Exact Schema Theory 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

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 new exact schema theory for genetic programming and variable-length genetic algorithms which is applicable to the general class of homologous crossovers. These are a group of operators, including GP one-point crossover and GP uniform crossover, where the offspring are created preserving the position of the genetic material taken from the parents. The theory is based on the concepts of GP crossover masks and GP recombination distributions (both introduced here for the first time), as well as the notions of hyperschema and node reference systems introduced in other recent research. This theory generalises and refines previous work in GP and GA theory.

1 Introduction

Genetic programming theory has had a difficult childhood. After some excellent early efforts leading to different approximate schema theorems [1, 2, 3, 4, 5, 6, 7], only very recently have schema theories become available which give exact formulations (rather than lower bounds) for the expected number of instances of a schema at the next generation. These exact theories are applicable to GP with one-point crossover [8, 9, 10], standard crossover and other subtree-swapping crossovers [11, 12, 13], and different types of subtree mutation and headless chicken crossover [14, 15].

Here we extend this work by presenting a new exact schema theory for genetic programming which is applicable to a very important and general class of operators which we call homologous crossovers. This group of operators generalises most common GA crossovers and includes GP one-point crossover and GP uniform crossover [16]. These operators differ from the standard subtree swapping crossover [1] in that they require that the offspring being created preserve the position of the genetic material taken from the parents.

The paper is organised as follows. Firstly, we provide a review of earlier relevant work on GP schemata and cover the key definitions and terms in Section 2. Then, in Section 3 we show how these ideas can be used to define the class of homologous crossover operators and build probabilistic models for them. In Section 4 we use these to derive schema theory results and an exact definition of effective fitness for GP with homologous crossover. In Section 5 we give an example that shows how the theory can be applied. Some conclusions are drawn in Section 6.

2 Background

Schemata are sets of points of the search space sharing some syntactic feature. For example, in the context of GAs operating on binary strings, the syntactic representation of a schema is usually a string of symbols from the alphabet {0,1,*}, where the character * is interpreted as a "don't care" symbol. Typically schema theorems are descriptions of how the number of members of the population belonging to a schema vary over time. Let $\alpha(H,t)$ denote the probability at time t that a newly created individual samples (or matches) the schema H, which we term the *total transmission probability* of H. Then an exact schema theorem for a generational system is simply [17]

$$E[m(H,t+1)] = M\alpha(H,t), \qquad (1)$$

where M is the population size, $m(H,t+1)$ is the number of individuals sampling H at generation $t+1$ and $E[\cdot]$ is the expectation operator. Holland's [18] and other worst-case-scenario schema theories normally provide a lower bound for $\alpha(H,t)$ or, equivalently, for $E[m(H,t+1)]$.

One of the difficulties in obtaining theoretical results on GP using the idea of schema is that finding a workable definition of a schema is much less straightforward than for GAs. Several alternative definitions have been proposed in the literature [1, 2, 3, 4, 6, 7, 5]. For brevity here we will describe only the definition introduced in [6, 7], since this is what is used in the rest of this paper. We will refer to this kind of schemata as *fixed-size-and-shape schemata*.

Syntactically a GP fixed-size-and-shape schema is 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 terminal sets used in a GP run. The primitive $=$ is a "don't care" symbol which stands for a *single* terminal or function. A schema H represents the set of all programs having the same shape as H and the same labels for the non-$=$ nodes. For example, if $\mathcal{F}$={`+, *`} and $\mathcal{T}$={`x, y`} the schema `(+ x (= y =))` represents the four programs `(+ x (+ y x))`, `(+ x (+ y y))`, `(+ x (* y x))` and `(+ x (* y y))`.

In [6, 7] a worst-case-scenario schema theorem was derived for GP with point mutation and one-point crossover; as discussed in [8], this theorem is a generalisation of the version of Holland's schema theorem [18] presented in [19] to variable size structures. One-point crossover works by using the same crossover point in both parent programs, and then swapping the corresponding subtrees like standard crossover. To account for the possible structural diversity of the two parents, the selection of the crossover point is restricted to the *common region*, the largest rooted region where the two parent trees have the same topology. The common region will be defined formally in Section 3.

One-point crossover can be considered to be an instance of a much broader class of operators that can be defined through the notion of the common region. For example, in [16] we defined and studied a GP operator, called *uniform crossover* (based on uniform crossover in GAs), in which the offspring is created by independently swapping the nodes in the common region with a uniform probability. If a node belongs to the boundary of the common region and is a function then also the nodes below it are swapped, otherwise only the node label is swapped. Many other operators of this kind are possible. We will call them *homologous crossovers*, noting that our definition is more restrictive than that in [20]. A formal description of these operators will be given in Section 3.

