

---

# Markov Chain Models for GP and Variable-length GAs with Homologous Crossover

---

**Riccardo Poli**

School of Computer Science  
The University of Birmingham  
Birmingham, B15 2TT, UK  
R.Poli@cs.bham.ac.uk

**Jonathan E. Rowe**

School of Computer Science  
The University of Birmingham  
Birmingham, B15 2TT, UK  
J.E.Rowe@cs.bham.ac.uk

**Nicholas Freitag McPhee**

Division of Science and Mathematics  
University of Minnesota, Morris  
Morris, MN, USA  
mcphee@mrs.umn.edu

## Abstract

In this paper we present a Markov chain model for GP and variable-length GAs with homologous crossover: a set of GP operators where the offspring are created preserving the position of the genetic material taken from the parents. We obtain this result by using the core of Vose's model for GAs in conjunction with a specialisation of recent GP schema theory for such operators. The model is then specialised for the case of GP operating on 0/1 trees: a tree-like generalisation of the concept of binary string. For these symmetries exist that can be exploited to obtain further simplifications. In the absence of mutation, the theory presented here generalises Vose's GA model to GP and variable-length GAs.

## 1 Introduction

After a strong initial interest in schemata [1, 2], the interest of GA theorists has shifted in the last decade towards microscopic Markov chain models, such as Vose's model, possibly with aggregated states [3, 4, 5, 6, 7, 8, 9, 10, 11].

In the last year or so the theory of schemata has made considerable progress, both for GAs and GP. This includes several new schema theorems which give exact formulations (rather than the lower bounds previously presented in the literature [12, 13, 14, 15, 16, 17, 18]) for the expected number of instances of a schema at the next generation. These exact theories model GP with one-point crossover [19, 20, 21], standard and other subtree-swapping crossovers [22, 23, 24], homologous crossover [25], and different types of subtree mutation and headless chicken crossover [26, 27]. While considerable progress has been made in GP schema theory, no Markov chain model for GP and variable-length GAs has ever been proposed.

In this paper we start filling this theoretical gap and present

a Vose-like Markov-chain model for genetic programming with homologous crossover [25]: a set of operators, including GP one-point crossover [16] and GP uniform crossover [28], where the offspring are created preserving the position of the genetic material taken from the parents. We obtain this result by using the core of Vose's theory in conjunction with a specialisation of the schema theory for such operators. This formally links GP schema theory and Markov chain models, two worlds believed by many people to be quite separate.

The paper is organised as follows. Given the complexity of the GP mechanics, exact GP schema theories, such as the exact schema theory for homologous crossover in [25], tend to be relatively complicated. Similarly, Vose's model for GAs [3] presents significant complexities. In the following section, we will summarise these theories providing as much detail as reasonable, occasionally referring to [3] and [25] for more details. Then, in Section 3 we present the extensions to both theories which allow the construction of a Markov chain model for GP and variable-length GAs with homologous crossover. In Section 4 we indicate how the theory can be simplified thanks to symmetries which exist when we restrict ourselves to 0/1 trees: a tree-like generalisation of the concept of binary string. In Section 5 we give an example. Some conclusions are drawn in Section 6.

## 2 Background

### 2.1 Nix and Vose's Markov Chain Model of GAs

The description provided here is largely based on [3, 29] and [4]. See [30] for a gentler introduction to this topic.

Let  $\Omega$  be the set of all possible strings of length  $l$ , i.e.  $\Omega = \{0, 1\}^l$ . Let  $r = |\Omega| = 2^l$  be the number of elements of such a space. Let  $P$  be a population represented as a multiset of elements from  $\Omega$ , let  $n = |P|$  be the population size, and let  $N$  be the number of possible populations;

in [3] it was shown that

$$N = \binom{n+r-1}{r-1}.$$

Let  $Z$  be an  $r \times N$  matrix whose columns represent the possible populations of size  $n$ . The  $i$ th column  $\Phi_i = \langle z_{0,i}, \dots, z_{r-1,i} \rangle^T$  of  $Z$  is the incidence vector for the  $i$ th population  $P_i$ . That is  $z_{y,i}$  is the number of occurrences of string  $y$  in  $P_i$  (where  $y$  is unambiguously interpreted as an integer or as its binary representation depending on the context).

Once this state representation is available, one can model a GA with a Markov chain in which the  $N$  columns of  $Z$  represent the states of the model. The transition matrix for the model,  $Q$ , is an  $N \times N$  matrix where the entry  $Q_{ij}$  represents the conditional probability that the next generation will be  $P_j$  assuming that the current generation is  $P_i$ .

In order to determine the values  $Q_{ij}$  let us assume that we know the probability  $p_i(y)$  of producing individual  $y$  in the next generation given that the current generation is  $P_i$ . To produce population  $P_j$  we need to get exactly  $z_{y,j}$  copies of string  $y$  for  $y = 0, \dots, r-1$ . The probability of this joint event is given by a multinomial distribution with success probabilities  $p_i(y)$  for  $y = 0, \dots, r-1$ , so [31]

$$Q_{i,j} = \frac{n!}{z_{0,j}! z_{1,j}! \dots z_{r-1,j}!} \prod_{y=0}^{r-1} (p_i(y))^{z_{y,j}}. \quad (1)$$

The calculations necessary to compute the probabilities  $p_i(y)$  depend crucially on the representation and the operators chosen. In [4] results for various GA crossover operators were reported. As noted in [3], it is possible to decompose the calculations using ideas firstly introduced in [29] as follows.

