On the Limiting Distribution of Program Sizes in Tree-Based Genetic Programming

Riccardo Poli, William B. Langdon, and Stephen Dignum

Department of Computer Science, University of Essex, UK

Abstract. We provide strong theoretical and experimental evidence that standard sub-tree crossover with uniform selection of crossover points pushes a population of a-ary GP trees towards a distribution of tree sizes of the form:

$$\Pr\{n\} = (1 - ap_a)\binom{an+1}{n}(1-p_a)^{(a-1)n+1}\,p_a^n$$

where n is the number of internal nodes in a tree and p_a is a constant. This result generalises the result previously reported for the case $a = 1$.

1 Introduction

For most problems the ratio between the size of the search spaces and the number of acceptable solutions grows exponentially with the size of the problem. So, even with today's powerful computers, for many problems one can hope to find solutions with a particular search algorithm only if the algorithms is biased in such a way to sample preferentially the areas of the search space where solutions are denser. This situation is often informally referred to as an *algorithm being well-matched to a problem*.

Having a full characterisation of the search biases of a search algorithm is a precondition to understand whether or not the algorithm is well-matched to a problem. (The second precondition is the availability of a characterisation of the problem, e.g., information on the distribution of solutions in the search space.)

In evolutionary algorithms this requires understanding the biases of the genetic operators. These biases are fairly well understood for mutation and crossover in the case of fixed-length representations (e.g., binary GAs) [4,16] and for selection (which is representation independent) [5,1,2,11]. However, the situation is much sketchier for variable-length representations. In particular, except for the limiting case of linear-trees (built only using arity-1 primitives and terminals) [12,15,13,14], we still know very little about the search biases of standard GP crossover.

In this paper we provide an exact characterisation of the limiting distribution of tree sizes towards which sub-tree crossover, when acting on its own, pushes the population. As we will see, obtaining this type of result is complex, and so we will limit our attention to the case where the primitive set includes only terminals and primitives of one other arity.

M. Ebner et al. (Eds.): EuroGP 2007, LNCS 4445, pp. 193–204, 2007.

The paper is organised as follows. In Section 2 we summarise results from branching processes theory. In Section 3 we develop a formulation for the fixed-point distribution of tree sizes under repeated crossover. In Section 4 we derive an equation that describes how the distribution of tree sizes changes generation after generation under the effects of crossover. The distribution proposed in Section 3 is one of the two elements of such an equation, when evaluated at its fixed-point. In Section 5 we develop an explicit formulation for the second element: the distribution of subtree sizes. In principle these ingredients would allow one to check mathematically whether the proposed size distribution is indeed a fixed point for crossover-based evolution. Proving this result is, however, beyond our mathematical capabilities. Therefore, to corroborate our conjecture we present strong empirical evidence and numerical integrations of the tree-size distribution evolution equation in Section 6. We make some final remarks in Section 7.

2 Mathematical Preliminaries

2.1 Branching Processes and Lagrange Distribution

In probability theory, a *discrete-time branching process* [17] is a Markov process that models a population in which each individual in a generation produces some random number of descendants, and where the probability of generating a successors, $p(a)$, is fixed. This leads to a (family) tree.

Branching processes have at least one application in GP: if no limit is imposed on tree size or depth, the tree shapes produced by the "grow" method, often used to initialise GP populations and to perform sub-tree mutation, obey a branching process. In this case a is the arity of primitives and p_a is the probability of using primitives of arity a when choosing nodes in the "grow" method.

The distribution of tree sizes for a branching process follows a *Lagrange distribution* [3,6]. More precisely, the probability of the process leading to a total of ℓ individuals being generated is

$$\Pr\{L=\ell\} = \begin{cases} 0 & \text{if } \ell = 0, \\ \frac{1}{\ell}\mathcal{C}(t^{\ell-1})\left\{(g(t))^{\ell}\right\} & \text{for } \ell = 1,2,3,\cdots, \end{cases} \tag{1}$$

where $g(t) = \sum_a p_a\, t^a$ is the probability generating function of the distribution p_a and $\mathcal{C}(t^m)$ denotes "the coefficient of t^m in".

