Chapter 18
Modelling Genetic Programming as a Simple Sampling Algorithm

David R. White, Ben Fowler, Wolfgang Banzhaf, Earl T. Barr

Abstract This chapter proposes a new model of tree-based Genetic Programming (GP) as a simple sampling algorithm that samples minimal schemata (subsets of the solution space) described by a single concrete node at a single position in the expression tree. We show that GP explores these schemata in the same way across three benchmarks, rapidly converging the population to a specific function at each position throughout the upper layers of the expression tree. This convergence is driven by covariance between membership of a simple schema and rank fitness. We model this process using Price's theorem and provide empirical evidence to support our model. The chapter closes with an outline of a modification of the standard GP algorithm that reinforces this bias by converging populations to fit schemata in an accelerated way.

Key words: Genetic programming, Schema theory, convergence, covariance, Price's theorem

David R. White
Dept. of Physics, University of Sheffield, Sheffield, UK e-mail: d.r.white@sheffield.ac.uk

Ben Fowler
Dept. of Computer Science, Memorial University of Newfoundland, St. John's, NL, Canada e-mail: fowler@mun.ca

Wolfgang Banzhaf
Dept. of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA e-mail: banzhafw@msu.edu

Earl T. Barr
CREST, University College London, London, UK e-mail: e.barr@ucl.ac.uk

18.1 Introduction

Previous attempts to characterise Genetic Programming (GP) have focused on complex schemata, that is 'templates' of full GP trees or subtrees composed of fixed positions that specify the presence of a given function from the function set, and other unspecified 'wildcard' positions that indicate any single function, or an arbitrary subtree; a schema represents a set of expression trees that share some syntactic characteristics. Unfortunately, the complicated definitions of such schemata has made resulting theories difficult to test empirically, and GP schema theory has been criticised for its inability to make significant predictions or inform the development of improved algorithms.

We present a dramatically simplified approach to GP schema theory, which considers only the distribution of functions at a single node in the tree, ignoring second-order effects that describe the context within which the node resides: for example, we do not consider parent-child node relationships. Our schemata are thus simple fixed-position 'point schemata', as illustrated in Figure 18.1. We show that the behaviour of GP viewed through the sampling of simple schemata is remarkably consistent: GP converges top-down from the root node, with nodes at each level gradually becoming 'frozen' to a given function from the function set, in a recursive fashion; an example of such convergence for the root node on one benchmark is illustrated in Figure 18.2. We model this behaviour as a competition between simple schemata, where the correlation between rank fitness and schema membership determines the 'winner' and drives convergence. We present empirical evidence to support our model.

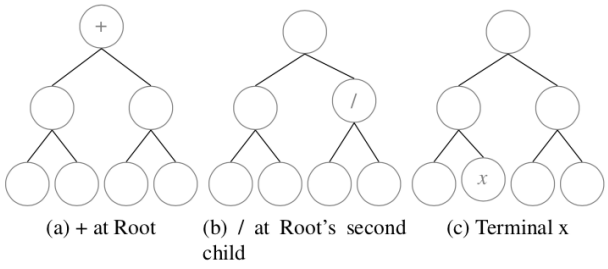

Fig. 18.1: Examples of Simple Single Point Schemata for a function set where the maximum arity is two. A single labelled node indicates a fixed position; all other nodes are unconstrained — provided that they are not on the path from root to the fixed node, a position may not even exist within trees belonging to the schemata.

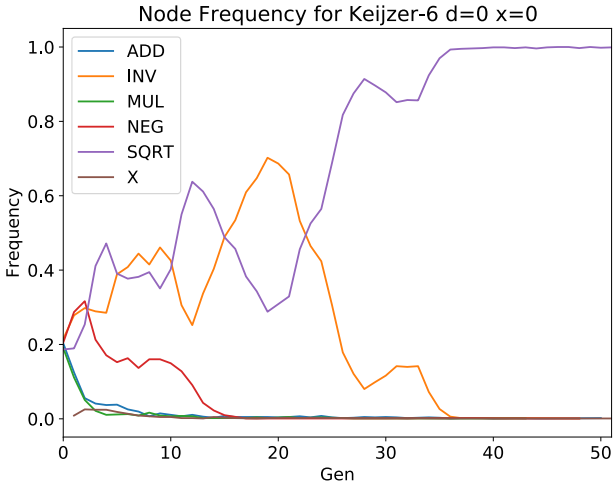

Fig. 18.2: Example convergence of the root node across a population for the Keijzer-6 benchmark. The root node distribution converges to the Sqrt function. Other runs exhibit similar behaviour.

