

# A schema theory analysis of mutation size biases in genetic programming with linear representations

**Nicholas Freitag McPhee**

Division of Science and Mathematics  
University of Minnesota, Morris  
Morris, MN, USA, 56267  
mcphee@mrs.umn.edu

**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

**Abstract- Understanding operator bias in evolutionary computation is important because it is possible for the operator’s biases to work against the intended biases induced by the fitness function. In recent work we showed how developments in GP schema theory can be used to better understand the biases induced by the standard subtree crossover when genetic programming is applied to variable length linear structures. In this paper we use the schema theory to better understand the biases induced on linear structures by two common GP subtree mutation operators: FULL and GROW mutation. In both cases we find that the operators do have quite specific biases and typically strongly oversample shorter strings.**

## 1 Introduction

Most (if not all) GP operators have a variety of biases with respect to both the syntax and the semantics of the trees they produce. Unfortunately these biases can often work against the biases implied by the fitness function, which makes understanding these biases crucial to understanding the behavior of and relationships among the various operators.

In [1, 2] we showed how recent developments in GP schema theory (e.g., [3, 4, 5, 6]) can be used to better understand the biases induced by the standard subtree crossover when genetic programming is applied to variable length linear structures. In particular we showed that subtree crossover has a very strong bias towards oversampling shorter strings and, in some senses, works against bloat. An important followup question, then, is what sort of biases are induced by the common GP mutation operators.

In this paper we apply the schema theory to this question, deriving exact schema equations for subtree mutation on linear structures, using both the FULL and GROW methods to generate the new, random subtrees. Iterating those equations on both a flat fitness landscape and a needle-in-a-haystack style problem, called the one-then-zeros problem, shows that both of these subtree mutation operators have strong biases with regard to the population’s length distribution. Similar to the bias of subtree crossover, we find that these mutation operators are strongly biased in favor of shorter strings in both these fitness domains.

In the following section (Section 2) we review the FULL and GROW mutation operators that will be the objects of this in-

vestigation, and then review the necessary basics of schema theory in Section 3. Section 4 presents exact schema theorems for both the FULL and GROW mutation operators for linear structures on a flat fitness landscape, as well as the results of iterating those schema equations. Section 5 presents the results for the one-then-zeros problem, a simple linear problem that exhibits bloat under standard crossover. We then present conclusions and ideas for future work in Section 6.

## 2 GP mutation

A wide variety of GP mutation operators have been suggested (e.g., [7, 8, 9, 10]), but in this paper we are going to focus on two of the earliest and still most common: FULL and GROW subtree mutation [11]. The schema theory is general enough, however, to handle many of these other mutation operators (see, e.g., [12]).

Both FULL and GROW mutation have the same basic form. First an individual is selected for mutation, then some node in that tree is chosen as the mutation point, and then some new random subtree is generated to be inserted at the mutation point. Thus both resemble crossover, but the subtree being inserted is randomly generated instead of coming from some other tree in the population.

Where FULL and GROW mutation differ is in how the new random subtrees are generated, and in particular how their sizes and shapes are determined. In FULL, each branch of the tree has a specified depth  $D$ . Thus non-terminals are selected until depth  $D$  is reached, at which point terminals are selected.<sup>1</sup> In GROW mutation, one chooses from the set of *all* functions and terminals every time, only terminating a branch if a terminal is chosen. Thus there is no *a priori* limit on either the size or depth of the resulting trees, and [10] shows that under certain circumstances the expected size of trees generated using this process is in fact infinite! As a result, the GROW generation method often has an additional depth limit  $D$  and the constraint that once depth  $D$  is reached only terminals are chosen. [10] presents some related tree generation algorithms that allow more control of the statistical distribution of the resulting trees.

---

<sup>1</sup>It is common for the selection of non-terminals and terminals to be done uniformly, and that will be followed here. As [10] points out, though, one could specify non-uniform selection distributions to bias the node selection process.

For a more detailed schema theory analysis of general subtree mutation operators, as well as the closely related headless chicken crossover, see [12].

### 3 Schema theorem for subtree mutations on linear structures

In [12] we provide the exact schema theory for subtree mutation. In this section we will specialize that to the case of GP on linear structures, and in the subsequent sections we will then use that result to better understand the effects of different mutations operations in two specific situations.

#### 3.1 Preliminary definitions

Before restating the schema theorem for subtree mutation, we present a few definitions that allow us to simplify the theory in the case of linear structures.