The approximate schema theorem in [6, 7] was improved in [9, 10], where an exact schema theory for GP with one-point crossover was derived which was based on the notion of hyperschema. 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. For example, the hyperschema `(* # (= x =))` represents all the programs with the following characteristics: a) the root node is a product, b) the first argument of the root node is any valid subtree, c) the second argument of the root node is any function of arity two, d) the first argument of this function is the variable `x`, e) the second argument of the function is any valid node in the terminal set. One of the results obtained in [10] is

$$\alpha(H,t) = (1 - p_{xo})p(H,t) + p_{xo}\alpha_{xo}(H,t) \tag{2}$$

where

$$\alpha_{xo}(H,t) = \sum_{k,l} \frac{1}{\mathbf{NC}(G_k, G_l)} \times \sum_{i \in C(G_k,G_l)} p(U(H,i) \cap G_k, t) p(L(H,i) \cap G_l, t) \tag{3}$$

and: p_{xo} is the crossover probability; $p(H,t)$ is the selection probability of the schema H;[1] $G_1, G_2, \cdots$ are an enumeration of all the possible program shapes, i.e. all the possible fixed-size-and-shape schemata containing = signs only; $\mathbf{NC}(G_k, G_l)$ is the number of nodes in the common region between shape G_k and shape G_l; $C(G_k, G_l)$ is the set of indices of the crossover points in such a common region; $L(H,i)$ is the hyperschema obtained by replacing all the nodes on the path between crossover point i and the root node with = nodes, and all the subtrees connected to those nodes with # nodes; $U(H,i)$ is the hyperschema obtained by replacing the subtree below crossover point i with a # node; if a crossover point i is in the common region between two programs but it is outside the schema H, then $L(H,i)$ and $U(H,i)$ are defined to be the empty set. The hyperschemata $L(H,i)$ and $U(H,i)$ are important because, if one crosses over at point i *any* individual in $L(H,i)$ with *any* individual in $U(H,i)$, the resulting offspring is always an instance of H. The steps involved in the construction of $L(H,i)$ and $U(H,i)$ for the schema H =`(* = (+ x =))` are illustrated in Figure 1.

As discussed in [8], it is possible to show that, in the absence of mutation, Equations 2 and 3 generalise and refine not only the GP schema theorem in [6, 7] but also the version of Holland's schema theorem [18] presented in [19], as well as more recent GA schema theory [21, 22].

Very recently, this work has been extended in [11] where a general, exact schema theory for genetic programming with subtree swapping crossover was presented. The theory is based on a generalisation of the notion of hyperschema and on a Cartesian node reference system which makes it possible to describe programs as functions over the space $\mathbb{N}^2$.

The Cartesian reference system is obtained by considering the ideal infinite tree consisting entirely of nodes of some fixed maximum arity a_{max}. This maximal tree would include 1 node of arity a_{max} at depth 0, a_{max} nodes of arity a_{max} at depth 1, $(a_{max})^2$ nodes of arity a_{max} at depth 2, and

[1] In fitness proportionate selection $p(H,t) = m(H,t)f(H,t)/(M\bar{f}(t))$, where $m(H,t)$ is the number of trees in the schema H at time t, $f(H,t)$ is their mean fitness, and $\bar{f}(t)$ is the mean fitness of the trees in the population.

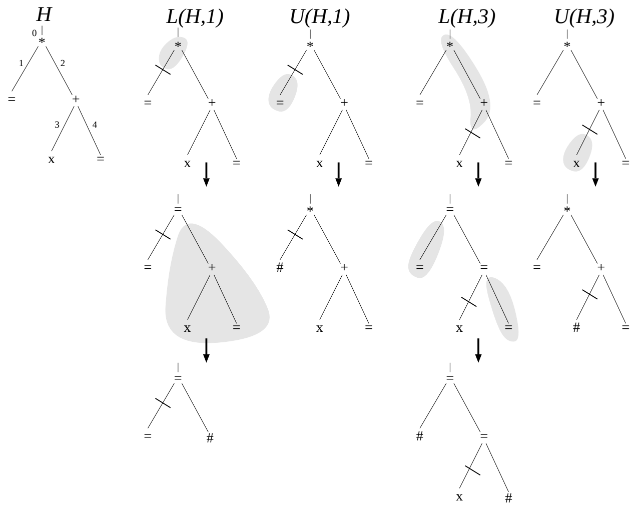

Figure 1: Example of a schema and some of its potential hyperschema building blocks. The crossover points in H are numbered as shown in the top left.

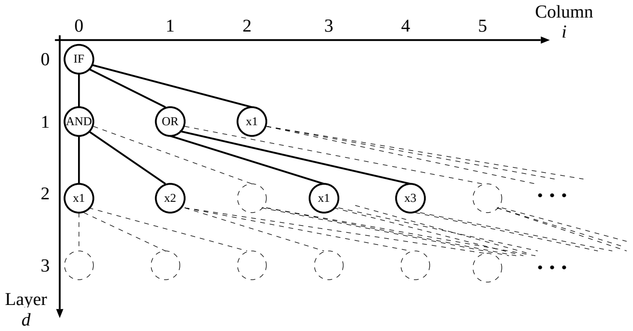

Figure 2: Syntax tree for the program `(IF (AND x1 x2) (OR x1 x3) x1)` represented in a tree-independent Cartesian node reference system for nodes with maximum arity 3. Unused nodes and links of the maximal tree are drawn with dashed lines. Only four layers and six columns are shown.

