Understanding the Biases of Generalised Recombination: Part I

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

Christopher R. Stephens csteph@essex.ac.uk
Department of Computer Science, University of Essex, UK, and
Instituto de Ciencias Nucleares, UNAM, A. Postal 70-543, México, D.F. 04510

Abstract

This is the first part of a two-part paper where we propose, model theoretically and study a general notion of recombination for fixed-length strings, where homologous recombination, inversion, gene duplication, gene deletion, diploidy and more are just special cases. The analysis of the model reveals that the notion of schema emerges naturally from the model's equations. In Part I, after describing and characterising the notion of generalised recombination, we derive both microscopic and coarse-grained evolution equations for strings and schemata and illustrate their features with simple examples. Also, we explain the hierarchical nature of the schema evolution equations and show how the theory presented here generalises past work in evolutionary computation. In Part II, the study provides a variety of fixed points for evolution in the case where recombination is used alone, which generalise Geiringer's theorem. In addition, we numerically integrate the infinite-population schema equations for some interesting problems, where selection and recombination are used together to illustrate how these operators interact. Finally, to assess by how much genetic drift can make a system deviate from the infinite-population-model predictions we discuss the results of real GA runs for the same model problems with generalised recombination, selection and finite populations of different sizes.

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 genetic algorithms (GAs), proposed by Holland, Goldberg, Whitley and others in the seventies and eighties were either approximate or not easily scalable (Holland, 1975; Goldberg, 1989; Whitley, 1992; Whitley, 1994). Exact probabilistic models have been developed, such as the dynamical systems model of Vose and collaborators (Vose, 1999; Rowe et al., 2002). More recently, an alternative exact approach, based on the notion of coarse graining of GA dynamics, has been proposed (Stephens and Waelbroeck, 1999). In physics, coarse graining methods are used to obtain more concise and easily understandable, but sometimes approximate, models of complex systems involving large numbers of microscopic degrees of freedom. The variables used to describe a system in a coarse grained model are often called *effective degrees of freedom*. Frequently, these represent macroscopic quantities which summarise the original microscopic degrees of freedom. Coarse grained models are often naturally suggested by the system under investigation itself. When applying coarse graining to EAs with crossover, schemata emerge naturally as the appropriate effective degrees of freedom. This is why the notion of schema has recently led to a spate of both new exact theoretical results (Stephens and Waelbroeck, 1999; Stephens et al., 1999; Stephens and Vargas, 2000; Poli, 2001; Poli and McPhee, 2003a; Poli and McPhee, 2003b) and practical recipes for implementation (McPhee and Poli, 2002; Poli, 2003). Also, schema models have revealed that there are tight links between GAs and Genetic Programming (GP) (Langdon and Poli, 2002).[1]

Exact models for EAs are vital, in that they offer the fundamental mathematical foundation upon which approximate models should derive their validity. Based on this exact foundation, theory should possess the twin properties of providing intuition and understanding at the qualitative level; while allowing for predictions for measurable quantities at the quantitative level. Comparison with empirical studies should be feasible at both these levels. Up to now, the vast majority of theoretical work in EAs for classical fixed-length binary and real-valued representations has been centred on the "canonical" GA, or on Evolutionary Strategies, with selection, mutation and "homologous" recombination. In the latter, a position (locus) in the offspring can *only* be filled using gene values (alleles) from *the same position* 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, some of which have been used in EAs. Gene duplication, for example, has been studied in biology (Clark, 1994), as well as in the context of GAs (Sawai and Adachi, 2000) and GP (Koza, 1995), while inversion, albeit of a more sophisticated kind than that considered here, was one of the operators used by Holland (1975) in the original formulation of the GA.

In Part I of this paper, we begin with an exact, microscopic probabilistic model for the evolution of a population of fixed length, ℓ, strings undergoing selection and generalised recombination, the latter accounting for *any* redistribution 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. With this more general form of recombination, there are potentially many more ways of reshuffling genetic material between parents and offspring when compared to homologous crossover ($(2\ell)^\ell$ versus 2^ℓ). Interestingly, however, 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, irrespective of how genetic material is redistributed. In addition, we show that, as in the case of homologous crossover, the schema evolution equations are hierarchical, in the sense that objects at a higher level of the hierarchy depend on the evolution of objects at a lower level.

In Part II, the study provides a variety of fixed points for evolution in the case where recombination is used alone, thus generalising Geiringer's theorem. In addition, we numerically integrate the infinite-population schema equations for some interesting problems, where selection and recombination are used together to illustrate how these operators interact. Finally, to assess by how much genetic drift can make a system deviate from the infinite-population-model predictions we discuss the results of real GA runs for the same model problems with generalised recombination, selection and finite populations of different sizes.

Part I is organised as follows. In Section 2 we introduce some basic definitions

[1] Microscopic and macroscopic (coarse grained) approaches are not in competition. Indeed, it is possible to prove that one can construct Vose-like Markov chain models for EAs by using exact schema equations and, vice versa, one can track schema frequencies by coarse graining the dynamical systems model's equations (Poli et al., 2001; 2004). So, EC theory is now effectively unified to a considerable degree (Stephens and Poli, 2004).

relating to the notion of schema, and summarise and further characterise the notion of generalised recombination that we previously introduced in (Poli and Stephens, 2005a; 2005b; Stephens and Poli, 2005a). In Section 3 we write down the microscopic evolution equations for strings and schemata that form the starting point for our study, using the standard interpretation of schemata as sets (an extensive characterisation of these equations in terms of projection operations is given in (Stephens and Poli, 2005a)) and illustrating the features of these equations with simple examples. We explain the hierarchical nature of the schema evolution equations in Section 4 and show how the theory presented here generalises past work in Section 5. Finally, we provide some conclusions and a summary of Part II in Section 6.

2 Background

In this section we present some background material that is necessary to understand the contributions of this paper. We start with some definitions related to the notion of schema, and then provide a description of generalised recombination and some of its properties.

2.1 Schemata

In this paper we restrict our attention to a search space of fixed-length strings of length ℓ, where string elements (or alleles) take values from a generic alphabet Ω of any fixed cardinality. If by $\times$ we indicate the Cartesian product operator, the search space is $\overbrace{\Omega \times \Omega \times \cdots \times \Omega}^{\ell \text{ times}}$. Following standard convention we will denote this by Ω^ℓ.

A *schema* is a subset of the search space. Syntactically we will represent schemata as strings in the standard notation based on the "don't care" symbol $*$. So, for example, the schema $h = {*}a{*}bc$ represents all strings in the search space whose second, fourth and fifth characters are a, b and c, respectively. The *order* of a schema is the number of non-$*$ (or defining) symbols in it. So, the order of h is 3. The *defining length* of a schema is the distance between the furthest defining symbols. The defining length of h is 3. The schema ${*}{*}{*} \cdots {*}$ represents Ω^ℓ.

In the following we will need to be able to express and perform operations on schemata both as sets and as strings with don't care symbols. So, here we introduce some notation that will facilitate this. In order to represent schemata in which one or more symbols are repeated a certain number of times, we use the standard computer science notation x^y to indicate pattern x repeated y times. So, for example, we may represent the schema ${*}{*} \cdots {*}$ ($*$ repeated ℓ times) as $*^\ell$, and we may represent the schema ${*}{*}{*}11111{*}$ as $*^3 1^5 *$. In some cases we will need to specify schemata where a pattern of alleles and don't care symbols is repeated a certain number of times. If the pattern is constant, we can do this by using brackets and the power notation just introduced. For example, $*^2(10)^5*$ represents the schema ${*}{*}1010101010{*}$. However, we will encounter cases where the pattern varies as it is repeated. In these cases we will make use of the *concatenation operator* $\bigotimes_i$. This operator is for concatenation what $\sum_i$ is for sums and $\prod_i$ for products. For example, the schema ${*}1{*}{*}1{*}{*}{*}1{*}{*}{*}{*}1$ could be represented as $\bigotimes_{i=1}^4 (*^i 1)$. In addition, we will use the convention that x^0 is the *empty sequence* whatever the sequence x (i.e. x^0 can be safely edited out from any sequence of characters).