Assuming that the current generation is  $P_i$ , we can write

$$p_i(y) = \sum_{m,n=0}^{r-1} s_{m,i} s_{n,i} r_{m,n}(y) \quad (2)$$

where  $r_{m,n}(y)$  is the probability that crossing over strings  $m$  and  $n$  yields string  $y$  and  $s_{x,i}$  is the probability of selecting  $x$  from  $P_i$ . Assuming fitness proportionate selection,

$$s_{x,i} = \frac{z_{x,i} f(x)}{\sum_{j=0}^{r-1} z_{j,i} f(j)}, \quad (3)$$

where  $f(x)$  is the fitness of string  $x$ .

We can map these results into a more recent formulation of Vose's model [4] by making use of matrices and operators. We start by treating the fitness function as a vector  $f$  of components  $f_k = f(k)$ . Then, if  $x$  is the incidence

vector representing a particular population, we define an operator  $\mathcal{F}$ , called the *selection scheme*,<sup>1</sup> which computes the selection probabilities  $s_{x,i}$  for all the members of  $\Omega$ . For proportional selection

$$\mathcal{F}(x) = \text{diag}(f)x / f^T x.$$

Then we organise the probabilities  $r_{m,n}(y)$  into  $r$  arrays  $M_y$  of size  $r \times r$ , called *mixing matrices*, the elements of which are  $(M_y)_{m,n} = r_{m,n}(y)$ . We finally define an operator  $\mathcal{M}$ , called the *mixing scheme*,

$$\mathcal{M}(x) = \langle x^T M_0 x, x^T M_1 x, \dots, x^T M_{r-1} x \rangle$$

which returns a vector whose components are the expected proportion of individuals of each type assuming that individuals are selected from the population  $x$  randomly (with replacement) and crossed over.

Finally we introduce the operator  $\mathcal{G} = \mathcal{M} \circ \mathcal{F}$ , which provides a compact way of expressing the probabilities  $p_i(y)$  since (for fitness proportionate selection)

$$p_i(y) = \{\mathcal{G}(\Phi_i)\}_y = \left\{ \mathcal{M} \left( \frac{\text{diag}(f)\Phi_i}{f^T \Phi_i} \right) \right\}_y$$

where the notation  $\{\cdot\}_y$  is used to represent the  $y$ th component of a vector. So, the entries of the transition matrix for the Markov chain model of a GA can concisely be written as

$$Q_{i,j} = n! \prod_{y=0}^{r-1} \frac{(\{\mathcal{G}(\Phi_i)\}_y)^{z_{y,j}}}{z_{y,j}!}. \quad (4)$$

In [29, 3, 4] it is shown how, for fixed-length binary GAs, the operator  $\mathcal{M}$  can be calculated as a function of the mixing matrix  $M_0$  only. This is done by using a set of permutation operators which permute the components of any generic vector  $x \in \mathbb{R}^r$ :

$$\sigma_j \langle x_0, \dots, x_{r-1} \rangle^T = \langle x_{j \oplus 0}, \dots, x_{j \oplus r-1} \rangle^T, \quad (5)$$

where  $\oplus$  is a bitwise XOR.<sup>2</sup> Then one can write

$$\mathcal{M}(x) = \langle (\sigma_0 x)^T M_0 \sigma_0 x, \dots, (\sigma_{r-1} x)^T M_0 \sigma_{r-1} x \rangle^T. \quad (6)$$

## 2.2 Exact GP Schema Theory for Homologous Crossover

In [25] the following exact schema theorem for GP with homologous crossover was reported:

<sup>1</sup>In this paper we have chosen to use the symbol  $\mathcal{F}$  to represent both the selection scheme of a GA and the function set used in GP, since this is the standard notation for both. This produces no ambiguity since the selection scheme is not used outside this section, and the function set is not referred to inside it.

<sup>2</sup>The operators  $\sigma_j$  can also be interpreted as permutation matrices.

$$\alpha(H, t) = (1 - p_{xo})p(H, t) + p_{xo} \sum_j \sum_k \sum_{l \in \mathcal{X}_{C(G_j, G_k)}} p_l^{C(G_j, G_k)} p(\Gamma(H, l) \cap G_j, t) p(\Gamma(H, \bar{l}) \cap G_k, t) \quad (7)$$

where

- $H$  is a GP schema, i.e. a tree composed of functions from the set  $\mathcal{F} \cup \{=\}$  and terminals from the set  $\mathcal{T} \cup \{=\}$ , where  $\mathcal{F}$  and  $\mathcal{T}$  are the function and terminals sets used in our GP system and the primitive  $=$  is a “don’t care” symbol which stands for a *single* terminal or function.
- $\alpha(H, t)$  is the probability that a newly created individual matches the schema  $H$ .
- $p_{xo}$  is the crossover probability.
- $p(H, t)$  is the selection probability of the schema  $H$ .
- $G_1, G_2, \dots$  are all the possible program shapes, i.e. all the possible schemata containing  $=$  signs only.
- $C(G_j, G_k)$  is the common region between programs of shape  $G_j$  and programs of shape  $G_k$ . The common region between two generic trees  $h_1$  and  $h_2$  is the set