##### 3.1.1 Representing schemata

Let  $\mathcal{F}$  be the set of possible non-terminal nodes,  $\mathcal{T}$  be the set of terminals, and  $\mathcal{C} = \mathcal{F} \cup \mathcal{T}$ . In a linear-structure GP individuals can then be seen as sequences of symbols  $c_0 c_1 \dots c_{N-1}$  where  $c_i \in \mathcal{F}$  for  $i < N - 1$  and  $c_{N-1} \in \mathcal{T}$ . We will then define a linear GP *schema* as the same kind of sequence  $c_0 c_1 \dots c_{N-1}$  except that a new “don’t care” symbol ‘=’ is added to both  $\mathcal{F}$  and  $\mathcal{T}$ .<sup>2</sup> Thus schemata represent sets of linear structures, where the positions labelled ‘=’ can be filled in by any element of  $\mathcal{F}$  (or  $\mathcal{T}$  if it is the terminal position). In the following we will use the superscript notation from theory of computation, where  $x^n$  indicates a sequence of  $n$   $x$ ’s;  $x^0$  is the empty string.

A few examples of schema are:

- $(=)^N$ : The set of all sequences of length  $N$ .
- $1(=)^a$ : The set of all sequences of length  $a + 1$  starting with a 1.
- $1(0)^a$ : The singleton set containing the string 1 followed by  $a$  0’s.

##### 3.1.2 Characterizing schemata

In order to count instances of schemata, we need a set of definitions that capture the ways in which schemata are selected and transformed.

**Definition 1 (Proportion in population)**  $\phi(H, t)$  is the proportion of strings in the population at time  $t$  matching schema  $H$ . For finite populations of size  $M$ ,  $\phi(H, t) = m(H, t)/M$ , where  $m(H, t)$  is the number of discrete instances of  $H$  at time  $t$ .

<sup>2</sup>This new ‘=’ symbol plays a role similar to that of the ‘#’ “don’t care” symbol in GA schema theory. For historical reasons, however, ‘#’ has been assigned another meaning in previous work [5, 6, 13]. Schemata which contain no ‘#’s are known as *fixed-size-and-shape* schemata; all schemata discussed in this paper are fixed-size-and-shape schemata.

**Definition 2 (Selection probability)**  $p(H, t)$  is the probability of selecting an instance of schema  $H$  from the population at time  $t$ . This is typically a function of  $\phi(H, t)$ , the fitness distribution, and the details of the selection operators. With fitness proportionate selection, for example,  $p(H, t) = \phi(H, t) * f(H, t) / \bar{f}(t)$ , where  $f(H, t)$  is the average fitness of all the instances of  $H$  in the population at time  $t$  and  $\bar{f}(t)$  is the average fitness of the population at time  $t$ .

**Definition 3 (Transmission probability)**  $\alpha(H, t)$  is the probability that the schema  $H$  will be constructed in the process of creating the population for time  $t + 1$  out of the population at time  $t$ . This will typically be a function of  $p(K, t)$  for the various schemata  $K$  that could play a role in constructing  $H$ , and of the details of the various recombination and mutation operators being used.

We can now model the standard evolutionary algorithm as the transformation

$$\phi(H, t) \xrightarrow{\text{selection}} p(H, t) \xrightarrow{\text{mutation crossover}} \alpha(H, t) \xrightarrow{\text{sampling}} \phi(H, t+1).$$

Here the arrows indicate that some new distribution (on the RHS of the arrow) is generated by applying the specified operation(s) to the previous distribution (on the LHS). So, for example, the process of selection can be seen as a transformation from the distribution of schemata  $\phi(H, t)$  to the selection probability  $p(H, t)$ . A crucial observation is that, for an infinite population,  $\phi(H, t + 1) = \alpha(H, t)$  for  $t \geq 0$ , which means we can iterate these transformations to *exactly* model the behavior of an infinite population over time.

The mutation operators we will be looking at in Sections 4 and 5 will depend on one further definition:

**Definition 4 (Creation probability)**  $\pi_{\text{mut}}(H)$  (or  $\pi_{\text{mut}}(H, t)$  if it depends on the generation  $t$ ) is the probability that some GP subtree mutation operator will generate a new, random subtree that is an element of the schema  $H$ .

##### 3.1.3 Creating instances of schemata

**Definition 5 (u and l)** We define for linear schema  $H = c_0 c_1 \dots c_{N-1}$  the following

$$u(H, i, k) = c_0 c_1 \dots c_{i-1} (=)^{k-i} \quad (1)$$

$$l(H, i, n) = (=)^{n-N+i} c_i c_{i+1} \dots c_{N-1} \quad (2)$$

Here  $u(H, i, k)$  is the schema of length  $k$  containing the leftmost  $i$  symbols of  $H$ , and  $l(H, i, n)$  is the schema of length  $n$  containing the rightmost  $N - i$  symbols of  $H$ .<sup>3</sup> The important thing about  $u$  and  $l$  is that if you use standard crossover to crossover any instance of  $u(H, i, k)$  at position  $i$  with any instance of  $l(H, i, n)$  at position  $n - N + i$ , the result will be an instance of  $H$  provided that  $k + n > N$  and  $0 \uparrow (N - n) \leq i < N \downarrow k$ , where  $\uparrow$  is a binary infix *max* operator, and  $\downarrow$  is a binary infix *min* operator. A related point which is crucial here is that if we perform subtree mutation

<sup>3</sup> $u$  and  $l$  are based on operators  $U$  and  $L$  (see, e.g., [12]) which contain the *upper* and *lower* parts of general, non-linear, GP schemata.

on any instance of  $u(H, i, k)$  at position  $i$ , where the substring swapped in is an instance of  $l(H, i, N - i)$ , then we get an instance of  $H$ . Further, these are the *only* ways to use standard crossover or subtree mutation to construct instances of  $H$ . Thus these definitions let us fully characterize the mechanisms for constructing instances of the linear schemata  $H$ .