generally $(a_{max})^d$ nodes at depth d. Then one could imagine organising the nodes in the tree into layers of increasing depth (see Figure 2) and assigning an index to each node in a layer. The layer number d and the index i can then be used to define a Cartesian coordinate system. Clearly, one could also use this reference system to locate the nodes of non-maximal trees. This is possible because a non-maximal tree can always be described using a subset of the nodes and links in the maximal tree. This is illustrated for the program `(IF (AND x1 x2) (OR x1 x3) x1)` in Figure 2. So, for example, the `IF` node would have coordinates (0,0), the `AND` would have coordinates (1,0), and the `x3` node would have coordinates (2,4). In this reference system it is always possible to find the route to the root node from any valid coordinate. Also, if one chooses a_{max} to be the maximum arity of the functions in the function set, it is possible to use this reference system to represent the structure of any program that can be constructed with that function set.

The theory in [11] is also applicable to standard GP crossover [1] with and without uniform selection of the crossover points, one-point crossover [6, 7], size-fair crossover [20], strongly-typed GP crossover [23], context-preserving crossover [24], and many others. The theory has also been recently extended to subtree mutation and headless chicken crossover [14, 15]. It does not, however, currently cover the class of homologous operators and the goal of this paper is to fill that theoretical gap.

3 Modelling Homologous Crossovers

Given a node reference system it is possible to define functions over it. An example of such functions is the *arity function* $A(d,i,h)$ which returns the arity of the node at coordinates (d,i) in h. For example, for the tree in Figure 2, $A(0,0,h)=3$, $A(1,0,h)=2$ and $A(2,1,h)=0$. Similarly, it is possible to define the *common region membership function* $\mathcal{C}(d,i,h_1,h_2)$ which returns **true** when (d,i) is part of the common region of h_1 and h_2. Formally, $\mathcal{C}(d,i,h_1,h_2)=\mathbf{true}$ when either $(d,i)=(0,0)$ or

$$\begin{aligned} A\left(d-1,i',h_1\right) = A\left(d-1,i',h_2\right) \neq 0 \\ \mathbf{and} \quad \mathcal{C}\left(d-1,i',h_1,h_2\right) = \mathbf{true}, \end{aligned}$$

where $i' = \lfloor i/a_{max} \rfloor$ and $\lfloor\cdot\rfloor$ is the integer-part function. This allows us to formalise the notion of *common region*:

$$C(h_1,h_2) = \{(d,i) \mid \mathcal{C}(d,i,h_1,h_2) = \mathbf{true}\}. \qquad (4)$$

This is the notion of common region used in the schema theorem for one-point crossover in Equation 2. As indicated before, one-point crossover selects the same crossover point in both parents by randomly choosing a node in the common region. An alternative way to interpret the action of one-point crossover is to imagine that the subset of nodes in $C(h_1,h_2)$ below such a crossover point are transferred from parent h_2 into an empty coordinate system, while all the remaining nodes in $C(h_1,h_2)$ are taken from parent h_1. Clearly, nodes representing the leaves of the common region should be transferred together with their subtrees, if any. Other homologous crossovers can simply be defined by selecting subsets of nodes in the common region differently.

A good way to describe and model the class of homologous crossovers is to extend the notions of crossover masks and recombination distributions used in genetics [25] and in the GA literature [26, 27, 28]. In a GA operating on fixed-length strings a crossover mask is simply a binary string. When crossover is executed, the bits of the offspring corresponding to the 1's in the mask will be taken from one parent, those corresponding to 0's from the other parent. For example, if the parents are the strings `aaaaaa` and `bbbbbb` and the crossover mask is `110100`, one offspring would be `aababb`. For operators returning two offspring it is easy to show that the second offspring can be obtained by simply complementing, bit by bit, the crossover mask. For example, the complement of the mask `110100`, `001011`, gives the offspring `bbabaa`. If the GA operates on strings of length N, then 2^N different crossover masks are possible. If, for each mask i, one defines a probability, p_i, that the mask is selected for crossover, then it is easy to see how different crossover operators can simply be interpreted as different ways of choosing the probability distribution p_i. For example, for strings of length $N = 4$ the probability distribution for one-point crossover would be $p_i = 1/3$ for the crossover masks $i = 1000, 1100, 1110$ and $p_i = 0$ otherwise, while for uniform crossover $p_i = 1/16$ for all 16 i's. The probability distribution p_i is called a *recombination distribution*.