If one considers a process where only nodes of arity a and 0 are allowed (i.e., $p_0 + p_a = 1$), then $g(t) = p_0 + p_a t^a$. So, for $\ell > 0$ we have

$$\mathcal{C}(t^{\ell-1})\left\{(g(t))^{\ell}\right\} = \mathcal{C}(t^{\ell-1})\left\{(p_0 + p_a t^a)^{\ell}\right\} = \mathcal{C}(t^{\ell-1})\left\{\sum_{k=0}^{\ell}\binom{\ell}{k} p_0^{\ell-k}\, p_a^k\, t^{ak}\right\}$$

Since $\mathcal{C}(t^{\ell-1})$ will pick out the coefficient of the power of t for which $\ell - 1 = ak$, i.e., $k = \frac{\ell-1}{a}$, we then have

$$\Pr\{L=\ell\} = \begin{cases} \frac{1}{\ell}\binom{\ell}{\frac{\ell-1}{a}}(1-p_a)^{\ell-\frac{\ell-1}{a}}\, p_a^{\frac{\ell-1}{a}} & \text{if } \ell - 1 \text{ is a multiple of } a, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Note that, since only arity 0 and arity a primitives are allowed, a tree with $\ell - 1$ nodes has $n = \frac{\ell-1}{a}$ internal nodes and $\ell = an + 1$. So, we can rewrite the previous equation in terms of internal nodes as

$$\Pr\{N = n\} = C_{\mathcal{T}}\,(a, n)\;(1 - p_a)^{(a-1)n+1}\,p_a^n, \tag{3}$$

where

$$C_{\mathcal{T}}\,(a, n) = \frac{1}{an + 1}\binom{an + 1}{n} \tag{4}$$

is a *generalised Catalan number* [7]. The Catalan number is the number of different trees with n internal nodes of arity a (and, of course, $(a - 1)n + 1$ leaves). This can be interpreted as saying that in a branching process all trees of a particular size have a probability of being created which depends only on how many nodes/primitives of each kind the tree contains. It also means that the different parts of the trees created by a branching process are uncorrelated.

2.2 Moments of the Tree-Size Distribution in a Branching Process

It is possible to compute the moments of Lagrange distributions starting from the the cumulants g_i of the probability density functions generated by power series in t of $g(t)$ [6,3]. The mean progeny produced by a branching process is:

$$E[L] = \frac{1}{1 - g_1} \tag{5}$$

and the variance is

$$Var[L] = \frac{g_2}{(1 - g_1)^3}, \tag{6}$$

where $g_1 = E[A]$ and $g_2 = Var[A]$, A being a stochastic variable representing a node's arity. Since trees contain only arity 0 and arity a nodes, we can easily compute these two cumulants:

$$g_1 = \sum_k kp_k = ap_a \tag{7}$$

$$g_2 = E[A^2] - (E[A])^2 = a^2 p_a - (ap_a)^2 = a^2 p_a(1 - p_a) \tag{8}$$

So, the mean tree size in our branching process is

$$E[L] = \frac{1}{1 - ap_a} \tag{9}$$

and the variance is

$$Var[L] = \frac{a^2 p_a(1 - p_a)}{(1 - ap_a)^3}. \tag{10}$$

From these two, we then obtain the second non-central moment

$$E[L^2] = Var[L] + (E[L])^2 = \frac{(a - 1)ap_a - a^2 p_a^2 + 1}{(1 - ap_a)^3} \tag{11}$$

Note that (9) matches the formula for the means size of programs built by the “grow” method reported in [10] and that (9) is a special case of it.

3 The Distribution of Tree Sizes Under Crossover

In the absence of selection, if a population of GP trees undergoes repeated crossovers, the population tends to a limiting distribution of sizes and shapes. This is the result of the specific bias of subtree crossover.[1] Effectively after a while, every node in every individual in the population will have been placed at its particular position as a result of one or multiple crossover events. So, any correlations present in the shapes in the initial generation will have been broken by crossover.