Let us now consider some useful operations we can perform on schemata seen as sets. Given any schema of order greater than 1, we can always represent it as the intersection of some other schemata of lower order. For example, the schema $h = {*}a{*}bc$ can be seen as the intersection between the schemata ${*}a{*}{*}{*}$, ${*}{*}{*}b{*}$ and ${*}{*}{*}{*}c$. So, we can write $h = *a*** \cap ***b* \cap ****c$. Because of this property, schemata of order 1 are particularly important, and, so, we introduce a special notation to represent them. We denote a schema of order 1 with its single defining symbol a at position s as H_s^a. Note this is simply a shorthand notation for $*^{s-1}a*^{\ell-s}$ that we can use whenever the length, ℓ, of the strings in the population is implied. So, if $\ell = 3$ we have $H_2^1 = *1*$ while $H_3^0 = **0$. Also, if $\ell = 5$, with this notation the schema $h = *a*bc$ can be rewritten as $h = H_2^a \cap H_4^b \cap H_5^c$.[2] More generally, a schema h of order n with defining symbols $a_1, a_2, \cdots, a_n$ at positions $l_1, l_2, \cdots, l_n$ can be represented as the following set intersection

$$h = \bigcap_{i=1}^{n} H_{l_i}^{a_i}.$$

The advantage of the set intersection notation is that it does not require the sequence l_i to be sorted. However, if the sequence l_i is ordered so that $l_1 \le l_2 \le \cdots \le l_n$, this same set can be represented using the string $h = *^{l_1-1}a_1*^{l_2-l_1-1}a_2*^{l_3-l_2-1}a_3 \cdots *^{l_n-l_{n-1}}a_n*^{\ell-l_n}$ or, more concisely, as

$$h = \bigotimes_{i=1}^{n}(*^{l_i-l_{i-1}-1}a_i)*^{\ell-l_n},$$

where we conventionally extended the l_i sequence by setting $l_0 = 0$ for notational convenience.

On some occasions it will be advantageous to use a concise notation for schemata of order higher than 1. Extending the order-1 schema notation H_s^a, we will denote with $H_{l_1,l_2,\cdots,l_n}^{a_1,a_2,\cdots,a_n}$ an order $n \le \ell$ schema with defining symbols $a_1, a_2, \cdots, a_n$ at positions $l_1, l_2, \cdots, l_n$. The interpretation is, of course, the one provided in the previous two equations. For example, a generic order-2 schema with alleles a and b at positions s and u (with $s \neq u$), respectively, is represented as $H_{s,u}^{a,b} = H_s^a \cap H_u^b$. Naturally, due to the commutativity of set intersection, there are symmetries in this representation, e.g. $H_{s,v}^{a,b} = H_{v,s}^{b,a}$. Also, clearly $H_{s,u}^{a,b}$ can also be represented using the string $*^{s-1}a*^{u-s-1}b*^{\ell-u}$ if $s < u$. If instead $s > u$, then $H_{s,u}^{a,b}$ is represented by $*^{u-1}b*^{s-u-1}a*^{\ell-s}$. As another example, order 3 schemata with exactly three alleles specified can be represented as $H_{s,u,q}^{a,b,c} = H_s^a \cap H_u^b \cap H_q^c$, where $s \neq u \neq q$.

As we have seen, all schemata of order 2 or above can be represented as intersections of order-1 schemata. Later in the paper we will compute intersections between order-1 schemata, and, so, we might wonder if the reverse of the afore-mentioned property holds. That is: are all order-1 schema intersections always representable as higher order schemata? The answer is "no": the intersection of any number of schemata is either a schema or it is the empty set $\emptyset$. For example, $*1*** \cap ***1* \cap *0*** = \emptyset$. A sufficient (but not necessary) condition that the result of $\bigcap_{i=1}^{n} H_{l_i}^{a_i}$ is guaranteed to be a schema of order n is that the schemata $H_{l_i}^{a_i}$ are all from different order-1 schema partitions.[3]

[2] Because the set-intersection operation commutes, we can also write $h = H_5^c \cap H_2^a \cap H_4^b$, etc.

[3] A set of sets $\{S_1, \cdots, S_n\}$ forms a partition of a set S if $S = \bigcup_i S_i$ and $\forall i \neq j$ we have that $S_i \cap S_j = \emptyset$. The search space can be partitioned using families of schemata, called *schema partitions* in the literature. The schemata in a schema partition are all of the same order and have don't care symbols at exactly the same loci. The whole set is obtained by setting defining symbols in the remaining loci in all possible ways. For example, for $\ell = 5$ the schemata $**0*0$, $**0*1$, $**1*0$ and $**1*1$ form a partition of Ω^ℓ. Schema partitions are often represented with strings of symbols from the alphabet $\{*, \#\}$, where the $\#$'s indicate the position of the defining characters in the schemata in a schema partition. In this notation the partition $\{**0*0, **0*1, **1*0, **1*1\}$ is represented as $**\#*\#$.

Although in the following we will mainly use schemata to coarse grain over a search space of fixed-length strings of a given length, we can also use them to coarse grain over other spaces that will be introduced later in the paper to model a GA using generalised recombination. In particular, whenever we have a space $\mathcal{V}^\ell$ of vectors $v = (v_1, \cdots, v_\ell)$ with elements $v_i \in \mathcal{V}$, where $\mathcal{V}$ is a generic set, we can use the schema notation to represent subsets of $\mathcal{V}^\ell$ where one or more coordinates take any value. For example, if $\mathcal{V} = \{1, 2, 3, 4\}$ and $\ell = 7$, the schema ***3*2* represents the set $\{v \in \mathcal{V}^\ell : v_4 = 3 \text{ and } v_6 = 2\}$. This will be very useful to coarse grain over crossover events, as will be shown in the following section.

Finally, a very important property of schemata in relation to probability distributions is the following. If a probability distribution $\Pr(v)$ is associated to the elements v of a space $\mathcal{V}^\ell$ (with $\sum_{v\in\mathcal{V}^\ell} \Pr(v) = 1$), be it the search space or some other space, and we consider the elements of the space as mutually exclusive events, then given a subset $S \subseteq \mathcal{V}^\ell$, the probability that one of the events $v \in S$ takes place is, clearly, $\Pr(S) = \sum_{v\in S} \Pr(v)$. Obviously, this applies also to schemata since they are special subsets. So, in the next sections every time we have a term of the form $\sum_{v\in h} \Pr(v)$ for some schema h and probability distribution $\Pr(v)$ we will replace that with $\Pr(h)$. Note that effectively $\Pr(h)$ is a marginal of $\Pr(v)$. This is the fundamental mechanism by which schemata allow coarse graining over the genetic dynamics of strings and over recombination events.

2.2 Generalised Recombination

In homologous recombination the loci in the offspring can by filled only by using alleles coming from corresponding loci in one of the parents. *Generalised recombination* is a form of recombination which allows the offspring's alleles to come from *any* locus in *any* of the parents.

In order to model mathematically a GA with this form of recombination we need a good notation to represent all possible recombination events. Homologous recombination events are often modelled using the notion of crossover mask. A crossover mask is a binary string of the same length as strings in the population. A 1 at a given position in the mask indicates that the allele at the corresponding position in the offspring should be taken from that locus in the first parent. A 0 in a given position of the mask indicates that the corresponding offspring's allele should be taken from the second parent. By restricting which masks are allowed and properly fixing the probability of choosing crossover masks (a distribution known as the recombination distribution), one can model any homologous crossover operator.

Crossover masks and recombination distributions 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 a model that allows 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 (see (Poli and Stephens, 2005b; 2005a; Stephens and Poli, 2005a)). Here, however, we focus only on one that naturally extends the notion of crossover masks and recombination distribution.

For strings of length ℓ, to represent a possible recombination event we use a *Generalised Crossover Mask* (GCM) $r = (m, v)$ where $m = m_1 \cdots m_\ell$ is an ℓ-component bit vector and $v = (v_1, \cdots, v_\ell)$ is a vector of integers whose components are in $\mathcal{N}_\ell = \{1, \cdots, \ell\}$.

So, $m \in \{0,1\} \otimes \{0,1\} \otimes \cdots \otimes \{0,1\} = \{0,1\}^\ell$ and $v \in \mathcal{N}_\ell \otimes \mathcal{N}_\ell \otimes \cdots \otimes \mathcal{N}_\ell = \mathcal{N}_\ell^\ell$, whereby we can see that GCMs live in the space $\mathcal{R}_\ell^\ell = \{0,1\}^\ell \otimes \mathcal{N}_\ell^\ell$, and so the total number of GCMs is $(2\ell)^\ell$, many more than the 2^ℓ possible masks for homologous recombination.

The semantics of the GCM representation is very simple. The elements in m specify which parent contributes the alleles that 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. More formally we can express 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 GCM $r = (m, v)$ as

$$h_i = m_i a_{v_i} + (1 - m_i) b_{v_i}, \tag{1}$$

where a_{v_i} is the allele from the first parent picked out by the GCM r, and similarly for b_{v_i} from the second.[4] Naturally, for our model of generalised recombination to be complete we need to specify the probability $p_c(r)$ of choosing any particular GCM, r. This is a generalisation of the notion of recombination distribution – the *Generalised Recombination Distribution* (GRD).