Let us now extend the notion of recombination distributions to genetic programming with homologous crossover. For any given shape and size of the common region we can define a set of *GP crossover masks* which correspond to all possible ways in which a recombination event can take place within the given common region. Because the nodes in the common region are always arranged so as to form a tree, it is possible to represent the common region as a tree or an equivalent S-expression. So, GP crossover masks can be thought of as trees constructed using 0's and 1's that have the same size and shape as the common region. So, for example, if the common region is represented by the set of node coordinates {(0,0),(1,0),(1,1)}, then there are eight valid GP crossover masks: `(0 0 0)`, `(0 0 1)`, `(0 1 0)`, `(0 1 1)`, `(1 0 0)`, `(1 0 1)`, `(1 1 0)` and `(1 1 1)`. The complement of a GP crossover mask is an obvious extension, where the complement $\bar{i}$ has the same structure as mask i but with the 0's and 1's swapped. In the following we will use χ_c to denote the set of the $2^{N(c)}$ crossover masks associated with the common region c, where $N(c)$ is the number of nodes in c. Since we are typically interested in the common region defined by two trees, we'll use $\chi(h_1, h_2)$ as a shorthand for $\chi_{C(h_1,h_2)}$.

Once χ_c is defined we can define a *fixed-size-and-shape recombination distribution* p_i^c which gives the probability that crossover mask $i \in \chi_c$ will be chosen for crossover between individuals having common region c. Then the set $\{p_i^c \mid \forall c\}$, which we call a *GP recombination distribution*, completely defines the behaviour of a GP homologous crossover operator, different operators being characterised by different assignments for the p_i^c. For example, the GP recombination distribution for uniform GP crossover with 50% probability of exchanging nodes is $p_i^c = (0.5)^{N(c)}$.

GP crossover masks and GP recombination distributions generalise the corresponding GA notions. Indeed, as also discussed in [8], GAs operating on fixed-length strings are simply a special case of GP with homologous crossover. This can be shown by considering the case of function sets including only unary functions and initialising the population with programs of the same length. Since in a linear GP system with fixed length programs every individual has exactly the same size and (linear) shape, only one common region c is possible. Therefore, only one fixed-size-and-shape recombination distribution p_i^c is required to characterise crossover. In variable length GAs and GP, multiple fixed-size-and-shape recombination distributions are necessary, one for every possible common region c.

4 Exact GP Schema Theory for Homologous Crossovers

Using hyperschemata and GP recombination distributions for homologous crossover, we obtain the following:

Theorem 1. *The total transmission probability for a fixed-size-and-shape GP schema H under homologous crossover is given by Equation 2 with*

$$\alpha_{xo}(H,t) = \sum_{h_1}\sum_{h_2} p(h_1,t)p(h_2,t) \sum_{i\in\chi(h_1,h_2)} p_i^{C(h_1,h_2)} \times \delta(h_1 \in \Gamma(H,i))\delta(h_2 \in \Gamma(H,\bar{i})) \tag{5}$$

where: the first two summations are over all the individuals in the population; $C(h_1, h_2)$ is the common region between program h_1 and program h_2; $\chi(h_1, h_2)$ is the set of crossover masks associated with $C(h_1, h_2)$; $\delta(x)$ is a function which returns 1 if x is true, 0 otherwise; $\Gamma(H, i)$ is defined below; $\bar{i}$ is the complement of crossover mask i.

$\Gamma(H, i)$ is defined to be the empty set if i 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 i 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 i that is labelled with a 0, then it is replaced with a #.
- All other nodes in H are left unchanged.

If, for example, H =`(* = (+ x =))`, as indicated in Figure 3(a), then $\Gamma(H, (0\ 1\ 0))$ is obtained by first replacing the root node with a = symbol (because the crossover mask has a function node 0 at coordinates (0,0)) and then replacing the subtree rooted at coordinates (1,1) with a # symbol (because the crossover mask has a terminal node 0 at coordinates (1,1)) obtaining `(= = #)`. The schema $\Gamma(H, (1\ 0\ 1))$, which forms a complementary pair with the previous one, is instead obtained by replacing the subtree rooted at coordinates (1,0) with a # symbol obtaining `(* # (+ x =))`, as illustrated in Figure 3(b).

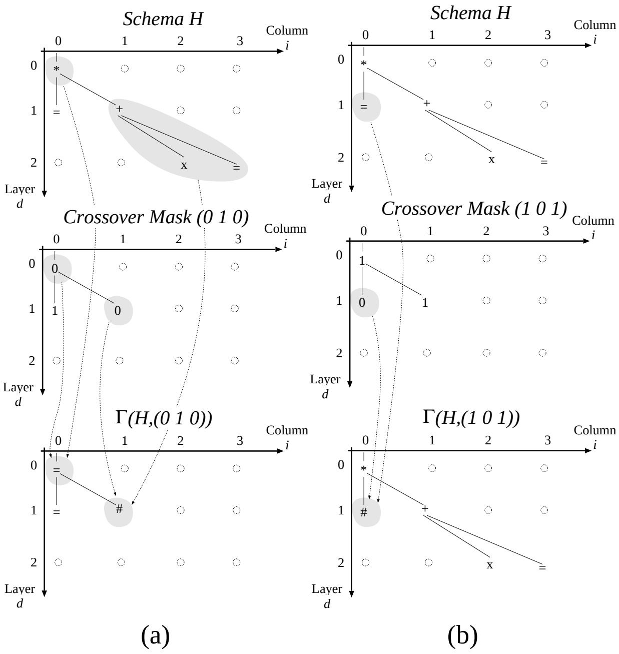

Figure 3: A complementary pair of hyperschemata $\Gamma(H,i)$ for the schema $H =$`(* = (+ x =))`.

