Advanced Population Diversity Measures in Genetic Programming

Edmund Burke, Steven Gustafson*, Graham Kendall, and Natalio Krasnogor

ASAP Research, School of Computer Science & IT
University of Nottingham, UK
{ekb,smg,gxk,nxk}@cs.nott.ac.uk

Abstract. This paper presents a survey and comparison of significant diversity measures in the genetic programming literature. This study builds on previous work by the authors to gain a deeper understanding of the conditions under which genetic programming evolution is successful. Three benchmark problems (Artificial Ant, Symbolic Regression and Even-5-Parity) are used to illustrate different diversity measures and to analyse their correlation with performance. Results show that measures of population diversity based on edit distances and phenotypic diversity suggest that successful evolution occurs when populations converge to a similar structure but with high fitness diversity.

1 Introduction

Maintaining population diversity in genetic programming is cited as crucial in preventing premature convergence and stagnation in local optima [1][2][3][4][5]. Diversity describes the amount of variety in the population defined by the genetic programming individuals structure or their performance. The number of different fitness values (phenotypes) [6], different structural individuals (genotypes) [7], edit distances between individuals [3] [8], and complex and composite measures [9] [10] [11] are used as measures of diversity.

In this paper, we examine previous uses and meanings of diversity, compare these different measures on three benchmark problems and extend our original study [12] with additional experiments, new analysis and new measures. Population diversity is related to almost every aspect of program evolution and extending the research in [12] will lead to a deeper understanding of evolution in genetic programming. As far as the authors are aware, all the significant diversity measures that occur in the genetic programming literature are reported.

2 Diversity Measures

Measures of diversity attempt to quantify the variety in a population and some methods attempt to control or promote diversity during evolution. The following section surveys both measures that provide a quantification of population

* Corresponding author

J.J. Merelo Guervós et al. (Eds.): PPSN VII, LNCS 2439, pp. 341–350, 2002.

diversity and methods used to actively promote and maintain diversity within genetic programming.

2.1 Population Measures

A common type of diversity measure is that of structural differences between programs. Koza [13] used the term *variety* to indicate the number of different genotypes populations contained. Landgon [7] argues that genotypic diversity is a sufficient upper bound of population diversity as a decrease in unique genotypes must also mean a decrease in unique fitness values.

Keijzer [10] measures program variety as a ratio of the number of unique individuals over population size and subtree variety as the ratio of unique subtrees over total subtrees. Tackett [14] also measures structural diversity using subtress and schemata frequencies. D'haeseleer and Bluming [11] define *behavior* and *frequency* signatures for each individual based on fitness and *gene* frequencies, respectively. The correlation between individuals' respective signatures represents the *phenotypical* and *genotypical* diversity.

When tree representations of genetic programs are considered as graphs, individuals can be compared for isomorphism [5] to obtain a more accurate measure of diversity. Determining graph isomorphism is computationally expensive for an entire population and not straightforward for genetic programs. However, counting the number of nodes, terminals, functions and other properties can be used to determine whether trees are *possible* isomorphs of each other.

McPhee and Hopper [1] investigate diversity at the genetic level by tagging each node created in the initial generation. Root parents, the parents whose tree has a portion of another individual's subtree swapped into it during crossover, are also tracked. McPhee and Hopper found that the number of unique tags dramatically falls after initial generations and, by tracking the root parents, after an average of 16 generations, all further individuals have the same common root ancestor.

Phenotypic measures compare the number of unique fitness values in a population. When the genetic programming search is compared to traversing a fitness landscape, this measure provides an intuitive way to think of how much the population covers that landscape. Other measures could be created by using fitness values of a population, as done by Rosca [5] with entropy and free energy. Entropy here represents the amount of disorder of the population, where an increase in entropy represents an increase in diversity.

2.2 Promoting Diversity

Several measures and methods have been used to promote diversity by measuring the difference between individuals. These methods typically use a non-standard selection, mating, or replacement strategy to bolster diversity. Common methods are neighborhoods, islands, niches, and crowding and sharing from genetic algorithms.

Eschelman and Schaffer [15] use Hamming distances to select individuals for recombination and replacement to improve over hill-climbing-type selection strategies for genetic algorithms. Ryan's [2] "Pygmie" algorithm builds two lists based on fitness and length to facilitate selection for reproduction. The algorithm maintains more diversity, prevents premature convergence and uses simple measures to promote diversity. De Jong et al [8] use multiobjective optimisation to promote diversity and concentrate on non-dominated individuals according to a 3-tuple of <*fitness, size, diversity*>. Diversity is the average square distance to other members of the population, using a specialised measure of edit distance between nodes. This multiobjective method promotes smaller and more diverse trees.