As a first example of how the representation of a GCM works consider the following example using standard one-point crossover for $\ell = 3$. The associated traditional crossover masks are 100 and 110, each invoked with probability $\frac{1}{2}$. These are equivalent to the GCMs $r_1 = (100, (1,2,3))$ and $r_2 = (110, (1,2,3))$. As a second example that illustrates the larger variety of ways in which parental genetic material can be distributed among the offspring, consider the case of $\ell = 2$, where the $(2 \times 2)^2 = 16$ GCMs 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 $p_c(r) = \frac{1}{16}$, 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 ordinary crossover masks. As a final example, the following GRD represents a single-parent inversion operator in the case of a three-locus system:

$$p_c(111, (2,1,3)) = p_c(111, (1,3,2)) = p_c(111, (3,2,1)) = \frac{1}{3}$$

In EA theory for homologous crossovers, the operation of bit-wise negation of a crossover mask is often used. Here we extend this notion to GCMs as follows. The *negation of a GCM* $r = (m, v)$ is the mask $\bar{r} = (\overline{m}, v)$, where $\overline{m}$ is the bit-wise negation of the bit-string m.

Since the GRD is a probability distribution over the space $\mathcal{R}_\ell^\ell$, we can use schemata (over this space) to represent entire classes of recombination events. Such schemata can be represented using GCMs in which some of the symbols have been replaced with $*$'s. The don't care symbols can be in the m component of a GCM, in the v component or both. The probability that a recombination event matching a *schema* R is the

[4] In this notation, homologous crossover events can be represented with GCMs of the form $r = (m, (1, 2, \cdots, \ell))$ where, effectively, m is the only element that can vary and, so, r can be seen as a traditional crossover mask.

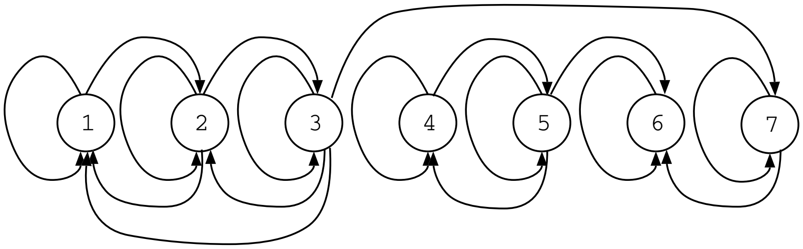

Figure 1: Example of order-1 mixing graph for $\ell = 7$.

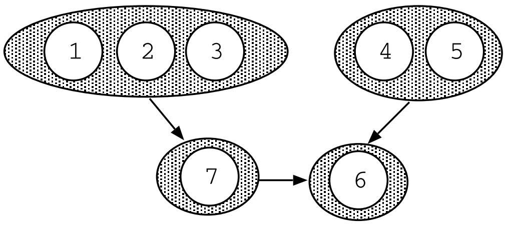

Figure 2: The order-1 recombination component graph for the mixing graph in Figure 1.

case is then given by $p_c(R)$ (see end of Section 2.1). So, for example, if $\ell = 5$, we can represent all events where crossover fills the first element of the offspring with the fourth element of the first parent with the schema $(1{*}{*}{*}{*}, 4{*}{*}{*}{*})$, where we omitted the commas between the elements of v for conciseness. The associated probability is $p_c(1{*}{*}{*}{*}, 4{*}{*}{*}{*}) = \sum_{r \in (1{*}{*}{*}{*}, 4{*}{*}{*}{*})} p_c(r)$.

2.3 Mixing Graphs and Recombination Components

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 set of *mixing graphs*. The first one of these is the *order-1 mixing graph*, the nodes of which 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. Figure 1 shows an example of an order-1 mixing graph for a 7-locus representation.

The network of causal influences is completely determined by the GRD. The elements c_{ij} of the connection matrix C for the order-1 mixing graph are given by

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

where $p_c({*}^{\ell}, {*}^{i-1} j {*}^{\ell-i})$ is a schema-based coarse graining of the GRD and $\delta(x) = 1$ if x is true, while $\delta(x) = 0$ otherwise. If there is a directed path between each pair of nodes in the order-1 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, i, of a particular string, $a_1 \cdots a_\ell$. An order-1 mixing generalised crossover allows for the migration of allele a to different loci in 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 recombination does not keep the alleles in their original position, i.e. allele a might migrate to loci different from i. Because of this, in repeated applications of crossover, a copy of the allele can be placed back into the original string $a_1 \cdots a_\ell$ (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 a mechanism for gene duplication in nature (Ridley, 1993)). 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 connection matrix is diagonal and so each node in the graph is isolated (having only a self-connection).

Naturally, many different intermediate situations are also possible. In all intermediate cases we can divide the order-1 mixing graph into two or more *order-1 recombination components*. These are characterised by the fact that all pairs of nodes in a component are mutually accessible by traversing only nodes and arcs in the component, while none of the nodes in a component is mutually accessible from any other node outside the component. In Figure 1, loci 1–3 form a recombination component, nodes 4 and 5 form another, and nodes 6 and 7 form two single-node components. Formally, recombination components are strongly connected components of the order-1 recombination graph. So, each locus belongs to one and only one component. Also, the components themselves form a directed acyclic graph (component graph) that we will call the *order-1 recombination component graph*. This has one node for each recombination component and an arc between two nodes if there is an edge between the corresponding components. Figure 2 shows an example of an order-1 recombination component graph.

We can define, in a very similar way, the notion of higher order mixing graphs, higher order recombination components, higher order component graphs and higher-order-mixing GRDs. For example, the order-2 mixing graph associated to a GRD includes $\binom{\ell}{2}$ nodes (labelled with all possible pairs of loci), and has connection matrix C with elements $c_{ij,mn}$ given by

$$c_{ij,mn} = \delta\big(p_c(*^\ell, *^{i-1}m*^{j-i-1}n*^{\ell-j}) > 0\big),$$

for $i < j$, while the elements $c_{ij,mn}$ for $i > j$ can be computed in a similar fashion.[5] If there is a directed path between each pair of nodes in the order-2 mixing graph (the mixing graph is strongly connected), we define the recombination to be *order-2 mixing*. Again, as for the case of the order-1 connection matrix, this property depends only on a set of relevant schema-based marginals of the GRD. Figure 3 shows an example of an order-2 mixing graph for a 4-locus representation. As one can easily verify, the graph is strongly connected and, so, the corresponding GRD is order-2 mixing.

We define a recombination distribution which is order-1 through to order-ℓ mixing a *fully mixing recombination distribution*.

3 Evolution Equations

In (Poli and Stephens, 2005b; 2005a) we derived evolution equations for strings and schemata for a GA using selection and generalised recombination. In (Stephens and Poli, 2005a) we extended the string and schema equations by adding bit-flip mutation and considering also the case of variable length strings. Here we summarise the main results providing only proof fragments wherever these are important for what follows.

[5] We use double indices to identify rows and columns. So, ij represents a row and mn a column of C.

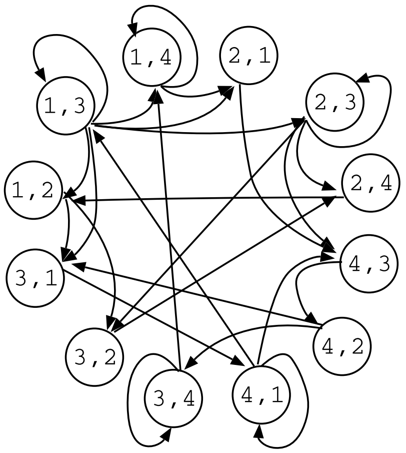

Figure 3: Example of order-2 mixing graph for $\ell = 4$. The graph has only one order-2 recombination component and the corresponding GRD is order-2 mixing.

3.1 Evolution Equations for Strings

Let us consider a generational evolutionary system with selection and generalised recombination exploring a search space of fixed-length strings Ω^ℓ. For simplicity let us consider the case where crossover is performed with 100% probability (as we will see below we can do this without loss of generality). Under these assumptions the expected frequency of a string $h = h_1 \cdots h_\ell \in \Omega^\ell$ is given by

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

where $E[\cdot]$ is the expectation operator, $\Phi(h, t+1)$ is the proportion of strings of type h in the population at generation $t+1$, $p(a,t)$ is the probability of selecting a string of type a as a parent from the population at generation t, $\mathcal{R}_\ell^\ell$ is the set of all possible GCMs, $p_c(r)$ 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. This takes the value 1 if h is created from a and b using the GCM r and 0 otherwise. Note that the string summations cover the entire search space Ω^ℓ rather than just the strings in the population because the selection probability for any string not currently in the population is zero.