$$C(h_1, h_2) = \{(d, i) | C(d, i, h_1, h_2)\},$$

where  $(d, i)$  is a pair of coordinates in a Cartesian node reference system (see [22, 25] for more details on the reference system used). The predicate  $C(d, i, h_1, h_2)$  is true if  $(d, i) = (0, 0)$ . It also true if  $A(d-1, i', h_1) = A(d-1, i', h_2) \neq 0$  and  $C(d-1, i', h_1, h_2)$  is true, where  $A(d, i, h)$  returns the arity of the node at coordinates  $(d, i)$  in  $h$ ,  $i' = \lfloor i/a_{\max} \rfloor$  and  $\lfloor \cdot \rfloor$  is the integer-part function. The predicate is false otherwise.

- For any given common region  $c$  we can define a set of *GP crossover masks*,  $\mathcal{X}_c$ , which contains all different trees with the same size and shape as the common region which can be built with nodes labelled 0 and 1.
- The *GP recombination distribution*  $p_l^c$  gives the probability that, for a given common region  $c$ , crossover mask  $l$  will be chosen from the set  $\mathcal{X}_c$ .
- A *GP hyperschema* is a rooted tree composed of internal nodes from  $\mathcal{F} \cup \{=\}$  and leaves from  $\mathcal{T} \cup \{=, \#\}$ . Again,  $=$  is a “don’t care” symbols which stands for exactly one node, while  $\#$  stands for any valid subtree.

- $\Gamma(H, l)$  is defined to be the empty set if  $l$  contains any node not in  $H$ . Otherwise it is the hyperschema obtained by replacing certain nodes in  $H$  with either  $=$  or  $\#$  nodes:
  - If a node in  $H$  corresponds to (i.e., has the same coordinates as) a non-leaf node in  $l$  that is labelled with a 0, then that node in  $H$  is replaced with a  $=$ .
  - If a node in  $H$  corresponds to a leaf node in  $l$  that is labelled with a 0, then it is replaced with a  $\#$ .
  - All other nodes in  $H$  are left unchanged.
- $\bar{l}$  is the complement of the GP crossover mask  $l$ . The complement of a mask is a tree with the same structure but with the 0’s and 1’s swapped.

### 3 Markov Chain Model for GP

In order to extend Vose’s model to GP and variable-length GAs with homologous crossover we define  $\Omega$  to be an indexed set of all possible trees of maximum depth  $\ell$  that can be constructed with a given function set  $\mathcal{F}$  and a given terminal set  $\mathcal{T}$ . Assuming that the initialisation algorithm selects programs in  $\Omega$ , GP with homologous crossover cannot produce programs outside  $\Omega$ , and  $\Omega$  is therefore a finite search space. Again,  $r = |\Omega|$  is the number of elements in the search space; this time, however,  $r$  is not  $2^\ell$ . All other quantities defined in Section 2.1 can be redefined by simply replacing the word “string” with the word “program”, provided that the elements of  $\Omega$  are indexed appropriately. With these extensions, all the equations in that section are also valid for GP, except Equations 5 and 6.

These are all minor changes. A major change is instead required to compute the probabilities  $p_i(y)$  of generating the  $y$ th program in  $\Omega$  when the population is  $P_i$ . Fortunately, these probabilities can be computed by applying the schema theory developed in [25] and summarised in Section 2.2. Since schema equations are applicable to schemata as well as to individual programs, it is clear that:

$$p_i(y) = \alpha(y, t) \quad (8)$$

where  $\alpha$  is calculated for population  $P_i$ . This can be done by specialising Equation 7. Doing this allows one to instantiate the transition matrix for the model using Equation 1. However, it is possible to express  $p_i(y)$  in terms of more primitive quantities as follows.

Let us specialise Equation 7 for the  $y$ th program in  $\Omega$ :

$$p_i(y) = (1 - p_{xo})p(y, t) + p_{xo} \sum_j \sum_k \sum_{l \in \mathcal{X}_{C(G_j, G_k)}} p_l^{C(G_j, G_k)} \times p(\Gamma(y, l) \cap G_j, t) p(\Gamma(y, \bar{l}) \cap G_k, t)$$

$$\begin{aligned}
&= (1 - p_{xo}) \sum_{h_1 \in \Omega} \delta(h_1 = y) p(h_1, t) \times \underbrace{\sum_{h_2 \in \Omega} p(h_2, t)}_{=1} \\
&\quad + p_{xo} \sum_j \sum_k \sum_{l \in \mathcal{X}(G_j, G_k)} p_l^{C(G_j, G_k)} \times \\
&\quad \sum_{h_1 \in \Omega} p(h_1, t) \delta(h_1 \in \Gamma(y, l)) \delta(h_1 \in G_j) \times \\
&\quad \sum_{h_2 \in \Omega} p(h_2, t) \delta(h_2 \in \Gamma(y, \bar{l})) \delta(h_2 \in G_k) \\
&= \sum_{h_1, h_2 \in \Omega} p(h_1, t) p(h_2, t) \times \\
&\quad \left[ (1 - p_{xo}) \delta(h_1 = y) + p_{xo} \sum_{l \in \mathcal{X}_{C(h_1, h_2)}} p_l^{C(h_1, h_2)} \times \right. \\
&\quad \left. \delta(h_1 \in \Gamma(y, l)) \delta(h_2 \in \Gamma(y, \bar{l})) \right],
\end{aligned}$$

where we used the fact that  $\sum_w \delta(x \in G_w) = 1$ .