McKay [4] applies the traditional fitness sharing concept from Deb and Goldberg [16] to test its feasibility in genetic programming. Diversity is the number of fitness cases found, and the sharing concept assigns a fitness based on an individual's performance divided by the number of other individuals with the same performance. McKay also studies negative correlation and a *root quartic negative correlation* in [9] to preserve diversity. Ekárt and Németh [3] apply fitness sharing with a novel tree distance definition and suggest that it may be an efficient measure of structural diversity. Bersano-Begey [17] track how many individuals solve which fitness cases and a pressure is added to individuals to promote the discovery of different or less popular solutions.

3 Experiment Design

Our initial study of population diversity measures [12] highlighted that phenotypic measures appeared to better correlate with better fitness. Runs which had better fitness in the last generation also tended to have higher phenotypic diversity measures. This appears to go against conventional wisdom in genetic programming which says that runs must converge to an "exploitation" phase where diversity is lost to focus on better individuals. However, it does agree with the intuitive idea that proper evolution *needs* diversity to be effective.

In this study we extend our original analysis [12] with new experiments and new measures of population diversity which we have adapted from diversity promoting methods. In analysing results, we measure the Spearman correlation [18] between diversity and fitness and examine standard deviations, minimum and maximum values and the diversity of all populations in every run and the best fitness of those populations.

Three common problems are used with common parameter values from previous studies. For all problems, a population size of 500 individuals, a maximum depth of 10 for each individual, a maximum depth of 4 for the tree generation half-n-half algorithm, standard tree crossover and internal node selection probability of 0.9 for crossover is used. Additionally, each run consists of 51 generations, or until the ideal fitness is found.

The Artificial Ant, Symbolic Regression and Even-5-Parity problems are used. All three problems are typical to genetic programming and can be found in

many studies, including [13]. The artificial ant problem attempts to find the best strategy for picking up pellets along a trail in a grid. The fitness for this problem is measured as the number of pellets missed. The regression problem attempts to fit a curve for the function $x^4 + x^3 + x^2 + x$. Fitness here is determined by summing the squared difference for each point along the objective function and the function produced by the individual. The even-5-parity problem takes an input of a random string of 0's and 1's and outputs whether there are an even number of 1's. The even-5-parity fitness is the number of wrong guesses for the 2^5 combinations of 5-bit length strings. All problems have an ideal fitness of low values (0=best fitness).

To produce a variety of run performances, where we consider the best fitness in the last generation, we designed three different experiments, carried out 50 times, for each problem. The first experiment, *random*, performs 50 independent runs. The experiment *stepped-recombination* does 50 runs with the same random number seed, where each run uses an increasing probability for reproduction and decreasing probability for crossover. Initially, probability for crossover is 1.0, and this is decreased by 0.02 each time (skipping value 0.98) to allow for exactly 50 runs and ending with reproduction probability of 1.0 and crossover probability 0.0. The last experiment, *stepped-tournament*, is similar but we begin with a tournament size of 1 and increment this by 1 for each run, until we reach a tournament size of 50. In the *random* and *stepped-tournament* experiments, crossover probability is set to 1.0 and the tournament size in *random* and *stepped-recombination* is 7. The *Evolutionary Computation in Java* (ECJ), version 7.0, [19] is used, where each problem is available in the distribution.

The following measures of diversity were introduced previously and are briefly described as they are collected for each generation in every run. **Genotype and phenotype** diversity count the number of unique trees for the genotype measure [7] and the number of unique fitness values in a population represents the phenotype measure [6]. The **entropy** measure is calculated for the population as in [5], where "p_k is the proportion of the population P occupied by population partition k", $-\sum_k p_k \cdot \log p_k$. A partition is assumed to be each possible different fitness value, but could be defined to include a subset of values. **Pseudo-isomorphs** are found by defining a 3-tuple of <terminals,nonterminals,depth> for each individual and the number of unique 3-tuples in each population is the measure. Two identical 3-tuples represent trees which could be isomorphic. **Edit distance 1 and 2** is the edit distance between individuals used by de Jong et al [8] (referred to as "ed 1" in the graphs) and an adapted version of Ekárt and Németh [3] ("ed 2"). Every individual in the population is measured against the best fit individual. This measure is then divided by the population size. The first measure (ed 1) is a standard edit distance measure where two trees are overlapped at the root node. Two different nodes, when overlapping, score a distance of 1 and equal nodes get 0. The edit distance is then the sum of all different nodes and normalised by dividing it by the size of the smaller tree. The second measure (ed 2) is slightly adapted back to its original formulation in [20] where the difference

between any two nodes is 1. The difference between two trees is then (defined in [3]):

$$dist(T_1, T_2) = \begin{cases} d(p,q) & \text{if neither } T_1 \text{ nor } T_2 \text{ have any children} \\ d(p,q) + \frac{1}{2} * \sum_{l=1}^{m} dist(s_l, t_l) & \text{otherwise} \end{cases}$$

Where T_1, T_2 are trees with roots p, q and possible children (m total) subtrees s, t. Two trees are brought to the same tree structure by adding "null" nodes to each tree. Note that the differences near the root have more weight, a possibly convenient description for genetic programming as it has been noted that programs converge quickly to a fixed root portion [1].