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 GCM $r = (m, v)$ can be computed using Equation 1. From this, it follows that

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

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 $I_{\bar{r}} = \{i : m_i = 0\}$ is the complementary set picked out from the second parent via the negation of GCM r (see Section 2.2). As the full genetic composition of h has to come from the parents together we have $I_r \cup I_{\bar{r}} = \{1, 2, \cdots, \ell\}$. By substituting this result into the evolution equation for h and reordering terms, we obtain

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

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. These subsets are intersections of order one schemata of the form introduced in Section 2.1. More precisely, for each GCM r these are

$$\Gamma(h, I_r) = \bigcap_{i\in I_r} H_{v_i}^{h_i} \qquad \text{and} \qquad \Gamma(h, I_{\bar{r}}) = \bigcap_{i\in I_{\bar{r}}} H_{v_i}^{h_i}$$

with the conventions $\bigcap_{i\in\{j\}} H_{v_i}^{h_i} = H_{v_j}^{h_j}$ and $\bigcap_{i\in\emptyset} H_{v_i}^{h_i} = *^\ell$. We call these *building blocks* for the string h. This leads directly 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*

$$E[\Phi(h,t+1)] = \sum_{r\in\mathcal{R}_\ell^\ell} p_c(r) p(\Gamma(h, I_r), t) p(\Gamma(h, I_{\bar{r}}), t). \tag{2}$$

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 obtain these 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 $(2\ell)^\ell$ GCMs, 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 GCM, there remains the question of how many combinations of strings or schemata can lead to a particular offspring. This is where building block schemata play a crucial role as for a given GCM there is uniquely only *one* relevant pair of schemata and, therefore, correspondingly only one term in the r.h.s. of Equation 2. For strings, however, there are an exponential number of terms to consider. Also, even if there are an exponential number of GCMs, 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 (Stephens and Waelbroeck, 1999), 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 order-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.

Thus, we begin by considering what happens when we coarse grain such that $h_1 \cdots h_\ell \rightarrow \bigcup_{h_s} h_1 \cdots h_s \cdots h_\ell = h_1 \cdots h_{s-1}{*}h_{s+1} \cdots h_\ell$. Thus

$$\begin{aligned} E[\Phi(h_1 \cdots h_{s-1}{*}h_{s+1} \cdots h_\ell, t+1)] &= \sum_{h_s} E[\Phi(h_1 \cdots h_s \cdots h_\ell, t+1)] \\ &= \sum_{h_s} \sum_{r \in \mathcal{R}_\ell^\ell} p_c(r) p\left(\Gamma(h, I_r), t\right) p\left(\Gamma(h, I_{\bar{r}}), t\right). \end{aligned}$$

If we use the notation *expr*$/y \leftarrow z$ to mean "replace y with z in expression *expr*", clearly $E[\Phi(h_1 \cdots h_{s-1}{*}h_{s+1} \cdots h_\ell, t+1)] = E[\Phi(h\Big/h_s \leftarrow *, t+1)]$. Because in each crossover event allele h_s in an offspring comes from one parent or the other (depending on the value of the s-th bit in the m component of the corresponding GCM) but not both at the same time, it then follows that

$$E[\Phi(h\Big/h_s \leftarrow *, t+1)] \quad = \sum_{r \in \mathcal{R}_\ell^\ell} p_c(r) p\left(\Gamma(h, I_r)\Big/h_s \leftarrow *, t\right) p\left(\Gamma(h, I_{\bar{r}})\Big/h_s \leftarrow *, t\right) \tag{3}$$

(see (Poli and Stephens, 2005b; 2005a) for a proof). This result can be iterated to coarse grain over any number of variables (Poli and Stephens, 2005b; Poli and Stephens, 2005a), leading to the following

Theorem 2 (Schema evolution equation)**.** *Equation 2 is applicable to both strings and schemata of any order.*

Interestingly, we can collect some terms in Equation 3. To see that, let us assume, without loss of generality, that $m_s = 1$. In other words, $s \in I_r$. Therefore

$$\left(\Gamma(h, I_r)\Big/h_s \leftarrow *\right) \quad = \quad \left(\bigcap_{i \in I_r \setminus \{s\}} H_{v_i}^{h_i}\right) \cap \overbrace{\left(H_{v_s}^{h_s}\Big/h_s \leftarrow *\right)}^{=*^\ell} = \bigcap_{i \in I_r \setminus \{s\}} H_{v_i}^{h_i}$$

where the operator $\setminus$ represents set subtraction and $\{s\}$ is the singleton set including only the integer s.

A similar result holds for $s \in I_{\bar{r}}$. This means that neither m_s nor v_s appear explicitly in any of the terms in Equation 3, except in the factor $p_c(r)$. So, we can rewrite the equation as:

$$\begin{aligned} & E[\Phi(h_1 \cdots h_{s-1}{*}h_{s+1} \cdots h_\ell, t+1)] \\ & \quad = \sum p_c(m_1 \cdots m_{s-1}{*}m_{s+1} \cdots m_\ell, v_1 \cdots v_{s-1}{*}v_{s+1} \cdots v_\ell) \\ & \qquad p\left(\Gamma(h, I_r)\Big/h_s \leftarrow *, t\right) p\left(\Gamma(h, I_{\bar{r}})\Big/h_s \leftarrow *, t\right) \end{aligned} \tag{4}$$

where the summation ranges over all GCMs of the form $r = (m_1 \cdots m_{s-1} m_{s+1} \cdots m_\ell, v_1 \cdots v_{s-1} v_{s+1} \cdots v_\ell) \in \mathcal{R}_\ell^{\ell-1}$. Naturally this result generalises to any number of "don't care" symbols, leading to the following theorem.

Theorem 3. *For a schema h with d don't care symbols at positions $l_1, \cdots, l_d$, the summation in Equation 2 can be turned into a summation over $(m', v') \in \mathcal{R}_\ell^{\ell-d}$ provided the recombination distribution p_c is replaced with the marginal distribution p'_c obtained by summing $p_c(m, v)$ over all m_{l_i} and v_{l_i} for $1 \leq i \leq d$.*

3.3 A More Explicit Notation

As we saw in Section 2.1, the intersection of order-1 schemata can be either a schema or the empty set. When, for a given GCM $r = (m, v) \in \mathcal{R}_\ell^\ell$, v is a permutation of the vector $(1, 2, \cdots, \ell)$, then $\Gamma(h, I_r) = \bigcap_{i \in I_r} H_{v_i}^{h_i}$ is guaranteed to be an ordinary schema and the same applies to $\Gamma(h, I_{\bar{r}})$. In order to be able to express exactly which schemata these are, we need to order the sets $I_r = \{i_1, i_2, \cdots, i_{|I_r|}\}$ and $I_{\bar{r}} = \{j_1, j_2, \cdots, j_{|I_{\bar{r}}|}\}$ 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 likewise for $I_{\bar{r}}$. For example, if $\ell = 4$ and $r = (m, v) = (1101, (4, 3, 4, 1))$, then I_r is obtained as follows: As before, we first 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, $I_{\bar{r}} = \{3\}$.

With this ordering, when v is a permutation, then $v_{i_k} < v_{i_{k+1}}$ for all k. Therefore we can use the concatenation operator introduced in Section 2.1 to express our building block schemata

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

where we adopt the conventions of Section 2.1 that $v_{i_0} = 0$ and $*^0$ is the empty sequence. For example, if $r = (m, v)$ with $m = 1101$ and $v = (4, 3, 4, 1)$ as above, and $h = 1001$, then $I_r = \{4, 2, 1\}$ and

$$\begin{aligned}
\Gamma(h, I_r) &= (*^{v_{i_1}-1} h_{i_1})(*^{v_{i_2}-v_{i_1}-1} h_{i_2})(*^{v_{i_3}-v_{i_2}-1} h_{i_3}) *^{4-v_{i_3}} \\
&= (*^{v_4-1} h_4)(*^{v_2-v_4-1} h_2)(*^{v_1-v_2-1} h_1) *^{4-v_1} \\
&= (*^{1-1} h_4)(*^{3-1-1} h_2)(*^{4-3-1} h_1) *^{4-4} \\
&= h_4 * h_2 h_1
\end{aligned}$$

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

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

3.4 Symmetry Induced by Independent Parent Selection

Collecting terms with common factors in string and schema evolution equations reveals a symmetry with respect to the order in which parents are selected. Interestingly, this is captured by the notion of negation for GCMs introduced in Section 2.2.