The hyperschemata $\Gamma(H,i)$ and $\Gamma(H,\bar{i})$ are generalisations of the schemata $L(H,i)$ and $U(H,i)$ used in Equation 2 (compare Figures 1 and 3). In general if one crosses over using crossover mask i *any* individual in $\Gamma(H,i)$ with *any* individual in $\Gamma(H,\bar{i})$, the resulting offspring is always an instance of H.

Once the concept of $\Gamma(H,i)$ is available, the theorem can easily be proven.

Proof. Let $p(h_1,h_2,i,t)$ be the probability that, at generation t, the selection-crossover process will choose parents h_1 and h_2 and crossover mask i. Then, let us consider the function

$$g(h_1,h_2,i,H) = \delta(h_1 \in \Gamma(H,i))\delta(h_2 \in \Gamma(H,\bar{i})).$$

Given two parent programs, h_1 and h_2, and a schema of interest H, this function returns the value 1 if crossing over h_1 and h_2 with crossover mask i yields an offspring in H. It returns 0 otherwise. This function can be considered as a measurement function (see [27]) that we want to apply to the probability distribution of parents and crossover masks at time t, $p(h_1,h_2,i,t)$. If h_1, h_2 and i are stochastic variables with joint probability distribution $p(h_1,h_2,i,t)$, the function $g(h_1,h_2,i,H)$ can be used to define a stochastic variable $\gamma = g(h_1,h_2,i,H)$. The expected value of γ is:

$$E[\gamma] = \sum_{h_1}\sum_{h_2}\sum_{i} g(h_1,h_2,i,H)p(h_1,h_2,i,t). \qquad (6)$$

Since γ is a binary stochastic variable, its expected value also represents the proportion of times it takes the value 1. This corresponds to the proportion of times the offspring of h_1 and h_2 are in H.

We can write

$$p(h_1,h_2,i,t) = p(i|h_1,h_2)p(h_1,t)p(h_2,t),$$

where $p(i|h_1,h_2)$ is the conditional probability that crossover mask i will be selected when the parents are h_1 and h_2, while $p(h_1,t)$ and $p(h_2,t)$ are the selection probabilities for the parents. In homologous crossover $p(i|h_1,h_2) = p_i^{C(h_1,h_2)}\delta(i \in \chi(h_1,h_2))$, so

$$\begin{aligned} & p(h_1,h_2,i,t) \\ &\quad = \quad p(h_1,t)p(h_2,t)p_i^{C(h_1,h_2)}\delta(i \in \chi(h_1,h_2)). \end{aligned}$$

Substituting this into Equation 6 with minor simplifications leads to the expression of α_{xo} in Equation 5. □

Equations 2 and 5 allow one to compute the exact total transmission probability of a GP schema in terms of microscopic quantities. It is possible, however, to transform this model into the following exact macroscopic model of schema propagation

Theorem 2. *The total transmission probability for a fixed-size-and-shape GP schema H under homologous crossover is given by Equation 2 with*

$$\begin{aligned} \alpha_{xo}(H,t) = \sum_j\sum_k \sum_{i\in\chi(G_j,G_k)} p_i^{C(G_j,G_k)} \times \qquad (7) \\ p(\Gamma(H,i)\cap G_j,t)p(\Gamma(H,\bar{i})\cap G_k,t). \end{aligned}$$

Proof. Let us start by considering all the possible program shapes G_1, G_2, $\cdots$. These schemata represent disjoint sets of programs. Their union represents the whole search space, so

$$\sum_j \delta(h_1 \in G_j) = 1.$$

We insert the l.h.s. of this expression and of an analogous expression for $\delta(h_2 \in G_k)$ in Equation 5 and reorder the terms obtaining:[2]

$$\begin{aligned} & \alpha_{xo}(H,t) \\ &\quad = \sum_j\sum_k\sum_{h_1}\sum_{h_2} p(h_1,t)p(h_2,t) \\ &\qquad \sum_{i\in\chi(h_1,h_2)} p_i^{C(h_1,h_2)}\delta(h_1 \in \Gamma(H,i))\delta(h_1 \in G_j) \\ &\qquad \delta(h_2 \in \Gamma(H,\bar{i}))\delta(h_2 \in G_k) \\ &\quad = \sum_j\sum_k\sum_{h_1\in G_j}\sum_{h_2\in G_k} p(h_1,t)p(h_2,t) \\ &\qquad \sum_{i\in\chi(h_1,h_2)} p_i^{C(h_1,h_2)}\delta(h_1 \in \Gamma(H,i))\delta(h_2 \in \Gamma(H,\bar{i})) \end{aligned}$$

[2] Note that $h_1 \in G_j \wedge h_2 \in G_k \Rightarrow C(h_1,h_2) = C(G_j,G_k)$.