Assuming the current population is  $P_i$ , we have that  $p(h, t) = s_h(t)$ . So, the last equation can be rewritten in the same form as Equation 2 provided we set

$$\begin{aligned}
r_{m,n}(y) &= \left[ (1 - p_{xo}) \delta(m = y) + \right. \\
&\quad \left. p_{xo} \sum_{l \in \mathcal{X}_{C(m,n)}} p_l^{C(m,n)} \delta(m \in \Gamma(y, l)) \delta(n \in \Gamma(y, \bar{l})) \right]. \tag{9}
\end{aligned}$$

Note that this equation could have been obtained by direct calculation, rather than through the specialisation of a schema theorem. However, this would still have required the definition and use of the hyperschema-returning function  $\Gamma$  and of the concepts of GP crossover masks and GP recombination distributions. Also, notice that the set of GP crossover masks also include masks containing all ones. These correspond to cloning the first parent. Therefore, by suitable readjustment of the probabilities  $p_l^{C(m,n)}$ , we can rewrite Equation 9 as

$$r_{m,n}(y) = \sum_{l \in \mathcal{X}_{C(m,n)}} p_l^{C(m,n)} \delta(m \in \Gamma(y, l)) \delta(n \in \Gamma(y, \bar{l})). \tag{10}$$

This formula is analogous to the case of crossover defined by masks for fixed-length binary strings [4].

## 4 Mixing Matrices for 0/1 Trees

As has already been stated in Section 2.1, for the case of fixed-length binary strings, the mixing operator  $\mathcal{M}$  can be written in terms of a single mixing matrix  $M_0$  and a group of permutation matrices. This works because the permutation matrices are a representation of a group that acts transitively on the search space. This group action describes the symmetries that are inherent in the definition of crossover

for fixed-length strings [4]. This idea can be generalised to other finite search spaces (see [32] for the detailed theory). However, in the case of GP, where the search space is a set of trees (up to some depth), the amount of symmetry is more limited and does not seem to give rise to a single mixing matrix.

In this section we will look at what symmetry does exist and the simplifications of the mixing operator it produces when we restrict ourselves to the space of *0/1 trees*. These are trees constructed using primitives from a terminal set  $\mathcal{T} = \{0_0, 1_0\}$  and from a function set  $\mathcal{F} = \bigcup_{i \in \iota} \mathcal{F}_i$  where  $\mathcal{F}_i = \{0_i, 1_i\}$ ,  $\iota$  is a finite subset of  $\mathbb{N}$ , and the subscripts 0 and  $i$  represent the arity of a 0/1 primitive.<sup>3</sup> It should be noted that the semantics of the primitives in 0/1 trees is unimportant for the theory, and that 0/1 trees are a generalisation of the notion of binary strings.<sup>4</sup>

Let  $\Omega$  be the set of 0/1 trees of depth at most  $\ell$  (where a program containing only a terminal has depth 1). Let  $L(\Omega)$  be the set of full trees of exactly depth  $\ell$  obtained by using the primitive set  $\mathcal{T} \cup \mathcal{F}_{i_m}$  where  $i_m$  is the maximum element in  $\iota$ . We term *node-wise XOR* the operation which, given two trees  $a$  and  $b$  in  $L(\Omega)$ , returns the 0/1 tree whose nodes are labelled with the result of the addition (modulo 2) of the binary labels of the nodes in  $a$  and  $b$  having corresponding coordinates; this operator is denoted  $a \oplus b$ .

For example, if we represent 0/1 trees in prefix notation,  $(1(101)(001)) \oplus (0(100)(011)) = (1(001)(010))$ .  $L(\Omega)$  is a group under node-wise XOR. Notice that the definition of  $\oplus$  extends naturally to pairs of trees with identical size and shape.

For each tree  $k \in \Omega$  we define a truncation function

$$\pi_k : L(\Omega) \longrightarrow \Omega$$

as follows. Given any tree  $a \in L(\Omega)$  we match up the nodes in  $k$  with the nodes in  $a$ , recursively:

1. The root nodes are matched.
2. The children of a matched node in  $k$  are matched to children of the corresponding node in  $a$  from the left. Recall that each node in  $a$  has the maximum possible arity, and that  $a$  has the maximum possible depth. Note that the arity of nodes in  $a$  will be reduced (if necessary) to that of the matching nodes in  $k$ .

This procedure corresponds to matching by co-ordinates. The effect of the operator  $\pi_k$  on a tree  $a \in L(\Omega)$  is to throw away all nodes that are not matched against nodes in

<sup>3</sup>Subscripts will be dropped whenever it is possible to infer the arity of a primitive from the context.

<sup>4</sup>The space of 0/1 trees obtained when  $\mathcal{F} = \mathcal{F}_1$  is isomorphic to the space of binary strings of arbitrary length.

$k$ . The remaining tree  $\pi_k(a)$  will then be of the same size and shape as  $k$ .