As we saw in the previous section, complete decorrelation in the different parts of a tree is a characteristic of branching processes. Within the class of trees of a given size, each shape is equally likely. So, we postulate that the limiting distribution of tree sizes under repeated crossover will be one where this happens. That is, we assume that at the fixed-point, the shape distribution is

$$\Pr\{\text{Shape with } n \text{ nodes of arity } a\} = w(n,a)\,(1-p_a)^{(a-1)n+1}\,p_a^n \tag{12}$$

where $w(n,a)$ is an appropriate sequence of weights to be determined and p_a is a parameter, also to be determined. So, the probability of picking a tree with n internal nodes from the population is

$$\Pr\{n\} = C_{\mathcal{I}}\,(a,n)\,w(n,a)\,(1-p_a)^{(a-1)n+1}\,p_a^n \tag{13}$$

What constraints do we have on the parameters $w(n,a)$ and p_a? Firstly, they must be such that the distribution of shapes is indeed a probability distribution. In particular we require

$$\sum_{n\geq 0} \Pr\{n\} = 1. \tag{14}$$

Secondly, it is well-known that on average subtree crossover does not alter the mean size of program trees in a population [14]. So, we also require that

$$\sum_{n\geq 0} (an+1)\Pr\{n\} = \mu_0, \tag{15}$$

where μ_0 is the average size of the individuals in the population at generation 0. Thirdly, we require (13) to be a generalisation of the results reported in [12,15,13,14] for arity 1 functions, which we here rewrite as

$$\Pr\{\ell\} = \ell r^{\ell-1}(1-r)^2, \tag{16}$$

where

$$r = (\mu_0 - 1)/(\mu_0 + 1). \tag{17}$$

We can do this by setting $a = 1$ in (13) and $\ell = n + 1$ and so $\Pr\{\ell\}$ in (16) is the same quantity as $\Pr\{n\}$ in (13). Equating the results we obtain

$$w(n,1)\,(1-p_1)\,p_1^n = (n+1)(1-r)^2 r^n \tag{18}$$

[1] Naturally, stochastic effects such as drift mean that in any finite population there is still random variation. However, in large populations these effects can be neglected.

since $C_{\mathcal{T}}(1,n) = 1$. The most natural match between r.h.s. and l.h.s. of (18) appears to be one where $p_1 = r$ and $w(n,1) = (n+1)(1-p_1)$.

This last constraint completely rules out that $w(a,n)$ be constant, indicating that the length distribution under subtree crossover cannot be purely the result of a branching process (i.e., it is not Lagrangian). Instead, it suggests that $\Pr\{\ell\}$ is the product between the frequency provided by a branching process and the length ℓ of programs. So, we postulate that in general

$$w(a,n) = (an+1)f(p_a) \tag{19}$$

where $f(p_a)$ is a function of p_a to be determined.

With this assumption, we impose (14), i.e., that probabilities sum to 1, obtaining:

$$f(p_a) = \frac{1}{\sum_{n\geq 0}(an+1)C_{\mathcal{T}}(a,n)\,(1-p_a)^{(a-1)n+1}\,p_a^n}. \tag{20}$$

The denominator of this equation is (by definition) $E[aN+1] = \sum_n (an+1)\Pr\{N=n\}$, where $\Pr\{N=n\}$ is given in (3). So, it is the expected length of the trees generated by a branching process where arity a nodes are used with probability p_a and arity 0 nodes used with probability $1-p_a$. So, from (9) we have

$$f(p_a) = (1-ap_a), \tag{21}$$

and so $w(a,n) = (an+1)(1-ap_a)$. As a result, we can now explicitly write the *tree-size distribution at the crossover fixed-point*:

$$\Pr\{n\} = (1-ap_a)\binom{an+1}{n}(1-p_a)^{(a-1)n+1}\,p_a^n \tag{22}$$

where we used the explicit expression of $C_{\mathcal{T}}(a,n)$ in (4). This is the fixed-point tree-size distribution we were looking for. This distribution belongs to a family of distributions called *Lagrange distributions of the second kind* [9,8], which, until now, have never been related to branching processes and trees.