The Spearman correlation coefficient is computed as [18] $1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N^3 - N}$. Where N is the number of items (50 runs), and d_i is the distance between each run's rank of performance and rank of diversity in the last generation. A value of -1.0 represents negative correlation, 0.0 is no correlation and 1.0 is positive correlation. For our measures, if we see ideal low fitness values, which will be ranked in ascending order (1=best,. . .,50=worst) and high diversity, ranked where (1=lowest diversity and 50=highest diversity), then the correlation coefficient should be strongly negative. Alternatively, a positive correlation indicates that either bad fitness accompanies high diversity or good fitness accompanies low diversity.

4 Results

Graphs of the 50 runs for all three experiments and all three problems were examined, along with minimum, maximum and standard deviations of best fitness and population diversity measures. Also, the Spearman correlation coefficient was calculated, correlating the diversity measures with best fitness across each set of 50 runs. This study involved 450 runs of 51 generations each, adding to a previous study [12] of the same size with different random seeds and different measures of diversity. While all three problems showed the same general trends, we focus on the artificial ant and even-5-parity.

Figure 1 shows for the artificial ant problem and *random* experiment, that diversity measures and fitness varied widely. The most dramatic activity occurs early with runs being similar until around generation 10, where they become quite varied. However, from Table 1 we can see several interesting phenomenon. First, by noting the genotype measure and best fitness standard deviations for the artificial ant experiment, we see little variance of best fitness (11.2,15.4,15.9) but large variance of genotype diversity (18.3,120.1,44.2). Also note that the genotype diversity for the random experiment in artificial ant and even-5-parity have very high minimum and maximum values, where the other measures minimum and maximum does not differ across experiments. This information leads us to believe that the genotype diversity measure does not suggest a strong correlation with varying run performance. Note how the other measures have consistent variation, as does fitness.

Using the Spearman correlation coefficient we investigated whether runs that produced good fitness had low/high diversity, where ties in ranks were solved by

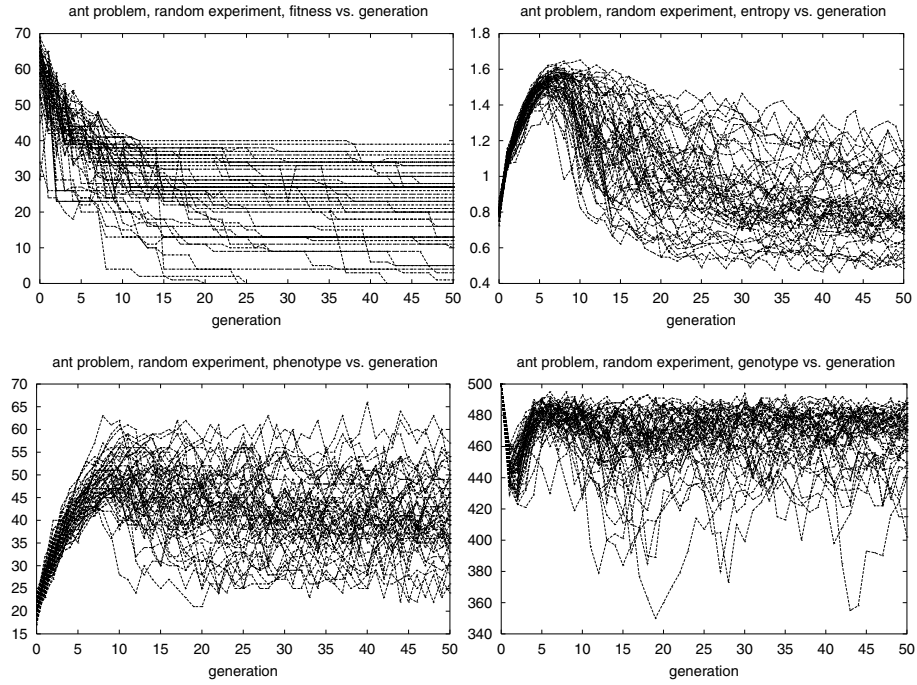

Fig. 1. 50 runs of best fitness per generation for the artificial ant random experiment and a graph for each of the diversity measures of entropy, phenotype and genotype diversity.

splitting the rank among the tying items (add possible ranks and average). Remembering that negative correlation (values close to -1.0) suggest high diversity is correlated with good performance (as we want to minimize fitness). Table 1 shows that high negative correlation is seen most consistently with entropy and phenotype diversity. In fact, only these two measure always produce negative correlation, indicating that a high phenotype variance and entropy values accompany the best fit runs.