For example, suppose the maximum depth is  $\ell = 3$  and the maximum arity is also 3. Let  $a \in L(\Omega)$  be the tree  $(1 (0 1 1 0) (1 0 1 1) (1 1 1 0))$  and let  $k = (0 (1 1 0) (0 1))$ . Then matching nodes and truncating  $a$  produces  $\pi_k(a) = (1 (0 1 1) (1 0))$ .

The group  $L(\Omega)$  acts on the elements of  $\Omega$  as follows. Let  $a \in L(\Omega)$  and  $k \in \Omega$ . Then define

$$a(k) = \pi_k(a) \oplus k$$

which means we apply addition modulo 2 on each matched pair of nodes. We have used the extended definition of  $\oplus$  since  $\pi_k(a)$  and  $k$  are guaranteed to have the same size and shape. In our previous example we would have  $a(k) = (1 (1 0 1) (1 1))$ .

We can extend the definition of  $\oplus$  further by setting

$$a \oplus k = a(k)$$

for any  $k \in \Omega$  and  $a \in L(\Omega)$ . The effect of this is essentially a relabelling of the nodes of the tree  $k$  in accordance with the pattern of ones found in  $a$ .

For each  $a \in L(\Omega)$  we define a corresponding  $r \times r$  permutation matrix  $\sigma_a$  with

$$(\sigma_a)_{i,j} = \delta((a \oplus i) = j)$$

**Lemma 1.** *Let  $m, n, y \in \Omega$  and let  $a \in L(\Omega)$ . Then for homologous crossover*

$$r_{m,n}(y) = r_{a \oplus m, a \oplus n}(a \oplus y)$$

**Proof:** Interpreting Equation 9 for 0/1 trees  $m, n$  and  $y$ , the following hold:

$$a \oplus m = a \oplus y \iff m = y$$

$$C(a \oplus m, a \oplus n) = C(m, n)$$

$$(a \oplus m) \in \Gamma(a \oplus y, l) \iff m \in \Gamma(y, l)$$

and the result follows. The third assertion follows from the fact that we are relabelling the nodes in tree  $m$  according to the pattern of ones in  $a$ , and we relabel the nodes in the hyperschema  $\Gamma(y, l)$  according to exactly the same pattern.  $\square$

Let us consider the GP schema  $G$  consisting only of “=” nodes representing the shape of some of the programs in  $\Omega$ . We denote with  $0^G$  the element of  $\Omega$  obtained by replacing the = nodes in  $G$  with 0 nodes.

**Theorem 2.** *On the space of 0/1 trees with depth at most  $\ell$  homologous crossover gives rise to a mixing operator*

$$\mathcal{M}(x) = \langle x^T M_0 x, x^T M_1 x, \dots \rangle$$

(where we are indexing vectors by the elements of  $\Omega$ ). Then for each fixed shape  $G$  of depth not bigger than  $\ell$  there exists a mixing matrix

$$M = M_0 \sigma$$

such that if  $y \in \Omega$  is of shape  $G$  then

$$M_y = \sigma_a^T M \sigma_a$$

for some  $a \in L(\Omega)$ .

**Proof:** Let  $y \in \Omega$  be of shape  $G$  as required. Construct a maximal full tree  $a$  of depth not bigger than  $\ell$  by appending a sufficient number of 0 nodes to the tree  $y$  so that each internal node in  $a$  has  $i_m$  children.<sup>5</sup>

Now suppose  $m, n \in \Omega$  are trees which cross together to form  $y$  with probability  $r_{m,n}(y)$ . Because crossover is assumed to be homologous, the set of the coordinates on the nodes in  $m$  must be a superset of the set of node coordinates of  $G$ . Likewise for  $n$ .

The  $m, n$ th component of  $\sigma_a^T M \sigma_a$  is

$$\begin{aligned} (\sigma_a^T M \sigma_a)_{m,n} &= \sum_v (\sigma_a^T M)_{m,v} (\sigma_a)_{v,n} \\ &= \sum_v \sum_w (\sigma_a)_{w,m} M_{w,v} (\sigma_a)_{v,n} \\ &= M_{a^{-1} \oplus m, a^{-1} \oplus n} \\ &= r_{a^{-1} \oplus m, a^{-1} \oplus n}(0^G) \\ &= r_{m,n}(a \oplus 0^G) \\ &= r_{m,n}(y \oplus 0^G) \\ &= r_{m,n}(y) \\ &= (M_y)_{m,n} \end{aligned}$$

where we have used the lemma to show

$$r_{a^{-1} \oplus m, a^{-1} \oplus n}(0^G) = r_{m,n}(a \oplus 0^G)$$

and  $a^{-1}$  is the inverse of the group element  $a$ . For 0/1 trees  $a^{-1} = a$  since  $a \oplus a = 0^{G_m}$ , where  $G_m$  is the schema representing the shape of the trees in  $L(\Omega)$ .  $\square$

## 5 A Linear Example

In this section we will demonstrate the application of this theory to an example. To keep the presentation of the calculations manageable in the space available this example must perforce be quite simple, but should still be sufficient to illustrate the key concepts.

