

Sub-tree Swapping Crossover and Arity Histogram Distributions

Stephen Dignum and Riccardo Poli

School of Computer Science and Electronic Engineering,
University of Essex,
Wivenhoe Park, Colchester, CO4 3SQ, UK
{sandig, rpoli}@essex.ac.uk

Abstract. Recent theoretical work has characterised the search bias of GP sub-tree swapping crossover in terms of program length distributions, providing an exact fixed point for trees with internal nodes of identical arity. However, only an approximate model (based on the notion of average arity) for the mixed-arity case has been proposed. This leaves a particularly important gap in our knowledge because multi-arity function sets are commonplace in GP and deep lessons could be learnt from the fixed point. In this paper, we present an accurate theoretical model of program length distributions when mixed-arity function sets are employed. The new model is based on the notion of an *arity histogram*, a count of the number of primitives of each arity in a program. Empirical support is provided and a discussion of the model is used to place earlier findings into a more general context.

Keywords: Genetic Programming, Sub-Tree Swapping Crossover, Program Length, Arity Histograms.

1 Introduction

Understanding how Genetic Programming (GP) explores the space of computer programs requires two things [12, Chapter 11]: a) characterising the search space itself, e.g., in terms of how fitness is distributed in it, and b) explaining how GP explores it, particularly in terms of the search biases of its genetic operators.

Research on the characterisation of the search space has provided evidence of how program functionality and fitness are distributed in program spaces (e.g., showing that beyond a certain minimum program length the distributions of program functionality and, therefore, fitness converge to a limit [6–9]). Research has also characterised the search bias of GP sub-tree swapping crossover in terms of program length distributions, providing an exact fixed point for trees with internal nodes of identical arity [11, 13] and an approximate fixed point for the mixed-arity case [1, 2].

Understanding the sampling of program length is of particular importance to GP. For example, this has the potential to shed light on the phenomenon of bloat (see [12, Chapter 11] for a recent survey on the topic). Indeed, the work on the fixed-point distributions of program lengths under sub-tree crossover mentioned above has led to a new bloat theory – *crossover-bias* – a number of suggestions for experimental parameter

selection and a new method to control sampling, by length, of GP operators – *operator equalisation*.

One crucial question still open is how to exactly model the limiting distribution of program lengths when mixed-arity function sets are employed. This is particularly interesting because multi-arity function sets are commonplace in GP and important lessons could be learnt from knowing the fixed point.

Note, this is not just a question of better accuracy. Until now, it has not been possible to explain a number of strange empirical findings for mixed arity representations. Why, for instance, are programs with certain compositions of primitives much more likely to be sampled than programs with another composition even if both have exactly the same length? Also, why is it that for smaller lengths, the empirical limiting distribution of program lengths shows a rugged zigzagging profile instead of following the smooth descent, with each succeeding length class sampled with less frequency, seen with common arity function sets? What we need is a model that can explain these phenomena. In this paper we present an accurate theoretical model of program length distributions for mixed-arity function sets which does just that.

The paper is organised as follows. In section 2, we describe a number of models for the prediction of program length based on the repeated application of GP sub-tree swapping crossover, with uniform selection of crossover points, on a flat fitness landscape, i.e., to determine the bias of this operator by removing all other effects. In Section 3, we extend this work and use a number of mathematical generalisations to produce a new model to predict individual occurrence in a population using arity histograms. This is then used to model length class frequencies exactly. Strong empirical evidence is provided in Section 4 to support both models; in particular we show how the length model can be successfully fitted to shorter length classes for mixed arity cases. In Section 5, we discuss the sampling implications of the models and their relationship to the work presented previously in this area notably its implications for program length sampling and GP bloat. Finally, we summarise our findings in Section 6.