We can now impose constraint (15), i.e., equality of means, to infer the value of p_a:

$$\begin{aligned}
\mu_0 &= \sum_{n\geq 0}(an+1)\Pr\{n\} \\
&= \sum_{n\geq 0}(1-ap_a)(an+1)^2 C_{\mathcal{T}}(a,n)\;(1-p_a)^{(a-1)n+1}\,p_a^n \\
&= (1-ap_a)\sum_{n\geq 0}(an+1)^2\Pr\{N=n\} \\
&= (1-ap_a)E[L^2] \qquad\qquad \text{(by definition)},
\end{aligned}$$

and so, from (11),

$$\mu_0 = \frac{(a-1)ap_a - a^2p_a^2 + 1}{(1-ap_a)^2}. \tag{23}$$

By solving this equation for p_a we obtain

$$p_a = \frac{2\mu_0 + (a-1) - \sqrt{((1-a) - 2\mu_0)^2 + 4(1-\mu_0^2)}}{2a(1+\mu_0)} \tag{24}$$

which, encouragingly, for $a = 1$ collapses to the familiar $p_a = \frac{\mu_0 - 1}{\mu_0 + 1}$ (see (17)).

4 Evolution of the Tree-Size Distribution

Let us term *supertree* the part of a tree remaining after the removal of a subtree rooted at a particular crossover point. The size, L, of a tree after crossover is a stochastic variable obtained by adding the size X of a supertree randomly drawn from the population with the size Y of a subtree also randomly drawn from the population. I.e., $L = X + Y$. It follows that the probability distribution of L is the convolution of the subtree size distribution with the supertree size distribution. That is:

$$\Pr\{L = \ell\} = \sum_{i=0}^{\ell} \Pr\{X = i\} \Pr\{Y = \ell - i\} \tag{25}$$

For the case where internal nodes of arity a only are allowed, effectively a supertree always contains ja nodes and a subtree always contains $ka + 1$ nodes, where j and k are suitable non-negative integers. Therefore, (25) can be rewritten in terms of internal nodes. So, if N is the number of internal nodes of the tree, N_X the internal nodes in the supertree and N_Y the internal nodes in the subtree

$$\Pr\{N = n\} = \sum_{i=0}^{n} \Pr\{N_X = i\} \Pr\{N_Y = n - i\} \tag{26}$$

Naturally we have can interpret $\Pr\{N_X = i\}$ as a marginal and, so,

$$\Pr\{N_X = i\} = \sum_k \Pr\{N_X = i, N = k\} = \sum_k \Pr\{N_X = i | N = k\} \Pr\{N = k\} \tag{27}$$

where $\Pr\{N_X = i | N = k\}$ is the probability of extracting supertrees of size i from individuals of size k and $\Pr\{N = k\}$ is the distribution of tree sizes in the population. Of course, $\Pr\{N_X = i | N = k\} = 0$ for $k < i$, and so

$$\Pr\{N_X = i\} = \sum_{k \geq i} \Pr\{N_X = i | N = k\} \Pr\{N = k\} \tag{28}$$

We can similarly decompose $\Pr\{N_Y = n - i\}$ obtaining

$$\Pr\{N_Y = n - i\} = \sum_{k \geq n-i} \Pr\{N_Y = n - i | N = k\} \Pr\{N = k\} \tag{29}$$

Naturally, standard GP crossover is symmetric, and, therefore, the probability of extracting a *supertree* of size i from a tree of size k is identical to the probability of extracting a *subtree* of size $k-i$ from a tree of size k, i.e., $\Pr\{N_X = i|N = k\} = \Pr\{N_Y = k-i|N = k\}$. So, we have

$$\Pr\{N_X = i\} = \sum_{k \geq i} \Pr\{N_Y = k-i|N = k\} \Pr\{N = k\}. \tag{30}$$

By substituting (29) and (30) in (31) we finally obtain our *size-distribution evolution equation*:

$$\Pr\{n\}_{\text{new}} = \sum_{i=0}^{n} \sum_{k_1 \geq i} p(k_1 - i, k_1) \Pr\{k_1\} \sum_{k_2 \geq n-i} p(n-i, k_2) \Pr\{k_2\} \tag{31}$$

where we used the shorthand notation $p(a,b) = \Pr\{N_Y = a|N = b\}$ and $\Pr\{a\} = \Pr\{N = a\}$.