For this example we will assume that the function set contains only unary functions, with the possible labels for both

<sup>5</sup>For example, if  $\ell = 3$ ,  $i_m = 3$ ,  $G$  is  $(= (= =))$  and  $y = (1 1 (1 1 1))$ , then  $a = (1 (1 0 0 0) (1 1 1 0) (0 0 0 0))$ .

functions and terminals being 0 and 1 (i.e.,  $\mathcal{F} = \mathcal{F}_1 = \mathcal{T} = \{0, 1\}$ ). As a result we can think of our structures as being variable length binary strings. We will let  $\ell = 2$  (i.e., we restrict ourselves to strings of length 1 or 2), which means that  $r = 6$  and

$$\Omega = \{0, 1, 00, 01, 10, 11\}.$$

We will also limit ourselves here to the mixing matrices for GP one-point crossover and GP uniform crossover; we could however readily extend this to any other homologous crossover operator.

### 5.1 GP one-point crossover

The key to applying this theory is to compute  $r_{m,n}(y)$  as described in Equation 9. In other words, for each  $y \in \Omega$  we need to construct a matrix  $M_y = r_{m,n}(y)$  that contains the probabilities that GP one-point crossover with parents  $m$  and  $n$  will yield  $y$ . Since  $r = |\Omega| = 6$ , this will yield six  $6 \times 6$  matrices. In the (fixed-length) GA case it would only be necessary to specify one mixing matrix, since symmetries would allow us to derive the others through permutations of the indices. As indicated in the previous section, the symmetries in 0/1 trees case are more complex, and one can not reduce the situation down to just one case. In particular we find, as mentioned above, that the set of mixing matrices for our variable-length GA case splits into two different subsets, one for  $y$  of length 1, and one for  $y$  of length 2, and the necessary permutations are generated by the group  $L(\Omega) = \{00, 01, 10, 11\}$ .

To make this more concrete, let us consider  $M_0$  and  $M_1$ , each of which has exactly one non-zero column:<sup>6</sup>

$$M_0 = \begin{array}{c|cccc} & 0 & 1 & 00 & \dots \\ \hline 0 & 1 & 0 & 0 & \dots \\ 1 & 1 & 0 & 0 & \dots \\ 00 & 1/2 & 0 & 0 & \dots \\ 01 & 1/2 & 0 & 0 & \dots \\ 10 & 1/2 & 0 & 0 & \dots \\ 11 & 1/2 & 0 & 0 & \dots \end{array}$$

$$M_1 = \begin{array}{c|cccc} & 0 & 1 & 00 & \dots \\ \hline 0 & 0 & 1 & 0 & \dots \\ 1 & 0 & 1 & 0 & \dots \\ 00 & 0 & 1/2 & 0 & \dots \\ 01 & 0 & 1/2 & 0 & \dots \\ 10 & 0 & 1/2 & 0 & \dots \\ 11 & 0 & 1/2 & 0 & \dots \end{array}$$

<sup>6</sup>Since these matrices are indexed by variable length binary strings instead of natural numbers, we have indicated the indices (0, 1, 00, 01, 10 and 11) along the top and left-hand side of each matrix. In  $M_0$ , for example, the value in position (1, 0) is 1 and (01, 0) is 1/2.

Clearly  $M_1$  is very similar to  $M_0$ . Indeed, Theorem 2 shows that  $M_1$  can be obtained by applying a permutation matrix to  $M_0$ :

$$M_1 = \sigma_{10}^T M_0 \sigma_{10},$$

where

$$\sigma_{10}^T = \begin{array}{c|cccccc} & 0 & 1 & 00 & 01 & 10 & 11 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 00 & 0 & 0 & 0 & 0 & 1 & 0 \\ 01 & 0 & 0 & 0 & 0 & 0 & 1 \\ 10 & 0 & 0 & 1 & 0 & 0 & 0 \\ 11 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}.$$

The situation is more interesting for the mixing matrices for  $y$  of length 2:

$$M_{00} = \begin{array}{c|cccccc} & 0 & 1 & 00 & 01 & 10 & 11 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 00 & 0 & 0 & 1 & 0 & 1/2 & 0 \\ 01 & 0 & 0 & 1 & 0 & 1/2 & 0 \\ 10 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 11 & 0 & 0 & 1/2 & 0 & 0 & 0 \end{array}$$

$$M_{01} = \begin{array}{c|cccccc} & 0 & 1 & 00 & 01 & 10 & 11 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 00 & 0 & 0 & 0 & 1 & 0 & 1/2 \\ 01 & 0 & 0 & 0 & 1 & 0 & 1/2 \\ 10 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 11 & 0 & 0 & 0 & 1/2 & 0 & 0 \end{array}$$

$$M_{10} = \begin{array}{c|cccccc} & 0 & 1 & 00 & 01 & 10 & 11 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 00 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ 01 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ 10 & 0 & 0 & 1/2 & 0 & 1 & 0 \\ 11 & 0 & 0 & 1/2 & 0 & 1 & 0 \end{array}$$

$$M_{11} = \begin{array}{c|cccccc} & 0 & 1 & 00 & 01 & 10 & 11 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 00 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 01 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 10 & 0 & 0 & 0 & 1/2 & 0 & 1 \\ 11 & 0 & 0 & 0 & 1/2 & 0 & 1 \end{array}$$

Here again we can write these mixing matrices as permutations of  $M_{00}$ , i.e.,

$$M_s = \sigma_s^T M_{00} \sigma_s$$

for  $s \in \{00, 01, 10, 11\}$ .  $M_{01}$ , for example, can be written as

$$M_{01} = \sigma_{01}^T M_{00} \sigma_{01}$$

where  $\sigma_{01}$  is as above.