18.2 Rationale for modelling Simple Schemata

Traditional GP, often referred to as tree-based GP, is a simple evolutionary algorithm that explores the space of expression trees. Algorithm 6 provides pseudocode for canonical tree-based GP using tournament selection. Two key observations from this pseudocode are apparent: (1) the variation operators applied *do not consider the semantics of the parents or their children* (see also [7]), and (2) the only point at which fitness is considered by the algorithm is *in selection via relative fitness, i.e. ranking*. We add a third observation from the wider GP literature: (3) GP's strength lies in its versatility: it may easily be applied to a wide variety of problems and demonstrates good performance across a range of problem domains.

Given these observations, any description of how GP searches the space of expression trees is limited to considering (a) syntactic properties of the population and (b) the rank fitness of individuals. Beyond relative fitness within a population, the semantics of individual trees is irrelevant. The building block hypothesis [8] (BBH) conjectures that GP crossover exchanges small subtrees that make above-average contributions to the fitness of an individual, despite GP's disregard for semantic concerns when selecting the location at which that subtree is inserted. That such subtrees should exist seems intuitively unlikely, but far more concerning is the notion that such subtrees should exist across the wide variety of problems GP has been successfully applied to. We are therefore skeptical of the BBH and regard crossover as a constrained macro-mutation operator; this viewpoint is congruent with rigorous

empirical studies that have shown crossover to be beneficial on only a limited subset of simple problems [15].

We therefore must consider only syntactic features when formulating a model of GP's behaviour, and furthermore consider only those features that are represented in a substantial number of individuals within the population, arriving at the following possible properties that may form part of a behavioural model for GP:

- Tree size
- Tree shape
- Function frequency per individual
- Functions at positions in the tree (simple schemata)
- Higher-order schemata

Tree size and shape depend on the level of bloat in a population; by definition, bloat does not contribute to solving a particular problem, so we disregard tree size and shape as an explanatory factor. Explorative data analysis of *population* function frequency showed some interesting behaviour, but it was too coarse-grained to maintain a strong relationship with semantics.

We are therefore left with simple and higher-order schemata. Given the limited success of higher-order schema theory, we restrict the discussion to models based on function positions within individual trees. Our preliminary investigations into the frequency of different functions at the root node showed promise, and also matched earlier results [12], which considered more complex rooted schemata.

We retain a simpler definition of point schemata; our definition utilises the grid system introduced by [10], which is illustrated in Figure 18.3. Using this coordinate system, we can formally define a simple schema:

Definition: A simple GP schema is a tuple (d, x, f) where d is the depth within a tree (root is defined as depth 0), x is the 'x index' identifying a node at that particular depth as in [10], and f is a function from the function set.

Examples of simple schemata are given in Figure 18.1.

18.3 Modelling GP

We are interested in modelling the frequency of simple schemata within the population; this is a heritable trait and can therefore be modelled using Price's covariance and selection theorem [11], which gives the expected change in a trait within a population:

$$\triangle z = \frac{1}{\bar{w}} cov(w_i, z_i) + \frac{1}{\bar{w}} E(w_i \triangle z_i) \tag{18.1}$$

Here, z is the prevalence of a trait within the population, $\triangle z$ is its expected change, $\bar{w}$ is the average number of children per individual, z_i is the amount of the trait within the individual (in our case, 1 if the individual belongs to a schema,