To clarify which operator we are working with, we introduce specialized forms of the transmission probability function  $\alpha$ , namely

- $\alpha_{\text{mut}}$  is the transmission probability due specifically to *mutation*,
- $\alpha_{\text{FULL}}$  is the transmission probability due specifically to subtree mutation using the FULL method, and
- $\alpha_{\text{GROW}}$  is the transmission probability due specifically to subtree mutation using the GROW method.

### 3.2 Schema theorem for general mutations

Given the above definitions we can now restate the schema theorem for node-invariant<sup>4</sup> subtree mutation (Corollary 6 from [12]) in a more compact form:

**Theorem 1 (Schema theorem for mutation on linear structures)** *The total transmission probability for a fixed-size-and-shape GP schema  $H = c_0 c_1 \dots c_{N-1}$  under node-invariant subtree mutation is*

$$\begin{aligned} \alpha_{\text{mut}}(H, t) &= \sum_{\substack{k>0 \\ 0 \leq i < k \\ 0 \leq i < N}} \left( \frac{1}{k} * p(u(H, i, k), t) * \pi_{\text{mut}}(l(H, i, N - i)) \right) \\ &= \sum_{\substack{k>0 \\ 0 \leq i < N \downarrow k}} \left( \frac{1}{k} * p(c_0 \dots c_{i-1} (=)^{k-i}, t) * \pi_{\text{mut}}(c_i \dots c_{N-1}) \right) \end{aligned} \quad (3)$$

It is important to realize that this is a general result and applies to many different subtree mutation operators, with the details of the specific operator encapsulated in the  $\pi_{\text{mut}}(c_i \dots c_{N-1})$ .

## 4 Flat fitness landscapes

Given Theorem 1 we can now specialize this for the case of a flat fitness landscape (i.e., all individuals have the same fitness). This is a valuable exercise because it allows us to see the biases of the operators without the added effects of fitness. These biases can, in some cases, be quite strong, and actually work against the (intended) bias induced by the fitness function. (See [2] for examples of this in the case of standard subtree crossover.)

<sup>4</sup>A mutation operator is *node-invariant* if the choice of the mutation point in the parent program depends only on its shape (see [12]). Most commonly used GP operators are node-invariant, and only node-invariant operators will be considered in this paper.

In this section we will specialize Theorem 1 for FULL and GROW mutation respectively, and then look at the length biases these operators induce on a flat fitness landscape.

Before doing this, though, we can use the fact that we are on a flat fitness landscape to make an immediate simplification. In particular we note that for a flat fitness landscape

$$p(H, t) = \phi(H, t),$$

i.e., the probability of selecting an individual matching the schema  $H$  is the same as a proportion  $\phi(H, t)$  of  $H$  in the population at that time.

Our concern here is to better understand the effect of these operators on the distributions of lengths, so we will focus solely on the *lengths* of the strings. If one was interested in other effects, such as mixing, then one would need to track a larger set of schemata, but in this case we only need consider schemata  $H$  of the form  $(=)^N$ .

### 4.1 FULL mutation

To specialize Theorem 1 for FULL mutation, we need to determine

$$\pi_{\text{FULL}}(c_i \dots c_{N-1}) = \pi_{\text{FULL}}((=)^{N-i}).$$

Since FULL always (and exclusively) generates strings of length  $D$ , we know that

$$\pi_{\text{FULL}}((=)^{N-i}) = \delta(N - i = D) \quad (4)$$

where  $\delta(b) = 1$  if  $b$  is *true* and 0 otherwise. This allows us to rewrite Theorem 1:

$$\begin{aligned} \alpha_{\text{FULL}}((=)^N, t) &= \langle \text{Theorem 1; simplifying } p \text{ and } \pi_{\text{FULL}}. \rangle \\ &= \sum_{\substack{k>0 \\ 0 \leq i < N \downarrow k}} \left( \frac{1}{k} * \phi((=)^k, t) * \delta(N - i = D) \right) \\ &= \langle (N - i = D) \equiv (i = N - D) \rangle \\ &= \sum_{k>0} \left( \frac{1}{k} * \delta(0 \leq N - D < N \downarrow k) * \phi((=)^k, t) \right) \\ &= \langle (k > 0 \wedge 0 \leq N - D < N \downarrow k) \equiv (0 \leq N - D < k) \rangle \\ &= \delta(D \leq N) * \sum_{k>N-D} \left( \frac{1}{k} * \phi((=)^k, t) \right) \end{aligned} \quad (5)$$