Let us make the effects induced by the symmetry in parent selection explicit. In Equation 2, for any given GCM $r = (m, v)$ we have a term of the form

$$p_c(r) p(\Gamma(h, I_r), t) p(\Gamma(h, I_{\bar{r}}), t).$$

Obviously, we also have a term corresponding to the negation $\bar{r}$ of such a mask. This is

$$p_c(\bar{r})p(\Gamma(h, I_{\bar{r}}), t)p(\Gamma(h, I_{\bar{\bar{r}}}), t) = p_c(\bar{r})p(\Gamma(h, I_{\bar{r}}), t)p(\Gamma(h, I_r), t),$$

since $\bar{\bar{r}} = r$. Therefore, the two terms involve the same selection probabilities, although with different coefficients, namely $p_c(r)$ and $p_c(\bar{r})$. So, we can rewrite Equation 2 as

$$E[\Phi(h, t+1)] \quad = \quad \sum_{r \in \widetilde{\mathcal{R}}_\ell^\ell} (p_c(r) + p_c(\bar{r}))\, p(\Gamma(h, I_r), t)p(\Gamma(h, I_{\bar{r}}), t) \qquad (5)$$

where $\widetilde{\mathcal{R}}_\ell^\ell$ is any set representing half of the space of possible GCMs such that $\forall r \in \widetilde{\mathcal{R}}_\ell^\ell \implies \bar{r} \notin \widetilde{\mathcal{R}}_\ell^\ell$ (for example, $\widetilde{\mathcal{R}}_\ell^\ell$ could be the set of all GCM (m, v) with $m_1 = 1$, i.e. $\widetilde{\mathcal{R}}_\ell^\ell = (1{*}^{\ell-1}, {*}^\ell)$).

3.5 The Case $p_{xo} < 1$

Let us re-consider our earlier assumption that crossover is performed with 100% probability, i.e. that no offspring are created by selection followed by cloning (a.k.a. reproduction) alone. What would happen if recombination was applied with probability p_{xo} and reproduction with probability $(1 - p_{xo})$? The evolution equations for a generic schema h would transform into

$$E[\Phi(h, t+1)] \quad = \quad (1 - p_{xo})p(h, t) + p_{xo} \sum_{r \in \mathcal{R}_\ell^\ell} p_c(r)p(\Gamma(h, I_r), t)p(\Gamma(h, I_{\bar{r}}), t). \qquad (6)$$

This might seem rather different from Equation 2, but this is not the case. Because $\mathcal{R}_\ell^\ell$ includes the trivial GCM $r' = (11 \cdots 1, (1, 2, 3, \cdots, \ell))$ and because $I_{r'} = \mathcal{N}_\ell$ and $\bar{I}_{r'} = \emptyset$, then, since $\Gamma(h, \emptyset) = {*}^\ell$, $p({*}^\ell, t) = 1$ and $\Gamma(h, \mathcal{N}_\ell) = h$, the action of the GCM r is simply to produce an exact copy of the first parent. That is, when the mask r' is used, crossover behaves exactly like reproduction. So, we can model crossover and reproduction together by using Equation 2 but with a modified recombination distribution $p'_c(r) = p_{xo}p_c(r) + (1 - p_{xo})\delta(r = r')$. So, any results obtained with Equation 2 (that is for $p_{xo} = 1$) can be generalised to the case $p_{xo} < 1$.

3.6 Infinite Population Assumption

Infinitely large populations are a mathematical idealisation that we will use extensively in Part II. This is an essential assumption if we wish to separate the intrinsic operator biases from the bias due to genetic drift. Also, from a practical point of view, the infinite-population assumption makes life easier whenever one wants to iterate schema and string evolution equations to evaluate the behaviour of an EA over multiple time steps – a technique we will use, for example, to assess the interactions between selection and recombination. It is then natural to ask what sort of errors one should expect to see when going from finite to infinite populations. Interestingly, it is possible to assess this from the schema evolution equations themselves (see (Poli et al., 1998)).

Let us start by considering that for a finite population, schema (and string) frequencies are of the form $\Phi(h, t) = \frac{m(h,t)}{M}$ where $m(h, t)$ is the number of strings in the population which match the schema h and M is the population size. If we know the state of the population at time t, the number of strings matching a schema at the next generation, $m(h, t+1)$, is a stochastic variable which is binomially distributed (Poli et al., 1998). Its success probability $\pi_h(t)$ is given by the r.h.s. of Equation 2,

$$\pi_h(t) = \sum_{r \in \mathcal{R}_\ell^\ell} p_c(r)p(\Gamma(h, I_r), t)p(\Gamma(h, I_{\bar{r}}), t),$$

and so, we can re-interpret Equation 2 as a statement about the mean of $m(h, t+1)$, i.e.

$$E[m(h, t+1)] = M\pi_h(t).$$

Once we know the success probability for a binomially distributed variable, we can calculate all the moments of the distribution and, indeed, the distribution itself. In the specific case of the variable $m(h, t+1)$, we have

$$\Pr\{m(h, t+1) = k\} = \binom{M}{k} \pi_h(t)^k (1 - \pi_h(t))^{M-k}.$$

Of particular interest is the variance of $m(h, t+1)$, $\mathrm{Var}[m(h, t+1)] = M\pi_h(t)(1 - \pi_h(t))$. From this we can easily compute the standard deviation of $\Phi(h, t+1)$, obtaining

$$\mathrm{StdDev}[\Phi(h, t+1)] = \sqrt{\frac{\pi_h(t)(1 - \pi_h(t))}{M}}.$$

So, *the deviations we should expect in each generation between the infinite population model predictions and the finite-population behaviour are of the order* $O\left(\frac{1}{\sqrt{M}}\right)$. These deviations can be seen as "noise", and, so, naturally, their importance depends on the amplitude of the corresponding "signal", $\pi_h(t)$ (see discussion on signal-to-noise ratios in (Poli et al., 1998)).

In addition, it is possible to use exact schema evolution equations to exactly study finite population evolution. Indeed, it is easy to construct a Markov chain model for generalised recombination. This can be done by applying Equation 2 to calculate the expected frequencies of all strings in Ω^ℓ at the next generation. For a given population, these can then be used as the success probabilities for a multinomial distribution which, for any possible population, provides the probability of it being the successor of the current population. So, by iterating this process for all possible "current" populations, we obtain the entries of the transition matrix of the Markov chain (we used this technique, for example, in (Poli et al., 2004)). Then, iterating the chain gives an exact probabilistic characterisation of the behaviour of a GA with finite populations.

The main obstacle to this procedure is that, as a stochastic process, a GA has an enormous number of possible states (populations). 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 (Nix and Vose, 1992). So, a Markov chain for a GA requires an immense ($N \times N$) transition matrix, implying computations that are much worse than exponential. This is why we prefer to investigate finite-population effects using actual runs in Part II of this paper.

3.7 Examples

As an example, let us write Equation 2 for a generic string of length $\ell = 2$

$$\begin{aligned}
& E[\Phi(ab, t+1)] \qquad (7)\\
= \; & p_c(11, (1,1))p(a{*} \cap b{*}, t) + p_c(11, (1,2))p(ab, t) + p_c(10, (1,1))p(a{*}, t)p(b{*}, t)\\
+ \; & p_c(10, (1,2))p(a{*}, t)p({*}b, t) + p_c(11, (2,1))p(ba, t) + p_c(11, (2,2))p({*}a \cap {*}b, t)\\
+ \; & p_c(10, (2,1))p({*}a, t)p(b{*}, t) + p_c(10, (2,2))p({*}a, t)p({*}b, t)\\
+ \; & p_c(01, (1,1))p(b{*}, t)p(a{*}, t) + p_c(01, (1,2))p({*}b, t)p(a{*}, t)\\
+ \; & p_c(00, (1,1))p(a{*} \cap b{*}, t) + p_c(00, (1,2))p(ab, t) + p_c(01, (2,1))p(b{*}, t)p({*}a, t)\\
+ \; & p_c(01, (2,2))p({*}b, t)p({*}a, t) + p_c(00, (2,1))p(ba, t) + p_c(00, (2,2))p({*}a \cap {*}b, t).
\end{aligned}$$

Collecting terms with common factors in this equation clearly shows the symmetry with respect to the order in which parents are selected, as mentioned in Section 3.4. For example, in the terms $p_c(11,(1,1))p(a{*}\cap b{*},t)$ and $p_c(00,(1,1))p(a{*}\cap b{*},t)$ we have that $(11,(1,1)) = (\overline{00},(1,1)) = \overline{(00,(1,1))}$ (by definition of negation for GCMs).