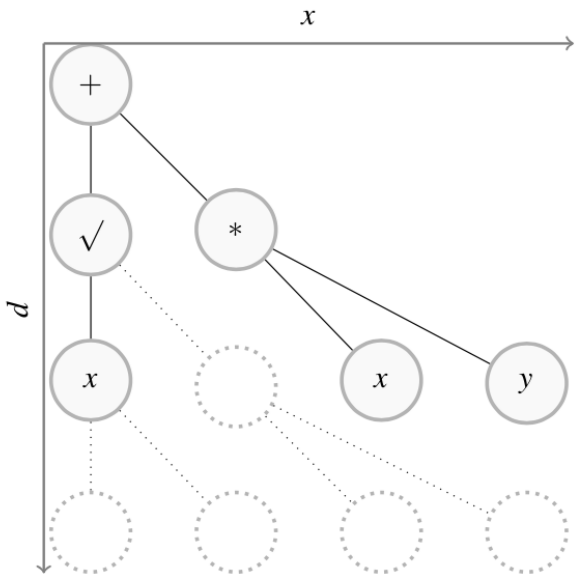

Fig. 18.3: Grid system for laying out expression trees, based on Poli and McPhee [10]. The example represents the expression tree $\sqrt{x}+xy$.

and 0 otherwise), and w_i is the number of children of individual i. As our population size is constant, the average fitness (number of children) of an individual is always one and thus the $1/\bar{w}$ may be discarded. In our case, z is the frequency of schema membership, i.e., the proportion of individuals belonging to a given schema.

The expected change in trait z over the population depends on two terms: the change due to selection and the change due to transmission. The change due to selection is determined by the covariance between possession of that trait and fitness that propagates the trait into the next generation, whilst the change due to transmission models the constructive and destructive effects of genetic operators in GP, which may add or remove an individual's schema membership.

18.3.1 Change in Schema Prevalence due to Selection

We first consider change due to selection. We model only tournament selection, a common selection method, which uses relative (ranked) fitness to select parents. After fitness evaluation, each individual is assigned a rank $r_1 \ldots r_N$ where r_1 is the best individual and r_N is the worst. As given by [2] the probability $p_s(i)$ of selecting

Algorithm 6 Canonical tree-based GP Algorithm

Input: $pop\_size, gens, p_{xo}, p_{mut}$
Output: Final population pop

```
1:  pop ← init_pop(pop_size)
2:  for 1 to gens do
3:      for p ∈ pop do
4:          eval(p)
5:      end for
6:      next_gen ← {}
7:      while size(next_gen) < pop_size do
8:          op ← select_op(xo, mut, p_xo, p_mut)
9:          if op == mut then
10:             ind ← tournament_selection(pop, 1)
11:             next_gen ← next_gen ∪ {mutate(ind)}
12:         end if
13:         if op == xo then
14:             ind ← tournament_selection(pop, 2)
15:             next_gen ← next_gen ∪ {xo(ind, ind2)}
16:         end if
17:     end while
18: end for
19: return pop
```

an individual i of rank r_i within a population size of n under a tournament of size k is given by:

$$p_s(r_i) = \frac{(n-i+1)^k - (n-i)^k}{n^k} \tag{18.2}$$

Any selected individual has a single child belonging to the same schema, modulo the opportunity for disruption due to the genetic operators as described below. We define $member(h, r_i)$ to be 1 if the individual ranked r_i is a member of h, and 0 otherwise, We therefore rewrite Equation 18.1, inserting our function and notation for schema membership:

$$\triangle|h| = cov(p_s(r_i),\ member(h, r_i)) + E(w_i \triangle z_i) \tag{18.3}$$

We can precisely calculate this term in practice, although we require full knowledge of schema membership for each fitness rank. It also represents an expectation, and errors may accumulate if predicting membership several generations ahead: indeed, stochastic variation in the early stages of a run may sometimes push the population away from convergence to the most highly fit schemata to secondary ones.

However, such an equation is still useful if it accurately reflects the mechanism that underlies GP, as we may use it to improve on current algorithms.