This, then, gives us:

**Theorem 2 (Schema theorem for FULL mutation on flat fitness landscapes)** *The transmission probability for FULL subtree mutation for depth  $D$  using linear structures on a flat fitness landscape is*

$$\alpha_{\text{FULL}}((=)^N, t) = \delta(D \leq N) * \sum_{k>N-D} \left( \frac{1}{k} * \phi((=)^k, t) \right).$$

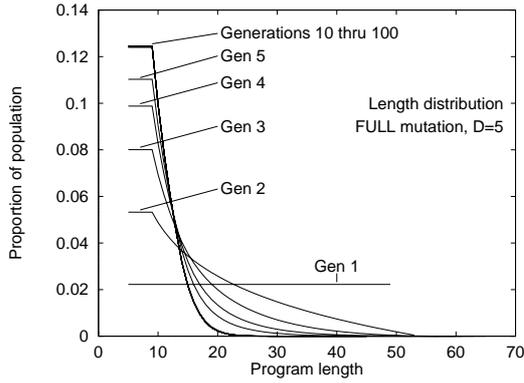


Figure 1: The distribution of lengths over time on a flat fitness landscape using a linear representation. Only FULL mutation is used, with  $D = 5$ . The initial population consists solely of individuals of length 45. The distribution after one round of mutation is a flat distribution, with each length between 5 and 49 equally likely. The distribution quickly moves towards a limit distribution where the proportion is constant for lengths 5 through 9, and then falls off steeply for longer strings.

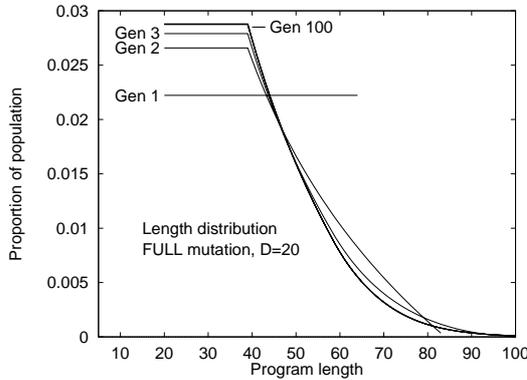


Figure 2: Same as Figure 1 except here  $D = 20$ . The results are quite similar, with the primary differences being a longer flat area and a higher average length. (Note that the range on the vertical axis is considerably smaller than in Figure 1.)

#### 4.2 The biases of FULL mutation

Given Theorem 2, we can iterate this equation from some initial distribution  $\phi((=)^N, 0)$  of lengths and get the *exact* distributions of lengths over time for the case of FULL mutation acting on an infinite population with flat fitness. Figures 1, 2, and 3 show the evolution of the length distribution and the average lengths over 100 generations for two different values of  $D$  (5 and 20).

There are several important things to note about these results:

- In each case the population quickly moves to a limit distribution, so after a brief redistribution of lengths, the distribution of lengths is stable.
- The population moves *very* quickly to an average length of  $2 * D - 1$ , and this average length is inde-

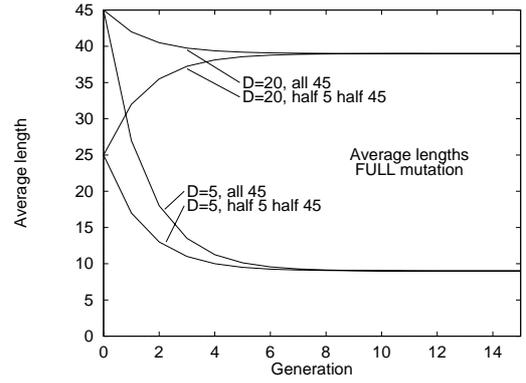


Figure 3: The average lengths using FULL mutation on a flat fitness landscape. The four plots show the effects of different choices of  $D$  and different initial conditions. In the “all 45” cases all the individuals in the initial population were of length 45. In the “half 5 half 45” cases, the initial population was evenly split between individuals of length 5 and length 45.

pendent of the initial distribution. Thus, after the initial redistribution of lengths is over, there will be neither growth (bloat) or shrinkage.

- $\alpha_{\text{FULL}}((=)^N, t)$  is
  - 0 if  $N < D$ ,
  - Some constant value if  $D \leq N < 2 * D$  (i.e., the distribution is “flat” in that region),
  - Rapidly decreasing for  $2 * D \leq N$ .
- The FULL mutation operator on linear structures samples strings whose length  $N$  is in the range  $D \leq N < 2 * D$  *much* more than strings whose length is outside this range.

[14] contains a number of theoretical results on the behavior of standard crossover and both FULL and GROW mutation on linear structures in the flat fitness case. Among these is a proof that the limit of the average length is indeed  $2 * D - 1$  (and therefore independent of the initial distribution), and that the distribution is flat for lengths  $D \leq N < 2 * D$ .