2 Background

In [11], a number of models were proposed to predict a limiting distribution of GP tree sizes when sub-tree swapping crossover, with uniform selection of crossover points, was applied on a flat fitness landscape. The limiting distribution of internal nodes for a -ary trees, those whose internal functions have a common arity, a , was shown to be the following Lagrange distribution of the second kind:

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

where $\Pr\{n\}$ is the probability of selecting a tree with n internal nodes and a is the arity of functions that can be used in the creation an individual. The parameter p_a was shown to be related to a and the average size of the individuals in the population at generation 0, μ_0 , according to the formula:

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

In [1], Equation (1) was generalised for mixed arity cases with an average internal arity \bar{a} replacing a . \bar{a} can be predicted from experimental parameters for traditional initialisation methods such as GROW and FULL [10] or determined at run time by calculating the average internal arity at generation 0. The Gamma function, Γ , was used to redefine the binomial coefficient (factorials are replaced using $\Gamma(n+1) = n!$) so that the model could accept non-integer average arity values. This resulted in the following equation:

$$\Pr\{n\} = (1 - \bar{a}p_{\bar{a}}) \frac{\Gamma(\bar{a}n + 2)}{\Gamma((\bar{a} - 1)n + 2)\Gamma(n + 1)} (1 - p_{\bar{a}})^{(\bar{a}-1)n+1} p_{\bar{a}}^n. \quad (3)$$

Note that this equation is also expressed in terms of internal node counts. Strong empirical support was found in [1] for both Equations (1) and (3).

A further generalisation to length classes,¹ i.e., to also include external nodes, or leaves, was found to be successful for a -ary trees in [2]. However, the generalisation to length classes for mixed-arity trees was found to be less successful, being unable to precisely predict the frequency for the smaller length classes, where a smooth descent was predicted by the models but a more rugged shape was found to occur. As an example, Figure 1 shows experimental and predicted results for sub-tree swapping crossover acting on trees with available arities of 1, 2, 3 and 4 (the experimental set-up is as described in section 4).

In related work [4], empirical evidence was provided to suggest that the probability of the occurrence of an individual in a GP population after repeated application of sub-tree swapping crossover on a flat fitness landscape would be determined by the individual's *arity histogram* – a count of the number of nodes in a tree of each arity (see Figure 2 for an example). Within length classes, programs with certain arity histograms were more likely to be found than others. However, within arity histogram classes there is no bias to sample certain program shapes, indicating that arity histograms represent the lowest level of granularity at which length-related biases occur in the presence of sub-tree swapping crossover.²

In the following sections we embrace the idea that if we are to exactly predict length distributions for mixed-arity cases, we will have to incorporate arity histograms in our models. For a -ary trees the arity histogram is, of course, simply the associated internal and external node counts, which explains the earlier success with the a -ary models.

3 Arity Histogram Model

From the work described the previous section, we know that we wish to predict the probability of occurrence of an individual with a particular arity histogram. If we choose n_a to represent a count of arity a nodes, we can define a particular arity histogram of an individual, as the tuple $(n_0, \dots, n_{a_{max}})$. Note, n_0 , is the number of leaves, i.e., nodes with an arity of zero. Using our new notation we can term our target probability, $\Pr\{n_0, \dots, n_{a_{max}}\}$.

¹ Length was derived using the relation $\ell = an + 1$.

² Note that unique programs are a subclass of program shapes which are a subclass of arity histogram classes which are in turn a subclass of program lengths.

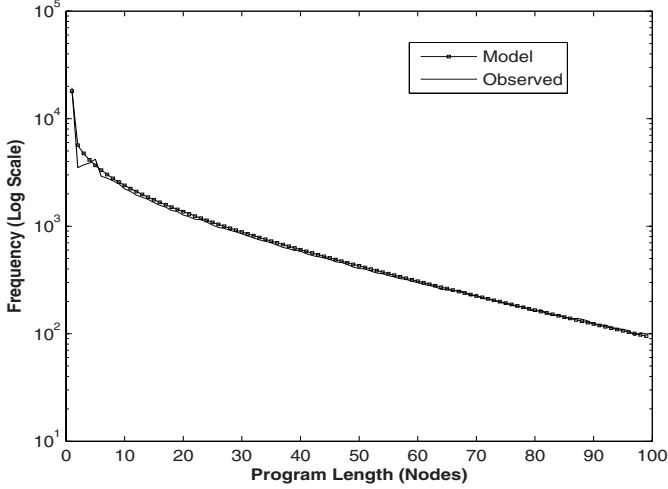


Fig. 1. Comparison between model ($\bar{a} = 2.5$) and empirical program length distributions for trees created with arity 1, 2, 3 and 4 functions and terminals only, initialised with FULL method (depth = 3, initial mean size $\mu_0 = 25.38$, mean size after 500 generations $\mu_{500} = 23.72$). Population size = 100,000.