If the size distribution we propose in (22) is indeed the limiting distribution of sizes obtained by crossover in the absence of selection, then when one replaces $\Pr\{k_i\}$ with (22) in the r.h.s. of the previous equation, upon simplification one should obtain (22) again. This requires, however, the specification of the distribution of subtree sizes $p(a,b)$. In the next section we will obtain this distribution for the case where the limiting distribution of shapes follow the second assumption we used to derive (22), i.e., that within each class of tree sizes, all possible tree shapes are equally likely.

5 Subtree Distribution at the Crossover Fixed-Point

Let $s(n,k)$ be the expected number of subtrees with k internal nodes in trees with n internal nodes (where we draw trees of a particular size with uniform probability). Naturally, $s(n,n) = 1$. Also, under the assumption that only a-ary nodes and leaves can be used in the tree, we also have $s(n,0) = (a-1)n+1$.

Let us consider all trees of size $n > 0$. These must all have a a-ary root node ($a > 0$). Let n_i be the size of child i of the root (naturally, $\sum_i n_i = n-1$). Then we can divide up the space of trees of size n into groups based on the values of n_i. In each group there are $\prod_i C_{\mathcal{T}}(a, n_i)$ trees and, so, the probability of randomly drawing a tree belonging to a specific group when sampling trees of length n is given by $\frac{\prod_i C_{\mathcal{T}}(a,n_i)}{C_{\mathcal{T}}(a,n)}$. So, for example, the first group is characterised by $n_1 = 0$, $n_2 = 0$, $\cdots$, $n_a = n-1$ and contains $(C_{\mathcal{T}}(a,0))^{a-1} C_{\mathcal{T}}(a, n-1) = C_{\mathcal{T}}(a, n-1)$ trees. So, the probability of randomly obtaining a member of this group is $C_{\mathcal{T}}(a, n-1) / C_{\mathcal{T}}(a,n) = \frac{an+1}{a(n-1)+1} \binom{a(n-1)+1}{n-1} / \binom{an+1}{n}$.

Let $s_i(n_i, k)$ be the expected number of subtrees of size k for child i of the root. We assume that we know these quantities and we want to compute $s(n,k)$ on the basis of the $s_i(n_i,k)$. Clearly, for most values of k and n_i, $s(n,k)$ is simply going to be the sum of the $s_i(n_i,k)$'s, i.e., the number of trees of a given size in our tree is just the sum of the trees of that same size in all the children of the root note. There are, however, special cases where we need to be more careful. In particular, if $k > n_i$, then $s_i(n_i, k) = 0$ and, therefore, $s_i(n_i, n) = 0$. However, $s(n, n) = 1$. So, there is one exception to the summation rule.

Below we will formalise the rule. However, before we do that, let us consider the effect of our assumption that within each length class all possible tree shapes happen with equal chance. This assumption leads to the fact that both trees and subtrees must follow the same subtree distribution, i.e., we have that $s_i(n_i, k) = s(n_i, k)$. Also, in order to compute $s(n, k)$ we need to sum over all possible ways in which we can draw the n_i's ensuring that the correct probability for each is considered. All this is accounted for in the following recursion:

$$s(n,k) = \begin{cases} 0 & \text{if } n < k, \\ 1 & \text{if } n = k, \\ \displaystyle\sum_{\sum n_i = n-1} \left(\frac{\prod_i C_{\mathcal{T}}(a, n_i)}{C_{\mathcal{T}}(a, n)} \right) \left(\sum_i s(n_i, k) \right) & \text{otherwise.} \end{cases} \tag{32}$$