#### 4.3 GROW mutation

Having specialized Theorem 1 for FULL mutation and explored its biases, we will now do the same for GROW mutation.

Here again we need to determine  $\pi_{\text{GROW}}((=)^{N-i}, t)$ , but in the GROW case it is not quite so simple since the length of the string generated by the GROW method is not deterministic. Another complication is that the GROW method has an optional parameter which limits the depth of the trees it will generate. In the linear case the trees generated by GROW tend to be small even without a limit, so in this analysis we will *not* limit the depth; it would, however, be a simple matter to include a depth limit if desired.

When using GROW mutation, the probability of generating a string of a given length ( $N - i$  in our case) is a function of the number of terminals,  $|\mathcal{T}|$ , the number of non-terminals,  $|\mathcal{F}|$ , and the total number of symbols  $|\mathcal{C}| = |\mathcal{T}| + |\mathcal{F}|$ . In particular we can calculate

$$\pi_{\text{GROW}}(c_i \dots c_{N-1}) = \pi_{\text{GROW}}((=)^{N-i})$$

by noting that the first  $N - i - 1$  symbols have to be non-terminals (and thus each chosen with probability  $q = |\mathcal{F}|/|\mathcal{C}|$ ), and the last symbol must be a terminal (and thus chosen with probability  $1 - q = |\mathcal{T}|/|\mathcal{C}|$ ). This then implies that

$$\pi_{\text{GROW}}((=)^{N-i}) = q^{N-i-1} * (1 - q). \quad (6)$$

We can then combine this with Theorem 1 to get

**Theorem 3 (Schema theorem for GROW mutation on flat fitness landscapes)** *The transmission probability for GROW subtree mutation with no depth limit using linear structures on a flat fitness landscape is*

$$\begin{aligned} \alpha_{\text{GROW}}((=)^{N-i}) \\ = \sum_{\substack{k > 0 \\ 0 \leq i < N \downarrow k}} \left( \frac{1}{k} * \phi((=)^k, t) * q^{N-i-1} * (1 - q) \right). \end{aligned}$$

#### 4.4 The biases of GROW mutation

As before, we can iterate the equation in Theorem 3 from some initial distribution  $\phi((=)^N, 0)$  of lengths and get the *exact* distribution of lengths over time for the case of GROW mutation acting on an infinite population with flat fitness.

Figure 4, for example, shows the changing distribution of lengths over time when  $q = 5/6$ . The first generation has a distribution that bears some similarity to the distributions obtained with FULL mutation, but subsequent generations have distributions that are extremely similar to the discrete gamma distributions obtained when using standard subtree crossover on a flat fitness landscape (see [1]).

As was the case with FULL mutation, we see that the distribution for GROW mutation approaches a limit distribution *very* quickly, and that this limit distribution samples the shorter strings (in this case, those with length less than 20) *much* more than it samples the longer ones. The limit distribution for GROW is also independent of the initial distribution, but it does vary with  $q$ . Figure 5 shows the limit distributions for five different values of  $q$ , and we see that both the height and width of the distributions vary a great deal. Figure 6 shows similar variation in the average lengths over this range of values of  $q$ . These results corroborate the suggestions in [10] that when using GROW mutation one should be very careful about the relative number of functions and terminals, as a “small” change in those sets could have a pronounced effect on the size bias induced by the mutation operator.

As mentioned above, the limit distributions in Figure 5 look very similar to the discrete gamma distributions obtained

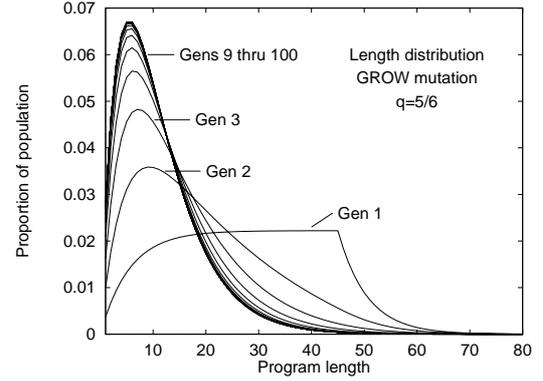


Figure 4: Similar to Figure 2 except here we are using GROW mutation with no depth limit and  $q = 5/6$ . The initial distribution was again all individuals of length 45. The distribution quickly moves to a limit that looks very similar to the discrete gamma distributions generated by subtree crossover alone (see [1]).