Below, we will attempt to identify this function by means of generalisation from previous results and intuition. The ‘acid test’ for the result of our generalisation will be whether or not it fits the empirical data in a variety of conditions.

Let us start by reviewing Equation (1), the original model for a -ary representations. We can see that in order to generalise it, we need to introduce the concept of multiple arities, particularly the associated p_a and n_a values.

First, we postulate that we now have a set of p_a values each associated with a single arity. If we interpret these as forming a probability distribution, we can then imagine that product ap_a in the first term of the equation, actually represents an ‘expectation’ of a .³ If this is correct, then the first term $(1 - ap_a)$ should be changed to $(1 - \sum_{a \geq 1} ap_a)$.

The original binomial coefficient term represents the number of ways of choosing internal nodes of the same arity, a , from the length of the resulting tree, $an + 1$. We need to alter this by selecting each arity count, n_a , from the tree length that can be built with this collection of arities, $\sum_{a \geq 1} an_a + 1$. Our binomial coefficient term, therefore, becomes the multinomial coefficient $\binom{\sum_{a \geq 1} an_a + 1}{n_0, \dots, n_{a_{max}}}$, where n_0 is the count of leaves, n_1 is the count of the functions with arity 1, etc.

The third term, $(1 - p_a)^{(a-1)n+1}$, can be broken into two parts. The superscript is simply the number of terminals for the tree, which we know to be n_0 . As with the first term we alter $(1 - p_a)$ to a mixed arity equivalent, which we postulate to be $(1 - \sum_{a \geq 1} p_a)$.

³ In our a -ary model: $E[a] = 0 \times (1 - p_a) + a \times p_a = ap_a$.

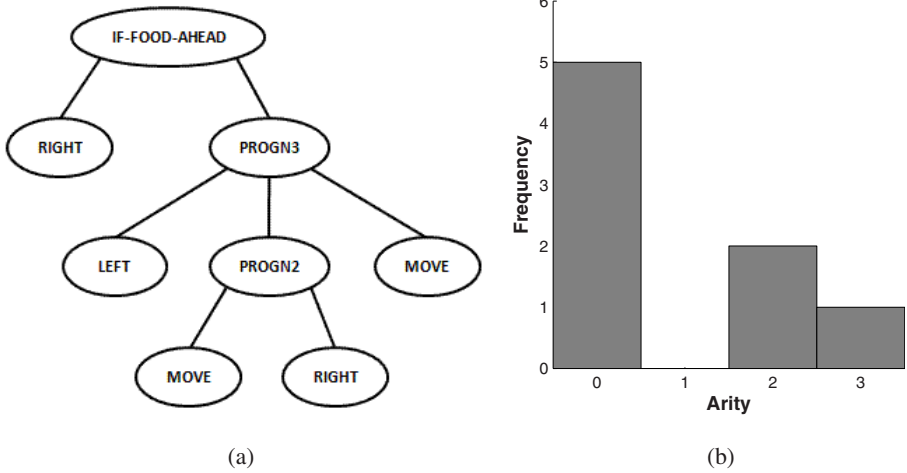


Fig. 2. A proposed solution for the Artificial Ant problem [5] (a) and its associated arity histogram (b)

Continuing this analogy, the final term, p_a^n , represents the value, p_a , to the power of the number of nodes, n . We need to now split out the term so that each value of p_a is associated with the appropriate n_a value. The most natural way to do this is to turn the final term into the product $\prod_{a \geq 1} p_a^{n_a}$.