18.3.2 Change in Schema Prevalence due to Operators

To model the potential change due to transmission, we must model crossover and mutation. If no crossover or mutation occurs, then the transmission probability is 1 and the case is closed; this is simple reproduction of the parent, and is omitted from algorithm 6 as we do not use it.

First, consider disruption. A single node schema can only be disrupted if the crossover or mutation point selected is above the node. Given a schema (d,x,f), there are only $d-1$ such points in the tree i.e. the potential for disruption occurs with probability $\leq d/size(tree)$, otherwise disruption will certainly not occur.

Assuming the selection of a disruptive node for crossover or mutation, there are three possibile outcomes: (1) the node is replaced with another node of the same type, preserving the schema; (2) the node is replaced with another node of a different type; (3) the node is removed from the tree due to a change in tree shape. The only possibility that preserves the schema is situation (1), and to do so two conditions are required: first, the inserted subtree must be of a shape such that the node exists in the new tree and, second, the node inserted at position (d,x) must be of type f.

If we assume that the distribution of different functions f out of the function set F is roughly uniform in the population (we can alternatively make a slightly weaker but more complicated assumption with the same conclusion), then the probability of disrupting the schema given the node exists in the donating tree is approximately $(|F|-1)/|F|$. This probability is therefore close to 1, and is actually an optimistic lower bound: leaf nodes are disproportionately likely to be inserted, increasing this probability further.

Given that the insertion of a suitable subtree that will preserve a schema is already small, particularly as d grows (because inserted subtrees are disproportionately small), we therefore make a simplifying assumption: that the probability of disruption to the schema given an insertion point between the root and our defining node is approximately 1, and therefore the probability of disruption is $\approx d/size(tree)$. Given growth in tree size we can see why convergence close to the root will occur: the probability of disrupting a schema is low and the selection term in Equation 18.3 will dominate, particularly as average tree size increases.

A similar argument can be used for the constructive case: one of the $(d-1)/size(tree)$ nodes must be selected for insertion, and in this case some trees may not contain a node at that position, so this is an optimistic upper bound. Given such an insertion point is selected, we must then select for insertion at that point a subtree deep enough with the correct shape to ensure the node exists within the schema, and the inserted node must be of the correct type. Again, as average tree size increases the probability of construction rapidly diminishes and we therefore assume the prob-

ability of construction approaches zero as d increases. We conjecture that this is the reason GP systems do not exhibit long-term learning.

18.4 Empirical Data Supporting the Model

Benchmark	Target Equation	Paper
Keijzer-6	$\sum_i^x \frac{1}{i}$	Keijzer [3]
Korns-12	$2-2.1cos(9.8x)sin(1.3w)$	Korns [4]
Vladislavleva-4	$\frac{10}{5+\sum_{i=1}^{5}(x_i-3)^2}$	Vladislavleva et al. [13]

Table 18.1: Benchmarks used

How does GP behave with respect to these schemata? We examined a set of three benchmarks, taken from [14] given in Table 18.1, and plotted the convergence behaviour of these schemata. Convergence plots for various schemata are shown in Figures 18.4 – 18.9. Similar results were found across the deeper layers of the tree, with a large proportion converging to no function, i.e. a tree shape that does not involve nodes at position (d,x).