So, the probability of drawing a tree with k internal nodes out of the class of trees of n nodes is given by

$$p(n,k) = \frac{s(n,k)}{\sum_k s(n,k)} = \frac{s(n,k)}{an+1} \tag{33}$$

since, rather obviously, $\sum_k s(n, k)$ is the total number of nodes in a tree with n internal nodes of arity a. As a result we can write

$$p(n,k) = \frac{\delta(k=n)}{an+1} + \delta(k<n) \sum_{\sum n_i = n-1} \left(\frac{\prod_i C_{\mathcal{T}}(a, n_i)}{C_{\mathcal{T}}(a, n)} \right) \sum_i \left(\frac{an_i + 1}{an+1} \right) p(n_i, k) \tag{34}$$

where $\delta(x) = 1$ if x is true, 0 otherwise.

6 Conjecture or Theorem?

In principle we now have all the ingredients to prove that (22) and the related distribution (34) are the fixedpoint for the tree-size-distribution evolution equation (31). However, the recursive nature of (34) and the complexities of simplifying sums of products of Catalan numbers make proving the result extraordinarily difficult (except for the case $a = 1$, since this leads directly to the result already proven in [12,15,13,14]).

To corroborate our result, we have therefore followed two alternative approaches. Firstly, we have collected empirical data on the size distributions obtained with different initialisations and for primitives of different arities in populations under the effect of crossover only. Secondly, we have performed a numerical integration of the r.h.s. of (31) at the assumed fixed point (for different values of a and μ_0) to verify if the resulting values for the l.h.s. matched the theoretical prediction. We describe the results of our tests in Sections 6.1 and 6.2.

6.1 Empirical Validation

We performed runs of a GP system in Java, with populations of 100,000 individuals, run for 500 generations. We used such large population sizes to reduce stochastic effects such as drift of the mean program size and to ensure that enough programs in each length-class were available. Similarly we performed a large number of generations to ensure that the initial conditions (see below) were completely washed out.

Only terminals and primitives of arity a (for $a = 1, 2, 3, 4, 5$) were allowed. Initialisation was performed using the "full" method. With this method, initial trees included $\mu_0 = d + 1$ primitives for $a = 1$ and $\mu_0 = \frac{1-a^{d+1}}{1-a}$ primitives for $a > 1$. Initial depth was 3, 4, or 8 (the root node being at depth 0).

During our runs we recorded histograms of program sizes, one for each generation, for sizes between 1 and 1000. Note that, because of the large population sizes and the particular objective of our runs (i.e., the study of size distributions), there was no need to collect data in multiple independent runs (as it is customary for other types of empirical studies).

In all cases the match between theoretical predictions and empirical data is striking. Compare, for example, the theoretical predictions and the empirical results shown in Figures 1–3.

6.2 Numerical Integration

The exact numerical integration of the r.h.s. of (31) at the assumed fixed point would require performing infinite sums, which is clearly impossible. So, we chose to limit sums over tree sizes to `limit` $= 5\mu_0$, effectively assuming that $\Pr\{n\} = 0$ for $n >$ `limit`. Naturally, this leads to some integration errors, but these turned out to be negligible for the purpose of confirming whether or not the resulting values for the l.h.s. matched the theoretical prediction.

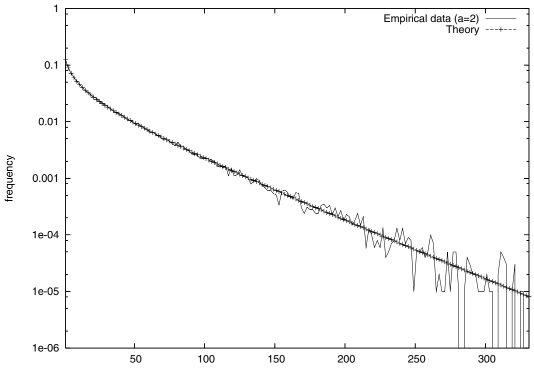

Fig. 1. Comparison between empirical and theoretical program size distributions for binary trees ($a = 2$) initialised with full method ($d = 4$, initial mean size $\mu_0 = 31$, mean size after 500 generations $\mu = 27.26044$)