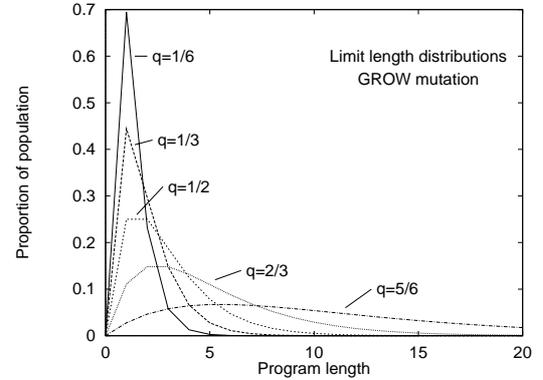


Figure 5: The limit length distributions for the flat fitness landscape when using GROW mutation with no depth limit and several different values of  $q$ . Clearly changing  $q$  can have a substantial effect on the height and width of the distribution, although each distribution looks very similar to the discrete gamma distributions generated by subtree crossover [1]. Note that the  $q = 5/6$  limit is the same as that in Figure 4, although on a different scale.

in [1] for standard subtree crossover. [14] proves that these distributions for GROW mutation are in fact also discrete gamma distributions. This is a rather surprising result since one would not *a priori* expect standard subtree crossover and grow mutation to necessarily induce the same length distributions. It suggests some relationship between the distribution of the lengths of the new, random strings generated by GROW mutation and the distribution of the lengths of the substrings chosen to swap in when using subtree crossover, but this relationship is not yet well understood.

## 5 The one-then-zeros problem

The one-then-zeros problem was introduced in [2] as a simple linear problem that exhibits bloating behavior under standard subtree crossover. In this problem we have  $\mathcal{F} = \{0+, 1+\}$

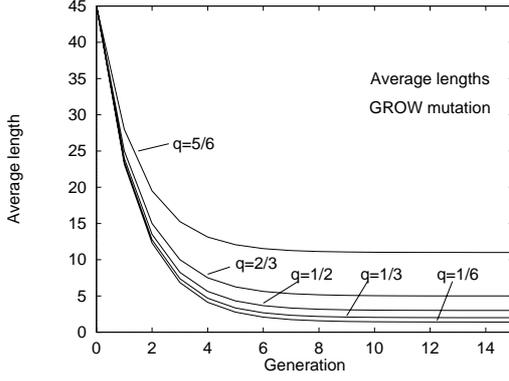


Figure 6: The average lengths over time when using GROW mutation on a flat fitness landscape for the same set of  $q$  values used in Figure 5.

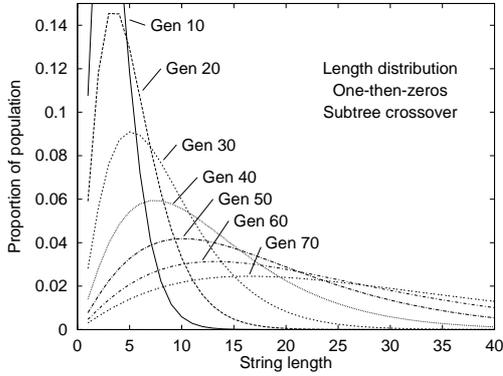


Figure 7: The length distributions for the one-then-zeros problem with standard subtree crossover, plotted every 10 generations. The general shapes resemble the gamma distributions obtained on a flat landscape using standard crossover, but the height of the distributions clearly drops and the peak moves to the right over time, reflecting the increase in average size.

and  $\mathcal{T} = \{0\}$ .  $0+$  and  $1+$  are both unary operators that add 0 and 1 respectively to their arguments. This gives us a problem that is essentially equivalent to studying variable lengths strings of 0's and 1's, with the constraint that the string always end in a 0. Fitness in this problem will be 1 if the string starts with a 1 and has zeros elsewhere, i.e., the string has the form  $1(0)^a$  where  $a > 0$ . The fitness will be 0 otherwise.

In [2] we used schema theory to show that standard subtree crossover induces bloat in this problem. More specifically, in the infinite population case the average size increases steadily for at least the 75 generations over which we iterated the schema equations. The overall length distributions appeared to be similar to the discrete gamma distributions obtained on the flat fitness landscapes, as shown in Figure 7. A key question, then, is whether FULL and GROW mutation will also induce bloat or gamma-like distributions.

Following a development similar to that in [2], we can derive the following specialization of Theorem 1 for the one-then-zeros problem:

**Theorem 4 (Schema theorem for mutation on one-then-zeros)** *The total transmission probabilities induced by mutation in the one-then-zeros problem for the schemata  $(=)^N$  and  $1(0)^a$  are given by:*

$$\alpha_{mut}((=)^N, t) = \sum_{\substack{k>0 \\ 0 \leq i < N \downarrow k}} \left( \frac{1}{k} * p(1(0)^{k-1}, t) * \pi_{mut}((=)^{N-i}) \right)$$

and

$$\alpha_{mut}(1(0)^a, t) = \sum_{k>0} \left( \frac{1}{k} * p(1(0)^{k-1}, t) * \pi_{mut}(1(0)^a) \right) + \sum_{\substack{k>0 \\ 1 \leq i < (a+1) \downarrow k}} \left( \frac{1}{k} * p(1(0)^{k-1}, t) * \pi_{mut}(0^{a+1-i}) \right).$$

One advantage of using the schema theory is that one can use it to precisely determine which schemata need to be tracked in order to iterate the schema equations. For the one-then-zeros problem, for example, working through the schema theory as is done in [2] shows that to track schemata of the form  $(=)^N$  we also need to track schemata of the form  $1(0)^a$ . Further, the theory shows that no other schemata need to be tracked, so we know that these are necessary and sufficient.