## 5.2 GP uniform crossover

Here will just show the mixing matrices  $M_0$  and  $M_{00}$  since, as we have seen, the other four matrices can be readily obtained from these using the permutation matrices  $\sigma_s$ :

$$M_0 = \begin{bmatrix} & 0 & 1 & 00 & 01 & 10 & 11 \\ 0 & 1 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 \\ 1 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 00 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 01 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 10 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 11 & 1/2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_{00} = \begin{bmatrix} & 0 & 1 & 00 & 01 & 10 & 11 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 00 & 1/2 & 1/2 & 1 & 1/2 & 1/2 & 1/4 \\ 01 & 0 & 0 & 1/2 & 0 & 1/4 & 0 \\ 10 & 0 & 0 & 1/2 & 1/4 & 0 & 0 \\ 11 & 0 & 0 & 1/4 & 0 & 0 & 0 \end{bmatrix}$$

Comparing these matrices to those obtained for one-point crossover one can see that these are symmetric, where those for one-point crossover were not, pointing out that uniform crossover is symmetric with respect to the parents, where one-point crossover is not. The matrices for uniform crossover also have considerably more non-zero entries than those for one-point crossover, highlighting the fact that uniform crossover provides more ways to construct any given string.

## 6 Conclusions

In this paper we have presented the first ever Markov chain model of GP and variable-length GAs. Obtaining this model has been possible thanks to very recent developments in the GP schema theory, which have given us exact formulas for computing the probability that reproduction and recombination will create any specific program in the search space. Our GP Markov chain model is then easily obtained by plugging this ingredient into a minor extension of Vose's model of GAs. This theoretical approach provides an excellent framework for studying the dynamics of evolutionary algorithms (in terms of transient and long-term behaviour). It also makes explicit the relationship between the local action of genetic operators on individuals and the global behaviour of the population.

The theory is applicable to GP and variable-length GAs with homologous crossover [25]: a set of operators where the offspring are created preserving the position of the genetic material taken from the parents. If one uses only unary functions and the population is initialised with programs having a fixed common length, a GP system using

these operators is entirely equivalent to a GA acting on fixed-length strings. For this reason, in the absence of mutation, our GP Markov chain model is a proper generalisation Vose's model of GAs. This is an indication that perhaps in the future it will be possible to completely unify the theoretical models of GAs and GP.

In the paper we analysed in detail the case of 0/1 trees (which include variable length binary strings), where symmetries can be exploited to obtain further simplifications in the model. The similarity with Vose's GA model is very clear in this case.

This paper is only a first step. In future research we intend to analyse in more depth the general case of tree-like structures to try to identify symmetries in the mixing matrices similar to those found for 0/1 trees. Also, we intend to study the characteristics of the transition matrices for the GP model, to gain insights into the dynamics of GP.

## Acknowledgements

The authors would like to thank the members of the EE-BIC (Evolutionary and Emergent Behaviour Intelligence and Computation) group at Birmingham, for useful discussions and comments. Nic would like to extend special thanks to The University of Birmingham School of Computer Science for graciously hosting him during his sabbatical, and various offices and individuals at the University of Minnesota, Morris, for making that sabbatical possible.

## References

- [1] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, USA, 1975.
- [2] Nicholas J. Radcliffe, "Schema processing", in *Handbook of Evolutionary Computation*, T. Baeck, D. B. Fogel, and Z. Michalewicz, Eds., pp. B2.5–1–10. Oxford University Press, 1997.
- [3] Allen E. Nix and Michael D. Vose, "Modeling genetic algorithms with Markov chains", *Annals of Mathematics and Artificial Intelligence*, vol. 5, pp. 79–88, 1992.
- [4] Michael D. Vose, *The simple genetic algorithm: Foundations and theory*, MIT Press, Cambridge, MA, 1999.
- [5] Thomas E. Davis and Jose C. Principe, "A Markov chain framework for the simple genetic algorithm", *Evolutionary Computation*, vol. 1, no. 3, pp. 269–288, 1993.
- [6] Günter Rudolph, "Stochastic processes", in *Handbook of Evolutionary Computation*, T. Baeck, D. B. Fogel, and Z. Michalewicz, Eds., pp. B2.2–1–8. Oxford University Press, 1997.
- [7] Günter Rudolph, "Genetic algorithms", in *Handbook of Evolutionary Computation*, T. Baeck, D. B. Fogel, and Z. Michalewicz, Eds., pp. B2.4–20–27. Oxford University Press, 1997.