Putting this altogether, we obtain our mixed-arity model for the limiting distribution of arity histograms created by sub-tree swapping crossover:

$$\Pr\{n_0, \dots, n_{a_{max}}\} = (1 - \sum_{a \geq 1} a p_a) \binom{\sum_{a \geq 1} a n_a + 1}{n_0, \dots, n_{a_{max}}} (1 - \sum_{a \geq 1} p_a)^{n_0} \prod_{a \geq 1} p_a^{n_a}. \quad (4)$$

This equation has now become a multivariate Lagrange distribution of the second kind. Note, the introduction of counts for program leaves will only affect the second and third terms. On closer inspection we can also see that there is in fact no need to calculate p_0 .⁴

Next, we need to create a model that will turn arity histogram probabilities into those of length classes. The set of arity histograms that represent a particular program length ℓ can be defined as:

$$\left\{ n_0, \dots, n_{a_{max}} : \sum_{a \geq 1} a n_a + 1 = \ell \right\}. \quad (5)$$

We can, therefore, calculate the probability of a particular program length by summing the probabilities for each of the associated arity histograms, i.e.,

$$\Pr\{\ell\} = \sum_{n_0, \dots, n_{a_{max}} : \sum_{a \geq 1} a n_a + 1 = \ell} \Pr\{n_0, \dots, n_{a_{max}}\}. \quad (6)$$

⁴ If we define p_0 to be $1 - \sum_{a \geq 1} p_a$ and allow the fourth term to run from $a = 0$, we could also omit the third term.

We term this a *Lagrange distribution of the third kind*. The formula clarifies that the length bias with which sub-tree swapping crossover samples program spaces is, in fact, the result of an even more primitive bias associated with arity histograms.

In the next section we will provide empirical support for our new models of arity-histogram and length distributions, to ensure that they continue to predict a -ary representation length distributions and will now accurately model mixed arity representations.

4 Empirical Validation

In order to verify empirically the models proposed, a number of runs of a GP system in Java were performed. A relatively large population of 100,000 individuals was used in order to reduce drift of average program size and to ensure that enough programs of each length class were available. The FULL initialisation method was used with non-terminals being chosen with uniform probability. Each run consisted of 500 generations. All results were averaged over 20 runs.

To check if the models presented in the previous section match experimental data we need to fit them to the data so as to identify the parameters p_a . This fit was achieved using a hill climber search program that reduced the mean squared error from that observed in the final generation and that predicted by the theoretical distribution, by altering the p_a values.⁵

Our first step was to see if there is evidence that arity histogram occurrence is modeled correctly. In Figure 3, we can see two views of the modeled and empirical data for each experiment. The X-Y plots on the left report the frequency for each arity histogram predicted via Equation (4) vs the corresponding empirical frequency. Note how the data points lie on, or very close to, the diagonal line that represents perfect prediction. Each point in the scatter plots on the right shows either the actual or the predicted frequency for an arity histogram vs the length class it corresponds to. The multiple points at each length are the elements of the set in Equation (5). Of particular interest is that even with a relatively large population size, certain histograms are exceptionally rare. For example, occurrences for a histogram consisting of arity one functions and a single terminal for the 1 & 3 arity experiment, are predicted to be less than 1 by the time we reach a length of 9 nodes and far less in the other experiment.

As we can see in Figures 4 and 5, the model in Equation (6) fits very well the frequencies associated to all length classes for mixed arities. Note in particular how the model, that incorporates the arity histogram model from Equation (4), now captures the fluctuating early values for mixed arity representations.

In order to confirm that Equation (6) is in fact a generalisation of earlier work and accurately predicts a -ary distributions, Figures 6 and 7 show the model and observed data from the final generation for 1-ary and 2-ary trees. In this case the p_a values for the model were calculated using Equation (2).

In essence, we now have evidence that we have isolated the fundamental components of the limiting length distribution for sub-tree swapping crossover. Further work is

⁵ Initial values of p_a were set uniform randomly between 0 and 0.2. A number of runs were performed and best results were found using small variations (less than 0.001%) with a high number of alternatives at each step (typically 100).