If one replaces a and b with some values from Ω, all schema intersections of the form ${*}a \cap {*}b$ turn into either the order-1 schema ${*}a$ (if $a = b$) or into the empty set $\emptyset$, for which $p(\emptyset,t) = 0$, and so it is possible to further simplify the equation. For example, if $a = b = 1$ and all GCMs have equal probability ($p_c(r) = 1/16$), we obtain

$$E[\Phi(11,t+1)] = \frac{p(1{*},t)^2 + p(1{*},t) + 2p(1{*},t)p({*}1,t) + p({*}1,t)^2 + 2p(11,t) + p({*}1,t)}{8}.$$

Notice that in order to solve for the dynamics of the strings in Equation 7 we need to have a solution for the building blocks $a{*}$, ${*}a$, $b{*}$ and ${*}b$. Notice, also, that the expected frequency of ab is a linear function of the selection probabilities of that string and its permutation ba and a linear-quadratic function of the selection probabilities of lower order schemata (building blocks). That is, if we term the latter $\mathfrak{b}_{ab}(t)$, we can rewrite:

$$\begin{aligned}
&E[\Phi(ab,t+1)] = \\
&\quad (p_c(11,(1,2)) + p_c(00,(1,2)))p(ab,t) + (p_c(11,(2,1)) + p_c(00,(2,1)))p(ba,t) + \mathfrak{b}_{ab}(t),
\end{aligned}$$

where

$$\begin{aligned}
\mathfrak{b}_{ab}(t) &= p_c(11,(1,1))p(a{*}\cap b{*},t) + p_c(10,(1,1))p(a{*},t)p(b{*},t) \\
&+ p_c(10,(1,2))p(a{*},t)p({*}b,t) + p_c(11,(2,2))p({*}a\cap {*}b,t) \\
&+ p_c(10,(2,1))p({*}a,t)p(b{*},t) + p_c(10,(2,2))p({*}a,t)p({*}b,t) \\
&+ p_c(01,(1,1))p(b{*},t)p(a{*},t) + p_c(01,(1,2))p({*}b,t)p(a{*},t) \\
&+ p_c(00,(1,1))p(a{*}\cap b{*},t) + p_c(01,(2,1))p(b{*},t)p({*}a,t) \\
&+ p_c(01,(2,2))p({*}b,t)p({*}a,t) + p_c(00,(2,2))p({*}a\cap {*}b,t)
\end{aligned}$$

and acts as a "source" for creating the string ab from lower order objects.

As an example, the evolution equation for the schema $a{*}$ (a building block for ab) can be obtained by simply replacing b with $*$ in Equation 7, obtaining

$$\begin{aligned}
&E[\Phi(a{*},t+1)] \\
&= p_c(11,(1,1))p(a{*}\cap{*}{*},t) + p_c(11,(1,2))p(a{*},t) + p_c(10,(1,1))p(a{*},t)p({*}{*},t) \\
&+ p_c(10,(1,2))p(a{*},t)p({*}{*},t) + p_c(11,(2,1))p({*}a,t) + p_c(11,(2,2))p({*}a\cap{*}{*},t) \\
&+ p_c(10,(2,1))p({*}a,t)p({*}{*},t) + p_c(10,(2,2))p({*}a,t)p({*}{*},t) \\
&+ p_c(01,(1,1))p({*}{*},t)p(a{*},t) + p_c(01,(1,2))p({*}{*},t)p(a{*},t) \\
&+ p_c(00,(1,1))p(a{*}\cap{*}{*},t) + p_c(00,(1,2))p(a{*},t) + p_c(01,(2,1))p({*}{*},t)p({*}a,t) \\
&+ p_c(01,(2,2))p({*}{*},t)p({*}a,t) + p_c(00,(2,1))p({*}a,t) + p_c(00,(2,2))p({*}a\cap{*}{*},t) \\
&= \big(p_c(11,(1,1)) + p_c(11,(1,2)) + p_c(10,(1,1)) + p_c(10,(1,2)) + \\
&\qquad p_c(01,(1,1)) + p_c(01,(1,2)) + p_c(00,(1,1)) + p_c(00,(1,2))\big)\, p(a{*},t) \\
&+ \big(p_c(11,(2,1)) + p_c(11,(2,2)) + p_c(10,(2,1)) + p_c(10,(2,2)) + \\
&\qquad p_c(01,(2,1)) + p_c(01,(2,2)) + p_c(00,(2,1)) + p_c(00,(2,2))\big)\, p({*}a,t).
\end{aligned}$$

That is

$$\begin{aligned}E[\Phi(a*,t+1)] &= \big(p_c(**,(1,1))+p_c(**,(1,2))\big)p(a*,t)\\ &+ \big(p_c(**,(2,1))+p_c(**,(2,2))\big)p(*a,t)\\ &= p_c(**,(1,*))p(a*,t)+p_c(**,(2,*))p(*a,t),\end{aligned}$$

as predicted from Equation 4. Note that there is now no corresponding $\mathfrak{b}(t)$ for order-1 schemata as they cannot be created from any more elementary object.

A much deeper analysis of the $\ell = 2$ case is provided in (Stephens and Poli, 2005b), where a complete, exact solution, is derived, showing how the dynamical behaviour is radically different from that of homologous crossover. Even in such a simple case, new qualitatively different behaviours are 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 general form of the evolution equations for $\ell = 3$ for the generic string abc is much bigger than for $\ell = 2$ (see explicit expansion in (Poli and Stephens, 2005b; Poli and Stephens, 2005a)), this including 216 terms – a number that, although quite big, is only $\frac{1}{64}$ of the number of terms one would get in the absence of coarse graining. Interestingly, also in this case, the expected frequency of abc is a linear function of the selection probabilities of that string and all its permutations and a linear-quadratic function of the selection probabilities of lower order schemata (building blocks). That is:

$$\begin{aligned}E[\Phi(abc,t+1)] &= (p_c(111,(1,2,3))+p_c(000,(1,2,3)))p(abc,t)\\ &+ (p_c(111,(1,3,2))+p_c(000,(1,3,2)))p(acb,t)\\ &+ (p_c(111,(2,1,3))+p_c(000,(2,1,3)))p(bac,t)\\ &+ (p_c(111,(2,3,1))+p_c(000,(2,3,1)))p(cab,t)\\ &+ (p_c(111,(3,1,2))+p_c(000,(3,1,2)))p(bca,t)\\ &+ (p_c(111,(3,2,1))+p_c(000,(3,2,1)))p(cba,t)+\mathfrak{b}_{abc}(t)\end{aligned}$$

Again, the symmetry in the selection process induces a symmetry in the equations whereby terms relating to a mask and its negation always collect.

Naturally, in order to solve for the string dynamics we need to have the dynamics of the building blocks that determine the driving term $\mathfrak{b}_{abc}(t)$. Fortunately, again, the equations for the building blocks can be derived by trivial syntactic manipulations on the string equations (see (Poli and Stephens, 2005b; 2005a; Stphens and Poli, 2005a)).

4 Hierarchical Nature of Schema Evolution Equations

In the examples in Section 3.7 for $\ell = 2$ and $\ell = 3$, string and schema evolution equations have right-hand sides with the same structure, i.e. 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-1 objects, in which case there is no forcing term. So, as in the case of normal recombination a hierarchical structure emerges, where the dynamics of objects at a higher level of the hierarchy can be expressed in terms of the dynamics of objects at a lower level. Strings are at the highest level of the hierarchy and order-1 schemata the lowest. As we will show in a moment this hierarchical organisation is a general property of schema evolution equations. To show this we will need to analyse the terms in Equation 2 in more detail.