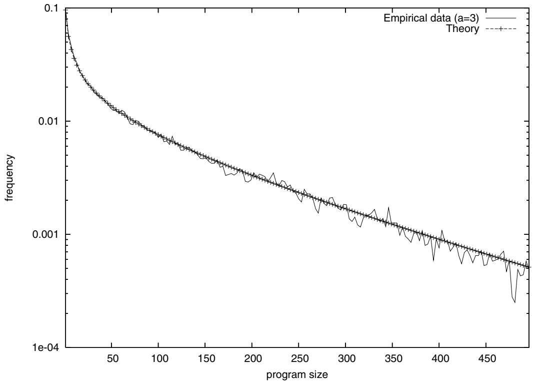

Fig. 2. Comparison between empirical and theoretical program size distributions for ternary trees ($a = 3$) initialised with full method ($d = 4$, initial mean size $\mu_0 = 121$, mean size after 500 generations $\mu = 109.10284$)

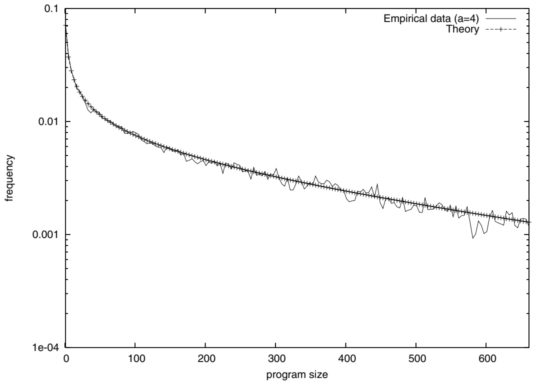

Fig. 3. Comparison between empirical and theoretical program size distributions for quaternary trees ($a = 4$) initialised with full method ($d = 4$, initial mean size $\mu_0 = 341$, mean size after 500 generations $\mu = 361.73052$)

We performed the integration for tree sizes (internal nodes) between 0 and `limit` inclusive, and for a variety of a's and μ_0's. In all cases, the output distribution computed via (31) was effectively indistinguishable from (22).

As an example of the degree of accuracy in the match between input and output size distributions, we show in Figure 4 a comparison between our conjectured program size distributions for binary trees and the output produced by (31). The plots of the distributions overlap almost perfectly. Indeed, absolute errors range between -2.8781×10^{-5} and -6.7263×10^{-7}, corresponding to relative errors between -0.023% and -0.049%.

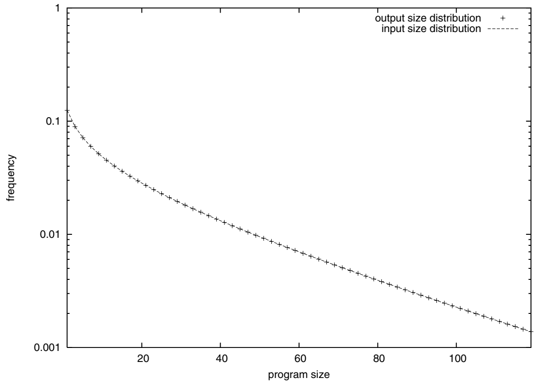

Fig. 4. Comparison between the program size distributions conjectured in (22) and output produced by (31) for binary trees ($a = 2$) of mean size $\mu = 27.26044$, as in Figure 1

7 Discussion and Conclusions

We have provided very strong theoretical and empirical evidence that the distribution of tree sizes towards which crossover pushes a population of unary, binary, ternary, etc. GP trees is a Lagrange distribution of the second kind. This result generalises results previously reported in [12,15,13,14].