### 5.1 FULL mutation

To specialize Theorem 4 for the FULL mutation with depth  $D$  we need to specialize the appropriate  $\pi_{mut}$  expressions:

$$\begin{aligned} \pi_{FULL}((=)^{N-i}) &= 1 \quad \text{if } i = N - D \\ &= 0 \quad \text{otherwise} \\ \pi_{FULL}(1(0)^a) &= 1/2^{D-1} \quad \text{if } a = D - 1 \\ &= 0 \quad \text{otherwise} \\ \pi_{FULL}(0^{a+1-i}) &= 1/2^{D-1} \quad \text{if } i = a + 1 - D \\ &= 0 \quad \text{otherwise} \end{aligned}$$

Plugging these into Theorem 4 yields

**Theorem 5 (Schema theorem for FULL mutation on one-then-zeros)** *The total transmission probabilities induced by FULL mutation in the one-then-zeros problem for the schemata  $(=)^N$  and  $1(0)^a$  are as follows. First,*

$$\alpha_{FULL}((=)^N, t) = \sum_{k>N-D} \left( \frac{1}{k} * p(1(0)^{k-1}, t) \right)$$

if  $N - D \geq 0$ , and is 0 otherwise. Second,

$$\alpha_{FULL}(1(0)^a, t) = \sum_{k>a-D+1} \left( \frac{1}{k} * p(1(0)^{k-1}, t) / 2^{D-1} \right)$$

if  $a \geq D - 1$ , and is 0 otherwise.

We iterated these equations for 100 generations as in Section 4.1 to see what biases FULL mutation had in this domain. Rather surprisingly, the length distribution generated when we apply FULL mutation to the one-then-zeros problem is in fact identical to the length distribution obtained when we applied FULL mutation to the flat fitness landscape earlier in Section 4.1. In other words, Figure 1 illustrates the behavior of FULL mutation on both problems.

The reason for this appears to be the fact that the odds of FULL mutation creating a “correct” offspring (i.e., generating the new, random substring necessary to generate an individual having fitness 1) is independent of the length of the parent. Thus FULL mutation will generate many unfit individuals in the one-then-zeros problem, but *all* lengths will have the *same* proportion of unfit strings which will be ignored in the selection process.

It seems likely that this is a rather unusual artifact of this particular problem, and that there are certainly many problems where the FULL mutation operator induces different distributions of lengths than the distribution it induces on the flat fitness landscape.

## 5.2 GROW mutation

To specialize Theorem 4 for GROW mutation (again assuming no depth limit) we again need to specialize the appropriate  $\pi_{\text{mut}}$  expressions. In this problem we have two non-terminals and one terminal, so  $q = 2/3$ . This then gives us

$$\begin{aligned}\pi_{\text{GROW}}((=)^{N-i}) &= (2/3)^{N-i-1} * (1/3) \\ \pi_{\text{GROW}}(1(0)^a) &= (1/3)^{a+1} \\ \pi_{\text{GROW}}(0^{a+1-i}) &= (1/3)^{a+1-i}\end{aligned}$$

Plugging these into Theorem 4 then yields the following:

**Theorem 6 (Schema theorem for GROW mutation on one-then-zeros)** *The total transmission probabilities induced by GROW mutation in the one-then-zeros problem for the schemata  $(=)^N$  and  $1(0)^a$  are*

$$\alpha_{\text{GROW}}((=)^N, t) = \left(\frac{2}{3}\right)^N * \sum_{k>0} \left(\frac{1}{k} * p(1(0)^{k-1}, t) * \left(\left(\frac{3}{2}\right)^{N-k} - 1\right)\right)$$

and

$$\alpha_{\text{GROW}}(1(0)^a, t) = \frac{\sum_{k>0} \left(\frac{1}{k} * p(1(0)^{k-1}, t) * (3^{(a+1)\downarrow k} - 1)\right)}{2 * 3^{a+1}}$$

Figure 8 shows that the length distribution induced by GROW mutation on the one-then-zeros problem is again quite similar to those induced in the flat fitness landscape (see, e.g., Figure 4). While the shapes are similar, however, the specifics of the distribution are indeed different, with the average length in the one-then-zeros case being smaller than that for the flat fitness landscape. This, however, is not surprising since the odds of GROW mutation generating a “correct”