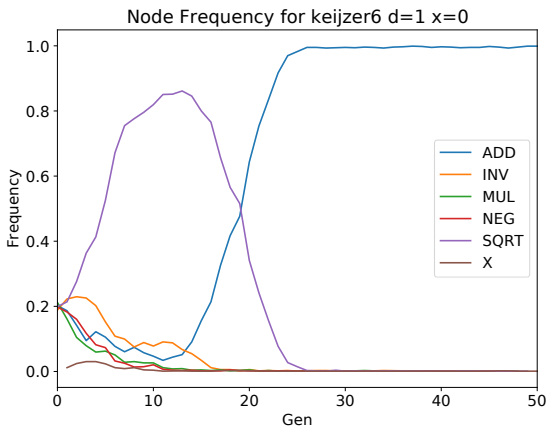

Fig. 18.4: Node Frequency for Keijzer6 d=1 x=0

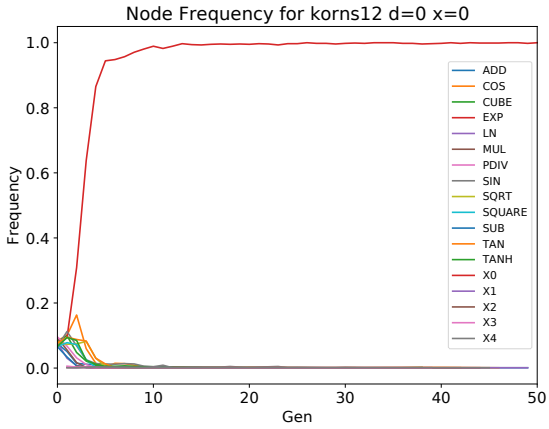

Fig. 18.5: Node Frequency for Korns12 d=0 x=0

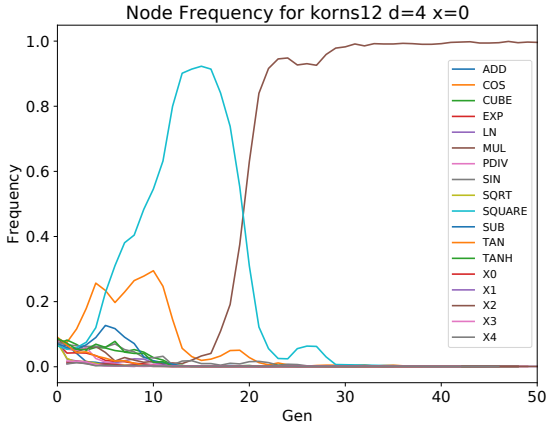

Fig. 18.6: Node Frequency for Korns12 d=4 x=0

There were two main exceptions to this convergence behaviour: firstly, when the sample size for a particular (d,x) coordinate was small, convergence may not be seen for that position. This was particularly likely to occur deep within the node grid and for problems with large node arities, because a given row in the tree grid has a^k x indices, where k is the greatest arity of functions in the function set. The more significant exception was lower levels in the tree in general, where crossover was more likely to impact convergence: crossover was exponentially more likely to impact a schema at a lower position within the tree than at a higher one, and thus

there was a counterbalance to the convergence resulting from rank-based selection. However, the average impact of a node deep in the tree in terms of overall semantics was low, so we did not consider it within our modelling.

Figure 18.10 shows an example of convergence behaviour across runs, where each line is the 'winning' function from a run. It was often the case that two functions demonstrated strong covariance and due to the stochastic nature of GP, convergence to any given function was not certain, but biased towards those with high covariance.

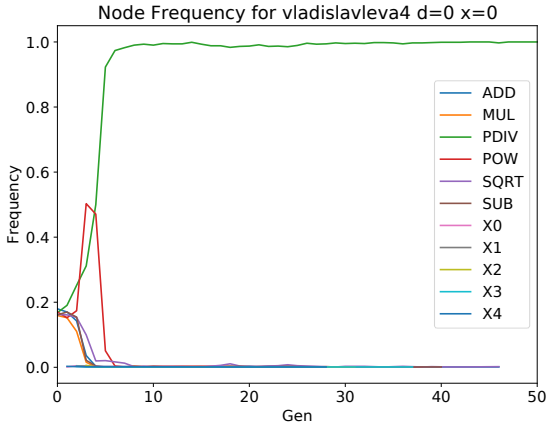

Fig. 18.7: Node Frequency for Vladislavleva4 d=0 x=0

Does covariance between schema membership and rank fitness drive convergence, i.e. is our model accurate? We plot graphs showing both convergence and covariance in Figures 18.11 – 18.13. Schema and frequency are colour-matched, the dashed lines indicating the covariance (right-hand axis). Clear spikes in covariance can be seen prior to population convergence. Note that if we were to anticipate convergence, the best predictor would be covariance, a signal which is clear far in advance of the rising edge of the convergence curve; this suggests it may be possible to exploit covariance measures to accelerate convergence and potentially improve the performance of GP. Note also that over time the clear covariance signal disappears, with covariance tending toward 0. Figure 18.13 shows that after a brief spike in covariance and a subsequent spike in node frequency, all frequencies went to 0 in this example, indicating that node (2,2) is unoccupied.