Let us consider a generic schema h and let us define $D(h)$ to be the set of loci where h has defining symbols, that is $D(h) = \{i | h_i \neq *\}$. Then, we can split the set of GCMs $\mathcal{R}_\ell^\ell$ into three disjoint sets:

$$\mathcal{R}_1(h) = \{r | D(h) \subseteq I_r\}, \quad \mathcal{R}_2(h) = \{r | D(h) \subseteq I_{\bar{r}}\}, \quad \text{and} \quad \mathcal{R}_3(h) = \mathcal{R}_\ell^\ell \setminus \mathcal{R}_1(h) \setminus \mathcal{R}_2(h),$$

where we could alternatively have expressed $\mathcal{R}_2(h) = \{r | D(h) \cap I_r = \emptyset\}$ or, extending the notion of negation from GCMs to sets of GCMs, as $\mathcal{R}_2(h) = \overline{\mathcal{R}}_1(h) = \{r | \bar{r} \in \mathcal{R}_1(h)\}$. Then we can rewrite Equation 2 as

$$\begin{aligned}
&E[\Phi(h, t+1)] \\
&= \sum_{r \in \mathcal{R}_1(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) + \sum_{r \in \mathcal{R}_2(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) \\
&+ \sum_{r \in \mathcal{R}_3(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) \\
&= \sum_{r \in \mathcal{R}_1(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) p(*^\ell, t) + \sum_{r \in \mathcal{R}_2(h)} p_c(r) p(*^\ell, t) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) \\
&+ \sum_{r \in \mathcal{R}_3(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) \\
&= \sum_{r \in \mathcal{R}_1(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) + \sum_{r \in \overline{\mathcal{R}}_1(h)} p_c(r) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) \\
&+ \sum_{r \in \mathcal{R}_3(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) \\
&= \sum_{r \in \mathcal{R}_1(h)} (p_c(r) + p_c(\bar{r})) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) + \sum_{r \in \mathcal{R}_3(h)} p_c(r) p(\bigcap_{i \in I_r} H_{v_i}^{h_i}, t) p(\bigcap_{i \in I_{\bar{r}}} H_{v_i}^{h_i}, t) \qquad (8)
\end{aligned}$$

Let us further split $\mathcal{R}_1(h)$ into

$$\mathcal{R}_1'(h) = \{r \in \mathcal{R}_1(h) : \forall i, j \in D(h), i \neq j \implies v_i \neq v_j\} \quad \text{and} \quad \mathcal{R}_1''(h) = \mathcal{R}_1(h) \setminus \mathcal{R}_1'(h).$$

Also, let us also assume that the elements of I_r are ordered based on the values of the elements of v (see Section 3.3). Then, for $r \in \mathcal{R}_1'(h)$ we have

$$\bigcap_{i \in I_r} H_{v_i}^{h_i} = \bigotimes_{k=1}^{|I_r|} \left(*^{v_{i_k} - v_{i_{k-1}} - 1} h_{i_k} \right) *^{\ell - v_{i_{|I_r|}}}$$

and so we can transform Equation 8 into the following theorem.

Theorem 4 (Schema hierarchy)**.** *The evolution of a string or schema h under selection and generalised recombination is governed by the following equation*

$$E[\Phi(h, t+1)] \quad = \sum_{r \in \mathcal{R}_1'(h)} (p_c(r) + p_c(\bar{r})) p \left(\bigotimes_{k=1}^{|I_r|} \left(*^{v_{i_k} - v_{i_{k-1}} - 1} h_{i_k} \right) *^{\ell - v_{i_{|I_r|}}}, t \right) + \mathfrak{b}_h(t) \tag{9}$$

the first component of which is linear in the selection probabilities of schemata obtained by *assigning the positions of the defining characters in h in all possible ways. The forcing term*

$$\begin{aligned}\mathfrak{b}_h(t) &= \sum_{r\in\mathcal{R}_1''(h)} (p_c(r)+p_c(\bar{r}))p(\bigcap_{i\in I_r} H_{v_i}^{h_i},t) \\ &+ \sum_{r\in\mathcal{R}_3(h)} p_c(r)p(\bigcap_{i\in I_r} H_{v_i}^{h_i},t)p(\bigcap_{i\in I_{\bar{r}}} H_{v_i}^{h_i},t)\end{aligned} \tag{10}$$

is linear-quadratic in the selection probabilities of schemata of lower order than h.

The order of the schemata $\bigotimes_{k=1}^{|I_r|}\left(*^{v_{i_k}-v_{i_{k-1}}-1}h_{i_k}\right)*^{\ell-v_{i_{|I_r|}}}$ is the same as h because, for $r\in\mathcal{R}_1'$, we have that $D(h)\subseteq I_r$ and $h_i = *$ for any $i\in I_r\setminus D(h)$. Also, the order of the schemata in $\mathfrak{b}_h(t)$ is lower than h for the following reasons: $[r\in\mathcal{R}_1''(h) \implies (\exists i\neq j; v_i=v_j)]$, and so $\bigcap_{i\in I_r} H_{v_i}^{h_i}$ is either $\emptyset$ or is a schema of lower order than h. Also, $[r\in\mathcal{R}_3(h) \implies (\exists i,j\in D(h), i\notin I_r, j\notin I_{\bar{r}})]$, and so $\bigcap_{i\in I_r} H_{v_i}^{h_i}$ and $\bigcap_{i\in I_{\bar{r}}} H_{v_i}^{h_i}$ are either $\emptyset$ or schemata of lower order than h.

Theorem 4 generalises the properties we observed in Section 3.7 for the cases $\ell=2$ and $\ell=3$. In particular, for order-1 schemata, the right-hand side of Equation 10 collapses to 0. This is because they cannot be composed of lower order objects.[6] So, in certain conditions, as in Part II of this paper, the dynamics of order-1 schemata is sufficient to reconstruct the dynamics of all higher-order objects.

5 Relationship with Previous Schema Theory

Let us analyse how Theorem 4 relates to previously published work on schema theory. As we have mentioned in Section 2.2 generalised recombination is more general than the notion of homologous crossover. Therefore, all general results obtained in this paper extend previous results including the exact schema theory for one-point crossover in (Stephens and Waelbroeck, 1999) and the exact schema theory for general homologous crossover in (Stephens, 2001).

As an illustrative example, in this section we will first derive a generalisation of Holland's schema theorem for the case of generalised recombination, and then we will show how Holland's original theorem emerges from a specialisation of this.

Theorem 5 (Generalised Holland's schema theorem)**.** *A lower bound for the expected frequency a string or schema h at the next generation under selection, reproduction and generalised recombination applied with probability p_{xo} is given by the following equation*

$$\begin{aligned}E[\Phi(h,t+1)] &\geq (1-p_{xo})p(h,t) \\ &+ p_{xo}\sum_{r\in\mathcal{R}_1(h)} (p_c(r)+p_c(\bar{r}))p(\bigcap_{i\in D(h)} H_{v_i}^{h_i},t) \\ &+ p_{xo}\sum_{r\in\mathcal{R}_3(h)} p_c(r)p(\bigcap_{i\in D(h)} H_{v_i}^{h_i},t)p(\bigcap_{i\in D(h)} H_{v_i}^{h_i},t)\end{aligned} \tag{11}$$

Proof. Remembering that $\mathcal{R}_1(h) = \{r|D(h)\subseteq I_r\}$, for $r\in\mathcal{R}_1(h)$ we have that $\bigcap_{i\in I_r} H_{v_i}^{h_i} = \bigcap_{i\in D(h)} H_{v_i}^{h_i}$ because $h_i = *$ for any $i\in I_r\setminus D(h)$. Also, recalling that $\mathcal{R}_3(h) = \mathcal{R}_\ell^\ell\setminus\mathcal{R}_1(h)\setminus\mathcal{R}_2(h)$ with $\mathcal{R}_2(h) = \{r|D(h)\subseteq I_{\bar{r}}\}$, for $r\in\mathcal{R}_3(h)$ we have that $\bigcap_{i\in D(h)} H_{v_i}^{h_i}\subset\bigcap_{i\in I_r} H_{v_i}^{h_i}$ and $\bigcap_{i\in D(h)} H_{v_i}^{h_i}\subset\bigcap_{i\in I_{\bar{r}}} H_{v_i}^{h_i}$ because the set $D(h)$ is always

[6] Formally one can verify this by observing that, for order-1 schemata, the set $D(h)$ contains only one element, and so all $r\in\mathcal{R}_\ell^\ell$ can only either be in $\mathcal{R}_1(h)$ or $\mathcal{R}_2(h)$. So, $\mathcal{R}_3(h)=\emptyset$. In addition, also $\mathcal{R}_1''(h)=\emptyset$.

split between I_r and $I_{\bar{r}}$. So, from Equation 8 we obtain

$$\begin{aligned} E[\Phi(h,t+1)] &= \sum_{r\in\mathcal{R}_1(h)} (p_c(r)+p_c(\bar{r}))p(\bigcap_{i\in I_r} H_{v_i}^{h_i},t) \qquad (12)\\ &+ \sum_{r\in\mathcal{R}_3(h)} p_c(r)p(\bigcap_{i\in I_r} H_{v_i}^{h_i},t)p(\bigcap_{i\in I_{\bar{r}}} H_{v_i}^{h_i},t) \\ &\geq \sum_{r\in\mathcal{R}_1(h)} (p_c(r)+p_c(\bar{r}))p(\bigcap_{i\in D(h)} H_{v_i}^{h_i},t) \qquad (13)\\ &+ \sum_{r\in\mathcal{R}_3(h)} p_c(r)p(\bigcap_{i\in D(h)} H_{v_i}^{h_i},t)p(\bigcap_{i\in D(h)} H_{v_i}^{h_i},t) \end{aligned}$$

By properly manipulating the recombination distribution, as indicated at the end of Section 3.1, we can then apply this result to the case in which crossover is invoked with probability p_{xo}. □

It is easy to see that this equation turns into Holland's schema theorem (to be more precise, into Whitley's version of it (Whitley, 1994)). For this purpose, we apply the previous theorem to the case of a homologous crossover operator.