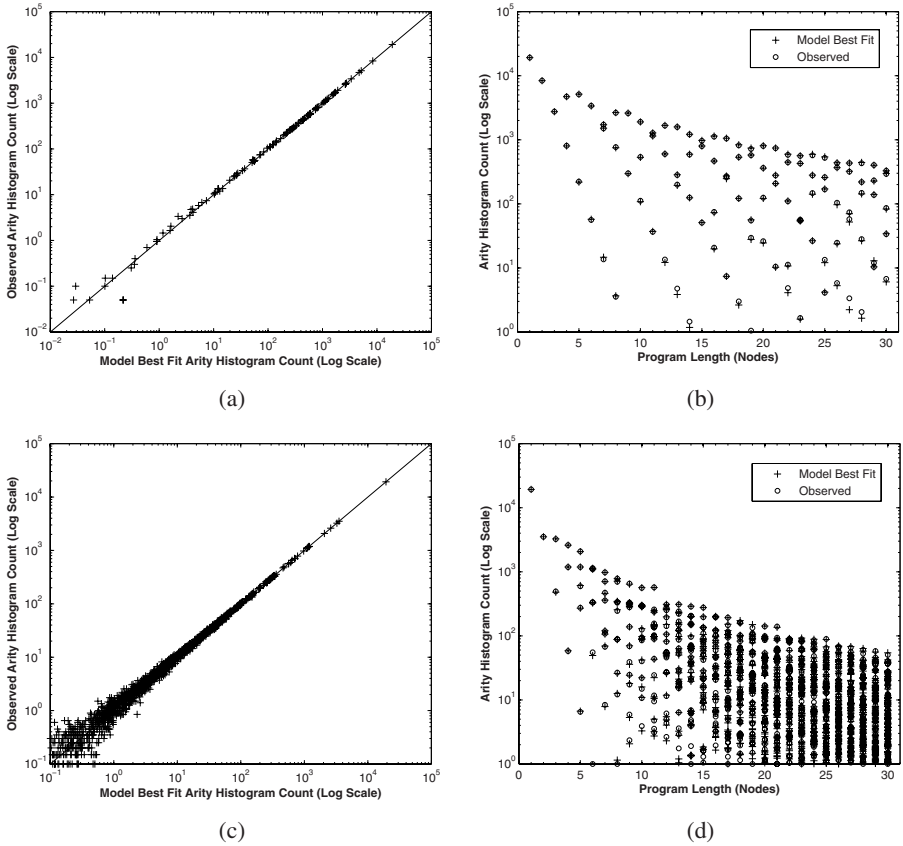


Fig. 3. Comparison between unique arity histogram count observations and arity histogram model predictions obtained by best fit for trees created with arity 1 and 3 functions (a) & (b), arity 1, 2, 3 and 4 functions (c) & (d), and terminals only. Experimental parameters as described in figures 4 and 5 respectively. Diagonal lines added to (a) & (c) represent perfect prediction.

required to make this a complete predictive model, i.e., we need a formula to determine p_a values for mixed arity representations. However, we can now place the findings from earlier work in this area into further context. This is discussed in the next section.

5 Sampling Implications

From our analysis we can now be confident in the assertion that the limiting distribution of program lengths for a GP population after the repeated application of sub-tree swapping crossover, with uniform selection of crossover points, on a flat fitness landscape, is determined solely by the mix of node arities in the initial population.

From the work provided in [4], we know that there is empirical evidence to show that there is no bias for sub-tree swapping crossover to place a particular node label at any

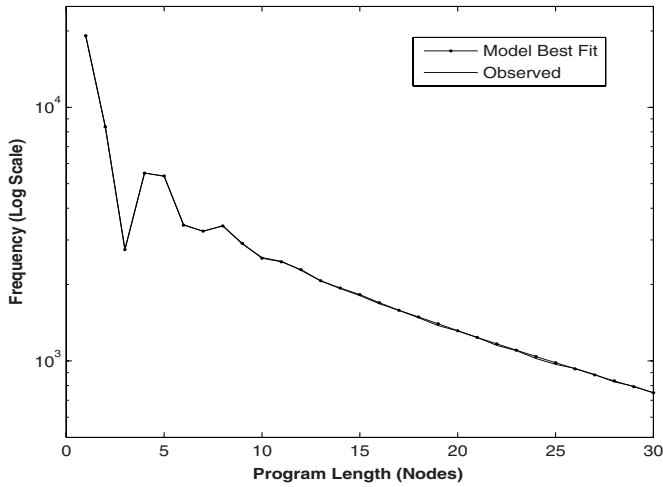


Fig. 4. Comparison between empirical length distributions and an arity histogram model obtained by best fit for trees created with arity 1 and 3 functions and terminals only, initialised with FULL method (depth = 3, initial mean size $\mu_0 = 15.00$, mean size after 500 generations $\mu_{500} = 15.75$). Population size = 100,000. $p_1 = 0.2186684078761787$, $p_3 = 0.15804781356057954$.