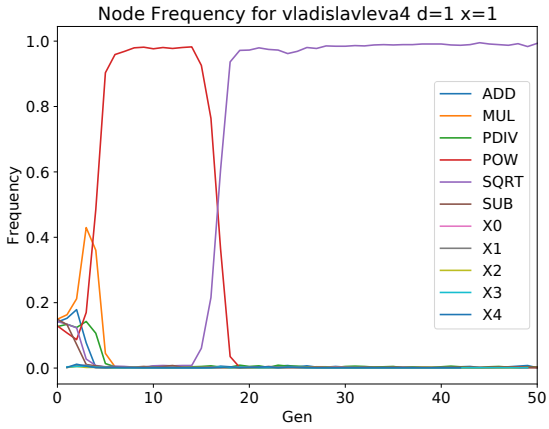

Fig. 18.8: Node Frequency for Vladislavleva4 d=1 x=1

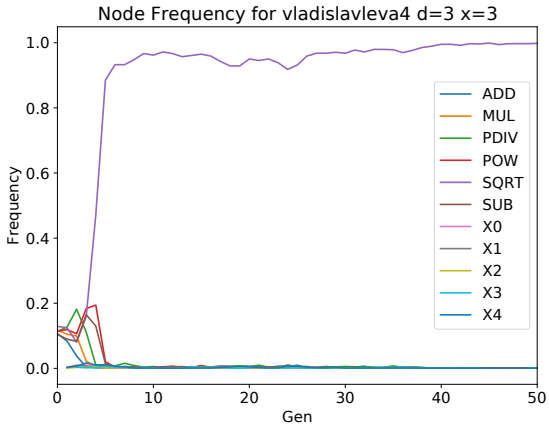

Fig. 18.9: Node Frequency for Vladislavleva4 d=3 x=3

18.5 Ways to improve GP

Based on our model of how GP searches the solution space, we propose a possible improvement to GP: a point schema should be "frozen" if high covariance is found between a given root-node schema and rank fitness. Freezing this node can be achieved by assigning the worst fitness to all individuals in the population that *do not* belong to that schema, forcing convergence. After a new generation has been produced and evaluated, we can check to see if any function dominates the schema: it must dominate both in terms of being substantial in number (e.g. $> 20\%$ of the

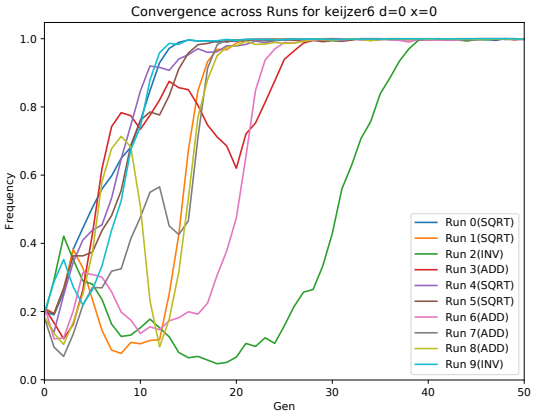

Fig. 18.10: Convergence across many runs for Keijzer6 Root Node. Runs can be driven towards a particular node based on the sample achieved. A handful of functions dominate each position, in this case ADD and SQRT.