- [8] Günter Rudolph, "Convergence analysis of canonical genetic algorithm", *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.
- [9] Günter Rudolph, "Models of stochastic convergence", in *Handbook of Evolutionary Computation*, T. Baeck, D. B. Fogel, and Z. Michalewicz, Eds., pp. B2.3–1–3. Oxford University Press, 1997.
- [10] Jonathan E. Rowe, "Population fixed-points for functions of unitation", in *Foundations of Genetic Algorithms 5*, Wolfgang Banzhaf and Colin Reeves, Eds. 1999, pp. 69–84, Morgan Kaufmann.
- [11] William M. Spears, "Aggregating models of evolutionary algorithms", in *Proceedings of the Congress on Evolutionary Computation*, Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, Eds., Mayflower Hotel, Washington D.C., USA, 6-9 July 1999, vol. 1, pp. 631–638, IEEE Press.
- [12] John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [13] Lee Altenberg, "Emergent phenomena in genetic programming", in *Evolutionary Programming — Proceedings of the Third Annual Conference*, A. V. Sebald and L. J. Fogel, Eds. 1994, pp. 233–241, World Scientific Publishing.
- [14] Una-May O'Reilly and Franz Oppacher, "The troubling aspects of a building block hypothesis for genetic programming", in *Foundations of Genetic Algorithms 3*, L. Darrell Whitley and Michael D. Vose, Eds., Estes Park, Colorado, USA, 31 July–2 Aug. 1994 1995, pp. 73–88, Morgan Kaufmann.
- [15] P. A. Whigham, "A schema theorem for context-free grammars", in *1995 IEEE Conference on Evolutionary Computation*, Perth, Australia, 29 Nov. - 1 Dec. 1995, vol. 1, pp. 178–181, IEEE Press.
- [16] Riccardo Poli and W. B. Langdon, "A new schema theory for genetic programming with one-point crossover and point mutation", in *Genetic Programming 1997: Proceedings of the Second Annual Conference*, John R. Koza, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, Eds., Stanford University, CA, USA, 13-16 July 1997, pp. 278–285, Morgan Kaufmann.
- [17] Justinian P. Rosca, "Analysis of complexity drift in genetic programming", in *Genetic Programming 1997: Proceedings of the Second Annual Conference*, John R. Koza, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, Eds., Stanford University, CA, USA, 13-16 July 1997, pp. 286–294, Morgan Kaufmann.
- [18] Riccardo Poli and William B. Langdon, "Schema theory for genetic programming with one-point crossover and point mutation", *Evolutionary Computation*, vol. 6, no. 3, pp. 231–252, 1998.
- [19] R. Poli, "Hyperschema theory for GP with one-point crossover, building blocks, and some new results in GA theory", in *Genetic Programming, Proceedings of EuroGP 2000*, Riccardo Poli, Wolfgang Banzhaf, and *et al.*, Eds. 15-16 Apr. 2000, Springer-Verlag.
- [20] Riccardo Poli, "Exact schema theorem and effective fitness for GP with one-point crossover", in *Proceedings of the Genetic and Evolutionary Computation Conference*, D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds., Las Vegas, July 2000, pp. 469–476, Morgan Kaufmann.
- [21] Riccardo Poli, "Exact schema theory for genetic programming and variable-length genetic algorithms with one-point crossover", *Genetic Programming and Evolvable Machines*, vol. 2, no. 2, 2001, Forthcoming.
- [22] Riccardo Poli, "General schema theory for genetic programming with subtree-swapping crossover", in *Genetic Programming, Proceedings of EuroGP 2001*, Milan, 18-20 Apr. 2001, LNCS, Springer-Verlag.
- [23] Riccardo Poli and Nicholas F. McPhee, "Exact schema theorems for GP with one-point and standard crossover operating on linear structures and their application to the study of the evolution of size", in *Genetic Programming, Proceedings of EuroGP 2001*, Milan, 18-20 Apr. 2001, LNCS, Springer-Verlag.
- [24] Nicholas F. McPhee and Riccardo Poli, "A schema theory analysis of the evolution of size in genetic programming with linear representations", in *Genetic Programming, Proceedings of EuroGP 2001*, Milan, 18-20 Apr. 2001, LNCS, Springer-Verlag.
- [25] Riccardo Poli and Nicholas F. McPhee, "Exact schema theory for GP and variable-length GAs with homologous crossover", in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, San Francisco, California, USA, 7-11 July 2001, Morgan Kaufmann.
- [26] Riccardo Poli and Nicholas Freitag McPhee, "Exact GP schema theory for headless chicken crossover and subtree mutation", in *Proceedings of the 2001 Congress on Evolutionary Computation CEC 2001*, Seoul, Korea, May 2001.
- [27] Nicholas F. McPhee, Riccardo Poli, and Jon 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 CEC 2001*, Seoul, Korea, May 2001.
- [28] Riccardo Poli and William B. Langdon, "On the search properties of different crossover operators in genetic programming", in *Genetic Programming 1998: Proceedings of the Third Annual Conference*, John R. Koza, Wolfgang Banzhaf, Kumar Chellappilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick Riolo, Eds., University of Wisconsin, Madison, Wisconsin, USA, 22-25 July 1998, pp. 293–301, Morgan Kaufmann.
- [29] Michael D. Vose and Gunar E. Liepins, "Punctuated equilibria in genetic search", *Complex Systems*, vol. 5, no. 1, pp. 31, 1991.
- [30] Melanie Mitchell, *An introduction to genetic algorithms*, Cambridge MA: MIT Press, 1996.
- [31] Murray R. Spiegel, *Probability and Statistics*, McGraw-Hill, New York, 1975.
- [32] Jonathan E. Rowe, Michael D. Vose, and Alden H. Wright, "Group properties of crossover and mutation", Manuscript submitted for publication, 2001.