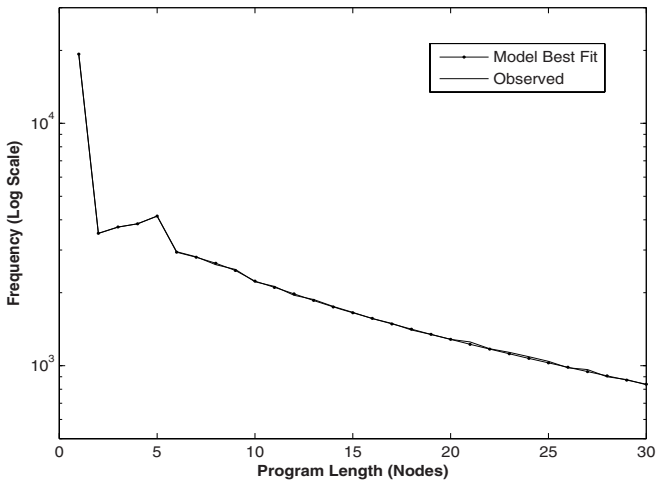


Fig. 5. Comparison between empirical length distributions and an arity histogram model obtained by best fit for trees created with arities 1, 2, 3 and 4 functions and terminals only, initialised with FULL method (depth = 3, initial mean size $\mu_0 = 25.38$, mean size after 500 generations $\mu_{500} = 23.72$). Population size = 100,000. $p_1 = 0.09117030091320417$, $p_2 = 0.08112567496250808$, $p_3 = 0.0702296050436014$, $p_4 = 0.0643780797663895$.

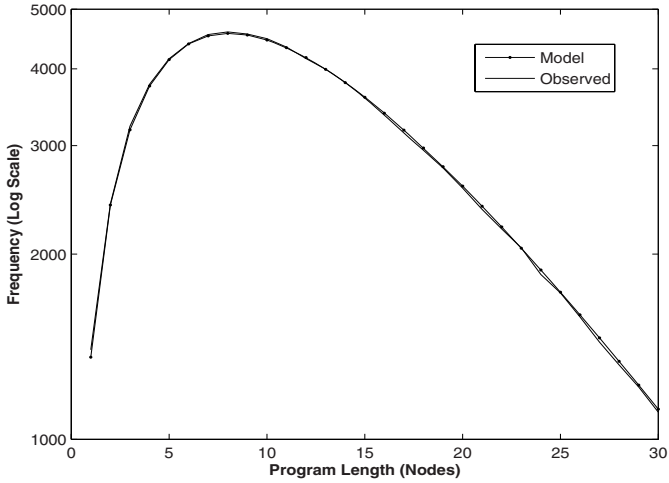


Fig. 6. Comparison between empirical length distributions and an arity histogram model created with arity 1 functions and terminals only, initialised with FULL method (depth = 15, initial mean size $\mu_0 = 16.00$, mean size after 500 generations $\mu_{500} = 16.15$). Population size = 100,000.

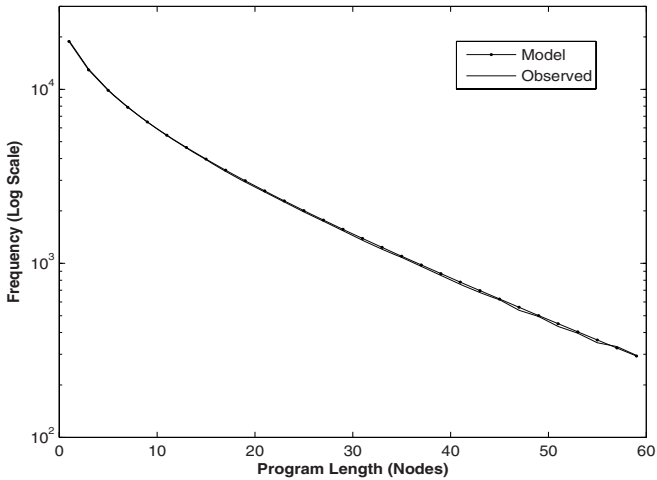


Fig. 7. Comparison between empirical length distributions and an arity histogram model created with arity 2 functions and terminals only, initialised with FULL method (depth = 3, initial mean size $\mu_0 = 15.00$, mean size after 500 generations $\mu_{500} = 14.19$). Invalid even lengths are ignored. Population size = 100,000.