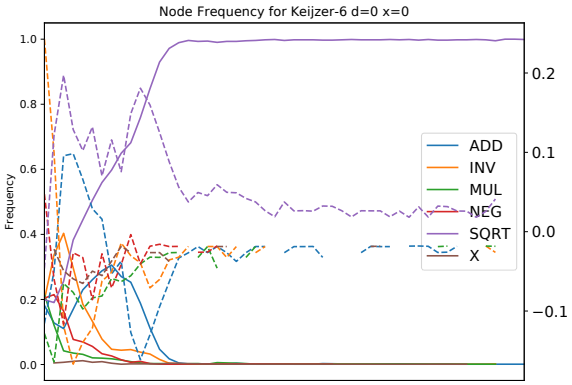

Fig. 18.11: Node Frequency (lines, left scale) and Covariance (dashed lines, right scale) for Keijzer6 Root Node

population size) and exhibit high covariance. Alternatively, we can simply replace nodes at coordinate (d,x) in all trees with the one we want to freeze (which should have the highest covariance with fitness), and then continue the run.

An even more sophisticated alternative is to replace GP entirely, using a sampling algorithm that descends from the root and systematically samples each schema in a 'racing' algorithm, selecting the node in a particular location that exhibits the highest covariance.

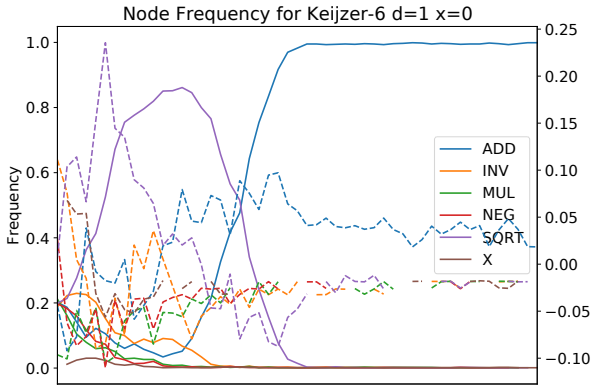

Fig. 18.12: Node Frequency (lines, left scale) and Covariance (dashed lines, right scale) for Keijzer6 d=1 x=0

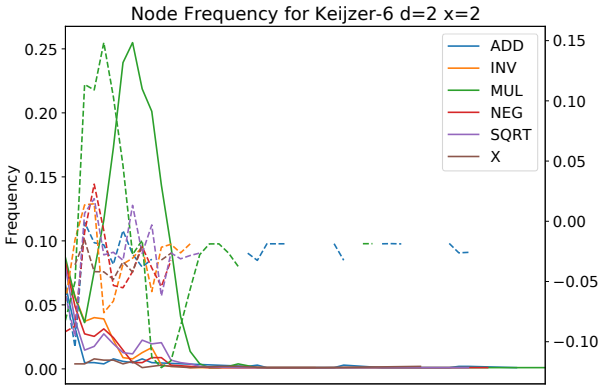

Fig. 18.13: Node Frequency (lines, left scale) and Covariance (dashed lines, right scale) for Keijzer6 d=2 x=2

These proposals require more theoretical and empirical work and we anticipate a need for careful comparison with existing algorithms and across more benchmark problems, all of which is beyond the scope of this chapter but will be subject to a future investigation.

18.6 Related Work

There are two areas of work that are related to our modeling: the development of schema theories for GP, and the empirical examination of the role of crossover. One of the motivating results for this work is the rigorous evaluation of crossover in GP [15], which found crossover was not greatly beneficial to the search. That work was itself based on early controversial and influential work by Luke and Spector [5, 6]. These papers cast doubt on the building block hypothesis and the utility of crossover in reassembling solutions which was - at the time - the predominant search operator used by GP researchers. Other early work came to the conclusion that both mutation and crossover operators contribute to the success of GP runs and that it is beneficial to apply both operators with at least 5% probability, at least in Linear GP [1].

GP Schema theory has a long and varied history, with many different approaches proposed by Rosca [12], O'Reilly [9], and Poli and McPhee [10]; the latter included a multi-paper exposition on schema theory for general crossover, and introduced the grid system used in this paper.

