. R. Stephens, A. H. Wright, and J. E. Rowe. On the search biases of homologuous crossover in linear genetic programming and variable-length genetic algorithms. In *Proceedings of GECCO 2002*, pages 868–876, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[18] R. Poli, C. R. Stephens, A. H. Wright, and J. E. Rowe. A schema-theory-based extension of Geiringer's theorem for linear GP and variable-length GAs under homologous crossover. In *Proceedings of FOGA-VII*, pages 45–62, Torremolinos, (4–6 September, 2002), 2003. Morgan Kaufmann.

[19] M. Ridley. *Evolution*. Blackwell Scientific Publications, Boston, 1993.

[20] J. E. Rowe, M. D. Vose, and A. H. Wright. Group properties of crossover and mutation. *Evolutionary Computation*, 10(2):151–184, 2002.

[21] H. Sawai and S. Adachi. A comparative study of gene-duplicated GAs based on pfGA and SSGA. In *Proceedings of GECCO-2000*, pages 74–81, Las Vegas, Nevada, USA, 10-12 July 2000. Morgan Kaufmann.

[22] C. R. Stephens. Some exact results from a coarse grained formulation of genetic dynamics. In *Proceedings of GECCO-2001*, pages 631–638, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.

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

[24] C. R. Stephens and J. M. Vargas. Effective fitness as an alternative paradigm for evolutionary computation I: General formalism. *Genetic Programming and Evolvable Machines*, 1(4):363–378, Oct. 2000.

[25] C. R. Stephens and J. M. Vargas. Effective fitness as an alternative paradigm for evolutionary computation II: Examples and applications. *Genetic Programming and Evolvable Machines*, 2(1):7–32, Mar. 2001.

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

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

[28] D. Whitley. An executable model of a simple genetic algorithm. In *Proceedings of FOGA-92*, Vail, Colorado, July 1992.

[29] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994.