position in a tree. All programs with a particular arity histogram are, therefore, equally likely to be sampled by the application of sub-tree swapping crossover in the absence of other operators. By extension, we can also say that all programs of a certain length are equally likely to be sampled for a -ary trees; this is not true, however, for mixed arity representations. If one wishes to ensure uniform sampling within length classes, alternative variation operators will need to be devised when mixed arity representations are employed with sub-tree swapping crossover.

Looking more closely at Equation (4), we can see that the first term will remain constant for all arity histograms whilst the second term, the multinomial coefficient, will increase the probability for arity histograms that can produce more shapes. The third and final terms decrease rapidly with increasing values of the n_a 's producing the eventual smooth curve. Therefore, arity histograms presented to Equation (6), that can produce more shapes than other arity histograms in a particular length class, will have a higher probability of being sampled within that class.

Disregarding the fluctuations shown in earlier length classes for mixed arity classes, Equation (6) is decreasing. The *crossover bias* bloat theory was originally proposed based upon evidence presented by the internal node count models and their decreasing nature, to recap:

- I In each generation selection populates the mating pool with relatively fit programs
- II The sub-tree swapping crossover operator will then produce children with a length distribution biased towards smaller programs irrespective of their fitness⁶
- III If smaller programs cannot obtain a relatively high fitness, which after the initial generations of any non-trivial GP problem is highly likely, they will be ignored by selection in the next generation
- IV Hence, average program size will increase as ever larger programs are placed into the mating pool

Equation (6) and the empirical work provided in Section 4 provide extra evidence to support this theory for sub-tree swapping crossover, in that our more pertinent length model varies only slightly from the smooth descent described for the internal node models presented in Section 2 and used as the basis for the theory in [1]. One can argue that any variation operator that has a bias towards smaller programs will cause bloat in this way and the theory should be renamed to *operator length bias*.

Recent work by Soule [15] has shown empirical evidence of a similar sampling effect for other types of crossover, in addition particular emphasis is placed on variations to sampling caused by altering the nature of the fitness landscape. Future research may look into the speed of convergence to Lagrangian type distributions for different variation operator and problem combinations, i.e., to enable us to gauge which experimental set-ups are likely encounter bloat earlier during an experimental run.

The internal node and length a -ary models presented in Section 2 can be used as predictive models without modification. The length model for mixed arity trees developed

⁶ It is important to note that there is no change in the average size of programs found in the mating pool from those produced in the resulting child population, i.e., the next generation. However, the distribution has a sampling bias towards smaller programs, with relatively few larger programs.

there remains a strong model for approximation if an exact fit for earlier length classes is not required. One could, for example, use this to implement broad structural convergence measures suggested in [4]. If a more exact model was required, a fit to internal node counts could be used.

The generalised mixed arity internal node model (Equation (3)) is also an interesting starting point to further analyse the arity histogram model proposed here. We can ask how was such a generalised model so successful when only leaves were removed from the investigation? For example, would Equation (6) collapse to Equation (3) with further analysis? This is left to future work.

Finally, recent work using length based *operator equalisation* methods [3, 14], i.e., those that modify selection probabilities according to current and desired length distributions, do not guarantee uniform sampling of unique programs within the length classes desired.⁷ One could imagine an extension to the method, however, to sample uniformly within length classes by storing a histogram of arity histograms, possibly using a hashing function as lengths increase. Indeed, as an alternative, it is possible to design an *arity histogram equaliser* to ensure that certain, desired, distributions of arity counts are selected.

6 Conclusions

In this paper we have generalised the Lagrange distribution of the second kind that has recently been shown to represent the limit program length distributions for sub-tree swapping crossover in the presence of single-arity function sets to the important and much more common multi-arity case.