$$
\begin{aligned}
&= \sum_j \sum_k \sum_{h_1 \in G_j} \sum_{h_2 \in G_k} p(h_1,t)p(h_2,t) \\
&\quad \sum_{i \in \chi(G_j,G_k)} p_i^{C(G_j,G_k)} \delta(h_1 \in \Gamma(H,i))\delta(h_2 \in \Gamma(H,\bar{i})) \\
&= \sum_j \sum_k \sum_{i \in \chi(G_j,G_k)} p_i^{C(G_j,G_k)} \sum_{h_1 \in G_j} p(h_1,t) \\
&\quad \delta(h_1 \in \Gamma(H,i)) \sum_{h_2 \in G_k} p(h_2,t)\delta(h_2 \in \Gamma(H,\bar{i})).
\end{aligned}
$$

Since $\sum_{h_1 \in G_j} p(h_1,t)\delta(h_1 \in \Gamma(H,i)) = p(\Gamma(H,i) \cap G_j,t)$ (and similarly for $p(\Gamma(H,\bar{i}) \cap G_k,t)$), this equation completes the proof of the theorem. □

This theorem is a generalisation of Equations 2 and 3. These, as indicated in Section 2, are a generalisation of a recent GA schema theorem for one-point crossover [21, 22] and a refinement (in the absence of mutation) of both the GP schema theorem in [6] and Goldberg's version [19] of Holland's schema theory [18]. The schema theorems in this paper also generalise other GA results (such as those summarised in [29]), as well as the result in [27, appendix], since they can be applied to linear schemata and even fixed-length binary strings. So, in the absence of mutation, *the schema theory in this paper generalises and refines not only earlier GP schema theorems but also old and modern GA schema theories for one- and multi-point crossover, uniform crossover and all other homologous crossovers*.

Once the value of $\alpha(H,t)$ is available, it is trivial to extend (as we did in [10, 11]) the notion of effective fitness provided in [21, 22] obtaining the following:

Corollary 3. *The effective fitness of a fixed-size-and-shape GP schema H under homologous crossover is*

$$
\begin{aligned}
f_{\text{eff}}(H,t) &= \frac{\alpha(H,t)}{p(H,t)} f(H,t) \\
&= f(H,t)\Big[1 - p_{xo}\Big(1 - \sum_{j,k} \sum_{i \in \chi(G_j,G_k)} p_i^{C(G_j,G_k)} \times \\
&\quad \frac{p(\Gamma(H,i) \cap G_j,t)p(\Gamma(H,\bar{i}) \cap G_k,t)}{p(H,t)}\Big)\Big]. \qquad (8)
\end{aligned}
$$

5 Example

Since the calculations involved in applying exact GP schema theorems can become quite lengthy, we will limit ourselves here to one extremely simple example. For applications of this and related schema theories see [12, 13, 14, 15, 30]. To make clearer the relationship between this work and our theory for one-point crossover, we will use the same example as in [10], this time using general homologous crossover operators instead of just one-point crossover.

Let us imagine that we have a function set $\{A_f, B_f, C_f, D_f, E_f\}$ including only unary functions, and the terminal set $\{A_t, B_t, C_t, D_t, E_t\}$. Since, all functions are unary, we can unambiguously represent expressions without parenthesis. In addition, since the only terminal in each expression is the rightmost node, we can remove the subscripts without generating any ambiguity. Thus, every member of the search space can be seen as a variable-length string over the alphabet $\{A, B, C, D, E\}$, and GP with homologous crossover is really a non-binary variable-length GA.

Let us now consider the schema AB=. We want to measure its total transmission probability (with $p_{xo} = 1$) under fitness proportionate selection and an arbitrary homologous crossover operator for the following population:

Population	Fitness
AB	2
BCD	2
ABC	4
ABCD	6

In order to apply Equation 7 we first need to number all the possible program shapes G_1, G_2, etc.. Let G_1 be =, G_2 be ==, G_3 be === and G_4 be ====. We do not need to consider other, larger shapes because the population does not contain any larger programs. We then need to evaluate the shape of the common regions to determine $\chi(G_j,G_k)$ for all valid values of j and k. In this case the common regions can be naturally represented using integers which represent the length of the common region. Since the length of the common region is the length of the shorter parent, we know $C(G_j,G_k) = \min(j,k)$. Then, for each common region c we need to identify the hyperschemata $\Gamma(\text{AB=},i)$ for all the meaningful crossover masks $i \in \chi_c$ and calculate $\Gamma(\text{AB=},i) \cap G_j$ for all meaningful values of j. These calculations are shown in Table 1. Using this table we can apply Equation 7, obtaining, after simplification and omitting t and the superscript c from p_i^c for brevity,

$$
\begin{aligned}
&\alpha(\text{AB=}) = \alpha_{xo}(\text{AB=}) \\
&\quad = \sum_{j,k=1}^{4} \sum_{i \in \{0,1\}^{\min(j,k)}} p_i p(\Gamma(H,i) \cap G_j)p(\Gamma(H,\bar{i}) \cap G_k) \\
&\quad = (p_0 + p_1)p(AB=)p(=) + \\
&\qquad (p_{00} + p_{11})p(AB=)p(==) + \\
&\qquad (p_{01} + p_{10})p(=B=)p(A=) + \\
&\qquad (p_{000} + p_{111})p(AB=)(p(===) + p(====)) + \\
&\qquad (p_{001} + p_{110})p(===)(p(AB=) + p(AB==)) + \\
&\qquad (p_{010} + p_{101})p(A==)(p(=B=) + p(=B==)) + \\
&\qquad (p_{011} + p_{100})p(=B=)(p(A==) + p(A===)).
\end{aligned}
$$