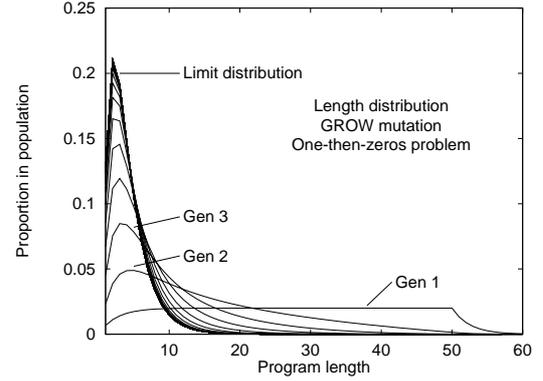


Figure 8: The length distributions for the one-then-zeros problem with GROW mutation. The general shape is quite similar to those obtained on the flat fitness landscape (see, e.g., Figure 4).

new, random substring falls exponentially with the length of the generated string. Thus short strings are much more likely generate fit individuals than long strings, generating a bias towards shorter strings. The surprising result, perhaps, is that the bias is not stronger.

## 6 Conclusions and future work

Continuing earlier work on subtree crossover on linear structures [1, 2], we have shown here that schema theory can be effectively applied to the problem of understanding the length biases induced by both FULL and GROW subtree mutation. On a flat fitness landscape we found that both operators had a distinct bias towards oversampling shorter strings, and in the case of GROW mutation we found that the bias was identical to that found earlier for subtree crossover. On the one-then-zeros problem, which was shown in [2] to bloat under subtree crossover, we found that both mutation operators quickly led to a stable state, after which there was no change in the distribution of lengths and, consequently, no change in the average length.

All the results reported here are for GP on linear structures, and one should always be cautious about over generalizing these sorts of results to the more general case of tree structures. Some of the results reported here for linear structures (such as those for FULL mutation) may well be qualitatively similar to the results for tree structures, while others (such as those for GROW mutation) may well exhibit important differences when applied to tree structures.

There are at least three obvious areas for extension of this work. One is to explore the biases of mutation when applied to more general tree structures. Another is to explore the biases of some of the other mutation operators that have been proposed (such as node level mutation). The third is to explore the biases induced by combinations of different operators, such as standard subtree crossover and mutation.

This last extension is particularly important because could allow us to begin the difficult process of understand how op-

erators interact. Evolutionary computation systems typically involve a wide variety of processes (e.g., selection, recombination, mutation, and fitness evaluation), and these many processes often interact in complex ways that are hard to predict beforehand or understand after the fact. Once schema equations like those presented here are generated, it is reasonably straightforward to combine and iterate them to see what happens when multiple operators interact.

It would also be nice to find closed forms for the fixed points of many of the recursive equations presented here. [14], for example, provides closed forms for several of the flat fitness cases, including a proof that the limit distribution induced by GROW mutation on a flat fitness landscape is in fact a discrete gamma distribution like that seen in [1] for standard subtree crossover. Unfortunately it is not clear how we might provide closed forms for more complex cases like those for the one-then-zeros problem.

## Acknowledgements

The authors would like to thank the members of the EEBIC (Evolutionary and Emergent Behaviour Intelligence and Computation) group at Birmingham for helpful comments and discussion. We are also grateful for the valuable feedback provided by two anonymous reviewers.

The first author 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.

## Bibliography

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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, William B. Langdon, Julian F. Miller, Peter Nordin, and Terence C. Fogarty, Eds., Edinburgh, 15-16 Apr. 2000, vol. 1802 of LNCS, pp. 163–180, Springer-Verlag.
- [6] Riccardo Poli, "Exact schema theorem and effective fitness for GP with one-point crossover", in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, Darrell Whitley, David Goldberg, Erick Cantu-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, Eds., Las Vegas, Nevada, USA, 10-12 July 2000, pp. 469–476, Morgan Kaufmann.
- [7] Una-May O'Reilly and Franz Oppacher, "A comparative analysis of GP", in *Advances in Genetic Programming 2*, Peter J. Angeline and K. E. Kinneer, Jr., Eds., chapter 2, pp. 23–44. MIT Press, Cambridge, MA, USA, 1996.
- [8] Una-May O'Reilly, *An Analysis of Genetic Programming*, PhD thesis, Carleton University, Ottawa–Carleton Institute for Computer Science, Ottawa, Ontario, Canada, 22 Sept. 1995.
- [9] Peter J. Angeline, "Subtree crossover: Building block engine or macro-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. 9–17, Morgan Kaufmann.
- [10] Sean Luke, "Two fast tree-creation algorithms for genetic programming", *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 274–283, 2000.
- [11] John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [12] Riccardo Poli and Nicholas F. McPhee, "Exact GP schema theory for headless chicken crossover and subtree mutation", in *Proceedings of the 2001 Congress on Evolutionary Computation CEC 2001 (this volume)*, Seoul, Korea, May 2001.
- [13] 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.
- [14] Jon E. Rowe and Nicholas F. McPhee, "The effects of crossover and mutation operators on variable length linear structures", Tech. Rep. CSRP-01-7, University of Birmingham, School of Computer Science, January 2001.