Naturally, there are important consequences of this result. As was done in [12], we can now compute, for example, the expected resampling probability for programs of different sizes. In particular, let us imagine that our GP system operating on a flat landscape is at the fixed point distribution and let $\mathcal{F}$ and $\mathcal{T}$ be the sizes of the function and terminal sets, respectively. Since, there are $\mathcal{F}^n\mathcal{T}^{(a-1)n+1}$ different programs with n internal nodes in the search space, it is now possible to compute the average probability $p_{\text{sample}}(n)$ that each of these will be sampled by standard crossover, namely

$$p_{\text{sample}}(n) = \frac{(1 - ap_a)}{\mathcal{F}^n\mathcal{T}^{(a-1)n+1}} \binom{an+1}{n} (1 - p_a)^{(a-1)n+1} \, p_a^n. \tag{35}$$

It is easy to study this function and to conclude that, for a flat landscape, standard GP will sample a particular short program much more often than it will sample a particular long one. For example, when $\mu_0 = 27.26044$ as in Figures 1 and 4, GP will heavily resample short programs, e.g., the same program of length 1 is resampled on average every 16 crossovers, every 89 crossovers for programs of length 3, and every 448 crossover for length 5. However, as program size grows the sampling probability drops dramatically. For example, the resampling rate for programs of length 21 is 1 in over 77 million.

In future work we intend to extend the results reported here to the case where primitives of different arities are used in a run. Also, we will attempt to find a mathematical proof that (22) is a fixed-point for (31).

References

1. T. Blickle and L. Thiele. A mathematical analysis of tournament selection. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA'95)*, pages 9–16, San Francisco, California, 1995. Morgan Kaufmann Publishers.
2. T. Blickle and L. Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, 1997.
3. P. C. Consul and L. R. Shenton. Use of Lagrange Expansion for Generating Discrete Generalized Probability Distributions. *SIAM Journal on Applied Mathematics*, 23(2):239–248, 1972.
4. H. Geiringer. On the probability theory of linkage in Mendelian heredity. *Annals of Mathematical Statistics*, 15(1):25–57, March 1944.
5. D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms.* Morgan Kaufmann Publishers, 1991.
6. I. J. Good. The Lagrange Distributions and Branching Processes. *SIAM Journal on Applied Mathematics*, 28(2):270–275, 1975.
7. P. Hilton and J. Pederson. Catalan numbers, their generalization, and their uses. *Mathematical Intelligencer*, 13:64–75, 1991.
8. K. Janardan. Weighted Lagrange Distributions and Their Characterizations. *SIAM Journal on Applied Mathematics*, 47(2):411–415, 1987.
9. K. Janardan and B. Rao. Lagrange Distributions of the Second Kind and Weighted Distributions. *SIAM Journal on Applied Mathematics*, 43(2):302–313, 1983.
10. S. Luke. Two fast tree-creation algorithms for genetic programming. *IEEE Transactions on Evolutionary Computation*, 4(3):274–283, Sept. 2000.
11. T. Motoki. Calculating the expected loss of diversity of selection schemes. *Evolutionary Computation*, 10(4):397–422, 2002.
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 J. F. Miller, M. Tomassini, P. L. Lanzi, C. Ryan, A. G. B. Tettamanzi, and W. B. Langdon, editors, *Genetic Programming, Proceedings of EuroGP'2001*, volume 2038 of *LNCS*, pages 126–142, Lake Como, Italy, 18-20 Apr. 2001. Springer-Verlag.
13. R. Poli and N. F. McPhee. General schema theory for genetic programming with subtree-swapping crossover: Part I. *Evolutionary Computation*, 11(1):53–66, Mar. 2003.
14. R. Poli and N. F. McPhee. General schema theory for genetic programming with subtree-swapping crossover: Part II. *Evolutionary Computation*, 11(2):169–206, June 2003.
15. J. E. Rowe and N. F. McPhee. The effects of crossover and mutation operators on variable length linear structures. Technical Report CSRP-01-7, University of Birmingham, School of Computer Science, Jan. 2001.
16. M. D. Vose. *The simple genetic algorithm: Foundations and theory.* MIT Press, Cambridge, MA, 1999.
17. H. W. Watson and F. Galton. On the Probability of the Extinction of Families. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 4:138–144, 1875.