This equation is valid for any homologous crossover operator, each of which is defined by the set of p_i. It is easy to specialises it for one-point crossover by using the

Mask	$\Gamma(\mathsf{AB=}, i)$	$\Gamma(\mathsf{AB=}, i) \cap G_j$			
i		$j = 1$	$j = 2$	$j = 3$	$j = 4$
0	#	=	==	===	====
1	AB=	∅	∅	AB=	∅
00	=#	∅	==	===	====
01	=B=	∅	∅	=B=	∅
10	A#	∅	A=	A==	A===
11	AB=	∅	∅	AB=	∅
000	==#	∅	∅	===	====
001	===	∅	∅	===	∅
010	=B#	∅	∅	=B=	=B==
011	=B=	∅	∅	=B=	∅
100	A=#	∅	∅	A==	A===
101	A==	∅	∅	A==	∅
110	AB#	∅	∅	AB=	AB==
111	AB=	∅	∅	AB=	∅
0000	∅	∅	∅	∅	∅
⋮	⋮	⋮	⋮	⋮	⋮

Table 1: Crossover masks and schemata necessary to calculate $\alpha_{xo}(\texttt{AB}\ =)$.

recombination distribution $p_0 = 1$, $p_{00} = p_{10} = 1/2$, $p_{000} = p_{100} = p_{110} = 1/3$ and $p_i = 0$ for all other crossover masks. This leads to the same result as in [10].

It is also easy to specialise the previous equation to uniform crossover by using the recombination distribution $p_i = (0.5)^{N(i)}$, where $N(i)$ is the length of crossover mask i. Doing so in this case yields $\alpha(\mathsf{AB=}, t) \approx 0.2806$. For the same example, in [10] we obtained $\alpha(\mathsf{AB=}, t) \approx 0.2925$ for one-point crossover, which indicates that uniform crossover is slightly less "friendly" towards the schema. We can also use Equation 8 to compute the effective fitness for the schema AB= for both uniform and one-point crossover, obtaining values of approximately 3.9 and 4.1, respectively. These values are very close to the actual average fitness of the schema in the current population, 4, suggesting that in this case disruption and creation effects tend to balance out. This is not always the case, however, as is shown in [10].

6 Conclusions

Unlike GA theory, which has made considerable progress in the last ten years or so, GP theory has typically been scarce, approximate and, as a rule, not terribly useful. This is not surprising given the youth of GP and the complexities of building theories for variable size structures. In the last year or so, however, significant breakthroughs have changed this situation radically. Today not only do we have exact schema theorems for GP with a variety of operators including subtree mutation, headless chicken crossover, standard crossover, one-point crossover, and all other subtree swapping crossovers, but this GP theory also generalises and refines a broad spectrum of GA theory, as indicated in Section 2.

We believe that this paper extends this series of breakthroughs. Here we have presented a new schema theory applicable to genetic programming and both variable- and fixed-length genetic algorithms with homologous crossover. The theory is based on the concepts of GP crossover masks and GP recombination distributions, both introduced here for the first time. As discussed in Section 4, this theory also generalises and refines a broad spectrum of previous work in GP and GA theory.

Clearly this paper is only a first step. We have not yet made any attempt to use our new schema evolution equations to understand the dynamics of GP or variable-length GAs with homologous crossover or to design competent GP/GA systems. In other recent work, however, we have specialised and applied the theory for other operators to understand phenomena such as operator biases and the evolution of size in variable length GAs [12, 13, 14, 15]. In the future we hope to be able to do the same and produce exciting new results with the theory presented here.

Acknowledgements

The authors would like to thank the members of the EEBIC (Evolutionary and Emergent Behaviour Intelligence and Computation) group at Birmingham, for useful discussions and comments. Nic 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. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[2] L. Altenberg, "Emergent phenomena in genetic programming," in *Evolutionary Programming — Proceedings of the Third Annual Conference* (A. V. Sebald and L. J. Fogel, eds.), pp. 233–241, World Scientific Publishing, 1994.

[3] U.-M. O'Reilly and F. Oppacher, "The troubling aspects of a building block hypothesis for genetic programming," in *Foundations of Genetic Algorithms 3* (L. D. Whitley and M. D. Vose, eds.), (Estes Park, Colorado, USA), pp. 73–88, Morgan Kaufmann, 31 July–2 Aug. 1994 1995.