Corollary 1. *A lower bound for the expected frequency of a string or schema, h, at the next generation under selection, reproduction and homologous recombination applied with probability p_{xo}, is given by the following equation*

$$E[\Phi(h,t+1)] \quad \geq \quad p(h,t)\left[1-p_{xo}\left(\sum_{r\in\mathcal{R}_3(h)} p_c(r)\right)\cdot(1-p(h,t))\right]. \qquad (14)$$

Proof. For homologous crossover all elements of the vector v must satisfy $v_i = i$. So,

$$\bigcap_{i\in D(h)} H_{v_i}^{h_i} = \bigcap_{i\in D(h)} H_i^{h_i} = h$$

and Equation 11 turns into

$$\begin{aligned} E[\Phi(h,t+1)] &\geq (1-p_{xo})p(h,t) \\ &+ p_{xo}p(h,t)\sum_{r\in\mathcal{R}_1(h)}(p_c(r)+p_c(\bar{r})) + p_{xo}p(h,t)^2\sum_{r\in\mathcal{R}_3(h)} p_c(r) \\ &= p(h,t)\left[1-p_{xo}\left(1-\underbrace{\sum_{r\in\mathcal{R}_1(h)}(p_c(r)+p_c(\bar{r}))}_{=1-\sum_{r\in\mathcal{R}_3(h)} p_c(r)} - p(h,t)\sum_{r\in\mathcal{R}_3(h)} p_c(r)\right)\right] \\ &= p(h,t)\left[1-p_{xo}\left(\sum_{r\in\mathcal{R}_3(h)} p_c(r)\right)\cdot(1-p(h,t))\right]. \qquad (15) \end{aligned}$$

□

Equation 14 is a more recognisable result. Indeed, under one-point crossover, the sum over crossover masks $\sum_{r\in\mathcal{R}_3(h)} p_c(r)$ turns into the traditional $\frac{\mathcal{L}(h)}{\ell-1}$, where $\mathcal{L}(h)$ is the defining length (see Section 2.1) of the schema h. So, Equation 14 *is* a version of Holland's schema theorem (see (Whitley, 1994)).

6 Conclusions and Summary of Part II

In this two-part paper we provide, within the context of a fixed length representation, a single, unified theoretical framework that is powerful enough to exactly model genetic systems that exhibit a rich array of genetic operators, far beyond those of the canonical GA. These include, for instance: gene duplication, gene deletion, inversion, homologous recombination, permutations, diploidy, multiple chromosomes etc. that are not only known to happen in nature but that have also been fruitfully used in EAs. This model includes as special cases previous models, such as the exact schema theory in (Stephens and Waelbroeck, 1999; Stephens, 2001) and the approximate model in (Whitley, 1994). Our overall analysis is motivated by the objective of understanding the search biases induced by such a large and powerful set of genetic operators.

In Part I, after describing and characterising the notion of generalised recombination, we derived both microscopic and coarse-grained evolution equations for strings and schemata and illustrated their features with simple examples. Also, we explained the hierarchical nature of the schema evolution equations and showed how the theory presented here generalises past work in EC.

In Part II, we study the biases of generalised recombination by finding the fixed points of the evolution equations for order-1 schemata for different classes of recombination distributions and under the standard assumption of infinite populations. As mentioned, understanding these biases is important, as the order-1 schemata determine the existence and stability of the fixed-points for higher-order schemata and strings as well. These are also dealt with in Part II. These infinite-population results allow us to construct a clear picture of the biases of generalised recombination for a general class of recombination distributions and to generalise Geiringer's theorem. In Part II we also discuss the expected behaviour of an evolutionary system under generalised recombination and selection. There we also consider how finite-population effects, such as drift, can alter the dynamics of the system. In addition, we test our quantitative and qualitative predictions by directly integrating the evolution equations for an infinite population with generalised recombination, with and without selection. We extend the technical analysis of generalised recombination by performing real runs of a GA with finite populations of different sizes, again, with and without selection. Although modelling natural evolution is beyond the scope of this paper, to illustrate the representational power of generalised recombination and the theory presented in the paper, in Part II we also show how one can generate models for non-binary strings, for systems with diploid representations, and even systems where multiple chromosomes coexist.

Acknowledgments

We thank Bill Langdon, Marc Schoenauer, the anonymous referees and the associate editor for their help and comments. EPSRC is thanked for financial support (grant number GR/T24616/01). Leverhulme Trust is thanked for a visiting professorship (F/OO213/J) for CRS. CRS thanks DGAPA of the UNAM for a Sabbatical Fellowship and Conacyt project 30422-E.

References

Clark, A. (1994). Invasion and maintenance of a gene duplication. *Proceedings of the National Academy of Sciences*, 91:2950–2954.

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

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

Koza, J. R. (1995). Gene duplication to enable genetic programming to concurrently evolve both the architecture and work-performing steps of a computer program. In *IJCAI-95 Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 734–740, Montreal, Quebec, Canada. Morgan Kaufmann.

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

McPhee, N. F. and Poli, R. (2002). Using schema theory to explore interactions of multiple operators. In Langdon, W. B., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M. A., Schultz, A. C., Miller, J. F., Burke, E., and Jonoska, N., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 853–860, New York. Morgan Kaufmann Publishers.

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

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

Poli, R. (2003). A simple but theoretically-motivated method to control bloat in genetic programming. In Ryan, C., Soule, T., Keijzer, M., Tsang, E., Poli, R., and Costa, E., editors, *Genetic Programming, Proceedings of the 6th European Conference, EuroGP 2003*, LNCS, pages 211–223, Essex, UK. Springer-Verlag.

Poli, R., Langdon, W. B., and O'Reilly, U.-M. (1998). Analysis of schema variance and short term extinction likelihoods. In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Riolo, R., editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 284–292, University of Wisconsin, Madison, Wisconsin, USA. Morgan Kaufmann.

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

Poli, R. and McPhee, N. F. (2003b). General schema theory for genetic programming with subtree-swapping crossover: Part II. *Evolutionary Computation*, 11(2):169–206.

Poli, R., McPhee, N. F., and Rowe, J. E. (2004). 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.

Poli, R., Rowe, J. E., and McPhee, N. F. (2001). 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. Morgan Kaufmann.

Poli, R. and Stephens, C. R. (2005a). Theoretical analysis of generalised recombination. Technical Report CSM-426, Department of Computer Science, University of Essex.

Poli, R. and Stephens, C. R. (2005b). Theoretical analysis of generalised recombination. In *CEC-2005*, pp 411-418, IEEE Press.

Ridley, M. (1993). *Evolution*. Blackwell Scientific Publications, Boston.

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

Sawai, H. and Adachi, S. (2000). A comparative study of gene-duplicated GAs based on pfGA and SSGA. In Whitley, D., Goldberg, D., Cantu-Paz, E., Spector, L., Parmee, I., and Beyer, H.-G., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 74–81, Las Vegas, USA. Morgan Kaufmann.

Stephens, C. R. (2001). Some exact results from a coarse grained formulation of genetic dynamics. In Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 631–638, San Francisco, California, USA. Morgan Kaufmann.

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

Stephens, C. R. and Poli, R. (2005a). Coarse grained dynamics for generalized recombination. Technical Report CSM-432, Department of Computer Science, University of Essex.

Stephens, C. R. and Poli, R. (2005b). Coarse graining in an evolutionary algorithm with recombination, duplication and inversion. In *CEC-2005*, pp1683-1690, IEEE Press.

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

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

Stephens, C. R., Waelbroeck, H., and Aguirre, R. (1999). Schemata as building blocks: Does size matter? In Banzhaf, W. and Reeves, C., editors, *Foundations of Genetic Algorithms 5*, pages 117–133. Morgan Kaufmann, San Francisco, CA.

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

Whitley, D. (1992). An executable model of a simple genetic algorithm. In Whitley, D., editor, *Foundations of Genetic Algorithms Workshop (FOGA-92)*, Vail, Colorado.

Whitley, L. D. (1994). A Genetic Algorithm Tutorial. *Statistics and Computing*, 4:65–85.