18.7 Conclusion

In this chapter, we have demonstrated how GP can be modeled using a simple single-node schema definition, and that empirical results show convergence throughout the population across benchmarks and experimental runs. In our model, crossover and mutation in GP act simply as "sampling" operators, in agreement with previous empirical experimentation. As the sampling occurs disproportionately towards the bottom of the tree, the driving force in GP is selection, making convergence inevitable. We propose modifying GP based on this model.

This work is incomplete: in particular, we need to examine convergence statistically across every node in a tree's grid; more benchmarks, particularly non-symbolic expression benchmarks, should be examined; and a full covariance-based algorithm should be implemented — either as a modification to GP, or else as a re-imagined sampling algorithm.

Future work includes testing this model in other ways: for example, modifying genetic operators to manipulate the disturbance and construction of simple schemata, and confirming the results agree with prediction.

Acknowledgements WB acknowledges funding from the Koza Endowment provided by MSU.

References

1. Banzhaf, W., Francone, F.D., Nordin, P.: The effect of extensive use of the mutation operator on generalization in genetic programming using sparse data sets. In: H.M. Voigt, W. Ebeling,

I. Rechenberg, H.P. Schwefel (eds.) International Conference on Parallel Problem Solving from Nature (PPSN-96), pp. 300–309. Springer (1996)
2. Blickle, T., Thiele, L.: A mathematical analysis of tournament selection. In: L. Eshelman (ed.) International Conference on Genetic Algorithms (ICGA-95), pp. 9–16. Morgan Kaufmann, San Francisco (1995)
3. Keijzer, M.: Improving symbolic regression with interval arithmetic and linear scaling. In: C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, E. Costa (eds.) European Conference on Genetic Programming, EuroGP 2003, pp. 70–82. Springer, Berlin, Heidelberg (2003)
4. Korns, M.F.: Accuracy in symbolic regression. In: R. Riolo, E. Vladislavleva, J. Moore (eds.) Genetic Programming Theory and Practice IX, pp. 129–151 (2011)
5. Luke, S., Spector, L.: A comparison of crossover and mutation in genetic programming. In: European Conference on Genetic Programming, pp. 240–248. Springer (1997)
6. Luke, S., Spector, L.: A revised comparison of crossover and mutation in genetic programming. In: European Conference on Genetic Programming, pp. 208–213. Springer (1998)
7. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric Semantic Genetic Programming. In: International Conference on Parallel Problem Solving from Nature, pp. 21–31. Springer (2012)
8. O'Reilly, U.M., Oppacher, F.: Using building block functions to investigate a building block hypothesis for genetic programming. Santa Fe Inst., Santa Fe, NM, Working Paper pp. 94–02 (1994)
9. O'Reilly, U.M., Oppacher, F.: The troubling aspects of a building block hypothesis for genetic programming. In: Foundations of Genetic Algorithms (FOGA-95), vol. 3, pp. 73–88. Elsevier (1995)
10. Poli, R., McPhee, N.F.: General schema theory for genetic programming with subtree-swapping crossover: Part i. Evolutionary Computation **11**(1), 53–66 (2003)
11. Price, G.R.: Selection and covariance. Nature **227**, 520–521 (1970)
12. Rosca, J.P., Mallard, D.H.: Rooted-tree schemata in genetic programming. In: K.E. Kinnear, W.B. Langdon, L. Spector, P.J. Angeline, U.M. O'Reilly (eds.) Advances in Genetic Programming, Vol 3, pp. 243–271 (1999)
13. Vladislavleva, E.J., Smits, G.F., Den Hertog, D.: Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. IEEE Transactions on Evolutionary Computation **13**, 333–349 (2008)
14. White, D.R., McDermott, J., Castelli, M., Manzoni, L., Goldman, B.W., Kronberger, G., Jaśkowski, W., O'Reilly, U.M., Luke, S.: Better GP benchmarks: community survey results and proposals. Genetic Programming and Evolvable Machines **14**(1), 3–29 (2013)
15. White, D.R., Poulding, S.: A rigorous evaluation of crossover and mutation in genetic programming. In: European Conference on Genetic Programming, pp. 220–231. Springer (2009)