[4] P. A. Whigham, "A schema theorem for context-free grammars," in *1995 IEEE Conference on Evolutionary Computation*, vol. 1, (Perth, Australia), pp. 178–181, IEEE Press, 29 Nov. - 1 Dec. 1995.

[5] J. P. Rosca, "Analysis of complexity drift in genetic programming," in *Genetic Programming 1997: Proceedings of the Second Annual Conference* (J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, eds.), (Stanford University, CA, USA), pp. 286–294, Morgan Kaufmann, 13-16 July 1997.

[6] R. 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* (J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, eds.), (Stanford University, CA, USA), pp. 278–285, Morgan Kaufmann, 13-16 July 1997.

[7] R. Poli and W. B. Langdon, “Schema theory for genetic programming with one-point crossover and point mutation,” *Evolutionary Computation*, vol. 6, no. 3, pp. 231–252, 1998.

[8] R. 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.

[9] 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* (R. Poli, W. Banzhaf, and *et al.*, eds.), Springer-Verlag, 15-16 Apr. 2000.

[10] R. 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), pp. 469–476, Morgan Kaufmann, July 2000.

[11] R. Poli, “General schema theory for genetic programming with subtree-swapping crossover,” in *Genetic Programming, Proceedings of EuroGP 2001*, LNCS, (Milan), Springer-Verlag, 18-20 Apr. 2001.

[12] R. Poli and N. 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*, LNCS, (Milan), Springer-Verlag, 18-20 Apr. 2001.

[13] N. F. McPhee and R. Poli, “A schema theory analysis of the evolution of size in genetic programming with linear representations,” in *Genetic Programming, Proceedings of EuroGP 2001*, LNCS, (Milan), Springer-Verlag, 18-20 Apr. 2001.

[14] R. Poli and N. F. 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.

[15] N. F. McPhee, R. Poli, and J. E. Rowe, “A schema theory analysis of mutation size biases in genetic programming with linear representations,” in *Proceedings of the 2001 Congress on Evolutionary Computation CEC 2001*, (Seoul, Korea), May 2001.

[16] R. Poli and W. B. Langdon, “On the search properties of different crossover operators in genetic programming,” in *Genetic Programming 1998: Proceedings of the Third Annual Conference* (J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, eds.), (University of Wisconsin, Madison, Wisconsin, USA), pp. 293–301, Morgan Kaufmann, 22-25 July 1998.

[17] R. Poli, W. B. Langdon, and U.-M. O’Reilly, “Analysis of schema variance and short term extinction likelihoods,” in *Genetic Programming 1998: Proceedings of the Third Annual Conference* (J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, eds.), (University of Wisconsin, Madison, Wisconsin, USA), pp. 284–292, Morgan Kaufmann, 22-25 July 1998.

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

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

[20] W. B. Langdon, “Size fair and homologous tree genetic programming crossovers,” *Genetic Programming And Evolvable Machines*, vol. 1, pp. 95–119, Apr. 2000.

[21] C. R. Stephens and H. Waelbroeck, “Effective degrees of freedom in genetic algorithms and the block hypothesis,” in *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)* (T. Bäck, ed.), (East Lansing), pp. 34–40, Morgan Kaufmann, 1997.

[22] C. R. Stephens and H. Waelbroeck, “Schemata evolution and building blocks,” *Evolutionary Computation*, vol. 7, no. 2, pp. 109–124, 1999.

[23] D. J. Montana, “Strongly typed genetic programming,” *Evolutionary Computation*, vol. 3, no. 2, pp. 199–230, 1995.

[24] P. D’haeseleer, “Context preserving crossover in genetic programming,” in *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, vol. 1, (Orlando, Florida, USA), pp. 256–261, IEEE Press, 27-29 June 1994.

[25] H. Geiringer, “On the probability theory of linkage in Mendelian heredity,” *Annals of Mathematical Statistics*, vol. 15, pp. 25–57, March 1944.

[26] L. B. Booker, “Recombination distributions for genetic algorithms,” in *FOGA-92, Foundations of Genetic Algorithms*, (Vail, Colorado), 24–29 July 1992. Email: booker@mitre.org.

[27] L. Altenberg, “The Schema Theorem and Price’s Theorem,” in *Foundations of Genetic Algorithms 3* (L. D. Whitley and M. D. Vose, eds.), (Estes Park, Colorado, USA), pp. 23–49, Morgan Kaufmann, 31 July–2 Aug. 1994 1995.

[28] W. M. Spears, “Limiting distributions for mutation and recombination,” in *Proceedings of the Foundations of Genetic Algorithms Workshop (FOGA 6)* (W. M. Spears and W. Martin, eds.), (Charlottesville, VA, USA), July 2000. In press.

[29] D. Whitley, “A genetic algorithm tutorial,” Tech. Rep. CS-93-103, Department of Computer Science, Colorado State University, Aug. 1993.

[30] R. Poli, J. E. Rowe, and N. F. McPhee, “Markov chain models for GP and variable-length GAs with homologous crossover,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, (San Francisco, California, USA), Morgan Kaufmann, 7-11 July 2001.