The generalisation has required to express this fixed point via the use of arity histograms which effectively generalise the internal node counts and average arities used in prior work. Arity histograms are the fundamental components of GP sub-tree swapping crossover with regard to program sampling.

This model has allowed us to understand a number of sampling effects and to accurately model not just the smooth descending curves of the internal node models but also those of the more rugged true length distributions, i.e., those that also include leaves. From this, we can now place a number of earlier findings into a more general context. We can also start making use of our new knowledge about the biases of crossover. For example, we can confirm hypotheses about bloat and generalise cures for it.

References

1. Dignum, S., Poli, R.: Generalisation of the limiting distribution of program sizes in tree-based genetic programming and analysis of its effects on bloat. In: Thierens, D., Beyer, H.-G., Bongard, J., Branke, J., Clark, J.A., Cliff, D., Congdon, C.B., Deb, K., Doerr, B., Kovacs, T., Kumar, S., Miller, J.F., Moore, J., Neumann, F., Pelikan, M., Poli, R., Sastry, K., Stanley, K.O., Stutzle, T., Watson, R.A., Wegener, I. (eds.) GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation, July 7-11, vol. 2, pp. 1588–1595. ACM Press, New York (2007)

⁷ Unless only an a -ary representation is used.

2. Dignum, S., Poli, R.: Crossover, sampling, bloat and the harmful effects of size limits. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 158–169. Springer, Heidelberg (2008)
3. Dignum, S., Poli, R.: Operator equalisation and bloat free GP. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 110–121. Springer, Heidelberg (2008)
4. Dignum, S., Poli, R.: Sub-tree swapping crossover, allele diffusion and GP convergence. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 368–377. Springer, Heidelberg (2008)
5. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
6. Langdon, W.B.: Convergence rates for the distribution of program outputs. In: Langdon, W.B., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A.C., Miller, J.F., Burke, E., Jonoska, N. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002, New York, July 9-13, pp. 812–819. Morgan Kaufmann Publishers, San Francisco (2002)
7. Langdon, W.B.: How many good programs are there? How long are they? In: De Jong, K.A., Poli, R., Rowe, J.E. (eds.) Foundations of Genetic Algorithms VII, Torremolinos, Spain, September 4-6, 2002, pp. 183–202. Morgan Kaufmann, San Francisco (2003)
8. Langdon, W.B.: Convergence of program fitness landscapes. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1702–1714. Springer, Heidelberg (2003)
9. Langdon, W.B., Poli, R.: Foundations of Genetic Programming. Springer, Heidelberg (2002)
10. Luke, S.: Two fast tree-creation algorithms for genetic programming. *IEEE Transactions on Evolutionary Computation* 4(3), 274–283 (2000)
11. Poli, R., Langdon, W.B., Dignum, S.: On the limiting distribution of program sizes in tree-based genetic programming. In: Ebner, M., O'Neill, M., Ekárt, A., Vanneschi, L., Esparcia-Alcázar, A.I. (eds.) EuroGP 2007. LNCS, vol. 4445, pp. 193–204. Springer, Heidelberg (2007)
12. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming (with contributions by J.R. Koza) (2008), <http://lulu.com>, <http://www.gp-field-guide.org.uk>
13. Poli, R., McPhee, N.F.: 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: Miller, J., Tomassini, M., Lanzi, P.L., Ryan, C., Tetamanzi, A.G.B., Langdon, W.B. (eds.) EuroGP 2001. LNCS, vol. 2038, pp. 126–142. Springer, Heidelberg (2001)
14. Silva, S., Dignum, S.: Extending operator equalisation: Fitness based self adaptive length distribution for bloat free GP. In: Vanneschi, L., Gustafson, S., Moraglio, A., De Falco, I., Ebner, M. (eds.) EuroGP 2009. LNCS, vol. 5481, pp. 159–170. Springer, Heidelberg (2009)
15. Soule, T.: Crossover and sampling biases on nearly uniform landscapes. In: Riolo, R.L., Soule, T., Worzel, B. (eds.) Genetic Programming Theory and Practice VI, Genetic and Evolutionary Computation, May 15-17, ch. 6, pp. 75–91. Springer, Ann Arbor (2008)