Figure 2 shows graphs for the same problem where every populations' best fitness and edit distance diversity measure are plotted. The artificial ant and even-5-parity graphs shown here demonstrate a very interesting phenomenon. Notice that best fitness values (close to 0) also consistently have low edit distance diversity, meaning that for populations containing the best fit individuals, those populations are similar to the best fit individual. The even-5-parity problem indicates that best fitness only occurs in populations that have low edit distance diversity. The artificial ant problem shows that poor fitness tends to occur in populations with higher edit distance diversity and also better populations have low edit distance diversity.

While the phenotypic measures seem to indicate that better performance is accompanied with higher diversity, the edit distance diversity results appear to

Table 1. Problems artificial ant and even-5-parity with experiments *random* (rand), *stepped-tournament* (step-t) and *stepped-recombination* (step-r). Values are from the *final population*. Best fitness ("b.fit") is the best fitness in the final generation. The Spearman coefficient shows perfect correlation with 1.0, negative correlation with -1.0 and no correlation with 0.0. Bold numbers are mentioned in the text and negative correlation indicates that best fitness is correlated with high diversity measure values.

artificial ant problem												
	spearman			min max						standard dev		
	random	step-t	step-r	random		step-t		step-r		random	step-t	step-r
b.fit	-	-	-	0.0	39.0	0.0	50.0	0.0	73.0	**11.221**	**15.378**	**15.944**
gene	0.2673	-0.0533	0.5110	**402.0**	**491.0**	1.0	476.0	271.0	487.0	**18.339**	**120.109**	**44.238**
isom	0.4135	0.0874	0.5816	75.0	339.0	1.0	291.0	67.0	354.0	66.0767	74.3093	63.1815
phene	**-0.2214**	**-0.2029**	**-0.0079**	24.0	57.0	1.0	54.0	13.0	62.0	8.2239	9.4150	7.3103
entro	**-0.358**	**-0.597**	**-0.4506**	0.4829	1.3339	0.0	1.2927	0.6010	1.3498	0.1939	0.2584	0.1958
ed1	-0.0128	-0.4799	-0.1646	0.0876	0.5082	0.0	0.3746	0.0558	0.8245	0.0890	0.0824	0.1110
ed2	0.2874	-0.4196	-0.0606	0.4864	7.0751	0.0	3.5343	0.5201	6.0184	1.3466	0.7675	1.0564
even-5-parity problem												
	spearman			min max						standard dev		
	random	step-t	step-r	random		step-t		step-r		random	step-t	step-r
b.fit	-	-	-	3.0	12.0	4.0	15.0	3.0	15.0	1.8762	2.2670	2.7734
gene	0.1788	-0.3165	0.3295	**412.0**	**482.0**	9.0	470.0	269.0	484.0	**14.0285**	**103.544**	**38.224**
isom	0.2388	0.2221	0.3500	45.0	89.0	1.0	119.0	23.0	123.0	10.0312	19.2771	20.2564
phene	**-0.7326**	**-0.7796**	**-0.8494**	6.0	16.0	1.0	15.0	3.0	15.0	1.8999	2.3756	2.3427
entro	**-0.6978**	**-0.7317**	**-0.763**	0.5431	0.9444	0.0	0.8996	0.0176	0.8829	0.08168	0.1840	0.1439
ed1	0.5628	0.4044	0.5853	0.0737	0.3664	0.0426	0.7840	0.0520	0.8484	0.0644	0.1296	0.1286
ed2	0.3806	0.3344	0.4738	0.3917	2.5040	0.1786	5.277	0.2846	3.4607	0.4327	0.8776	0.7364

contradict that by suggesting that better performance is in populations with low edit distance diversity. In fact, these results indicate something quite interesting, that genetic programming is most successful when populations converge to a similar structure but in a manner which preserves diversity.

Figure 3 demonstrates that when the Spearman correlation is calculated for every population during evolution (150 runs total for each problem) how the different diversity measures correlate with performance *during* evolution. For the different problems some diversity measures correlate better at different times during evolution. Notice the early random behaviour around generations 5-10, the same time of divergence in the graphs in Figure 1 and also the general point of convergence of root ancestors, described in [12] [1].

5 Conclusions

The measures of diversity surveyed and studied here indicate that genotype diversity may not be useful for capturing the dynamics of a population, because of the low correlation. This is also suggested in [2][10]. The fitness based measures of phenotypes and entropy appear to correlate better with run performance. The measures of edit distance diversity, one being a traditional edit distance and the other giving more weight to differences near the root, seem to provide useful information about populations with good/poor performance. Better fit individuals come from populations with low edit distance diversity, meaning that the population is similar to the best fit individual. This information accompanied

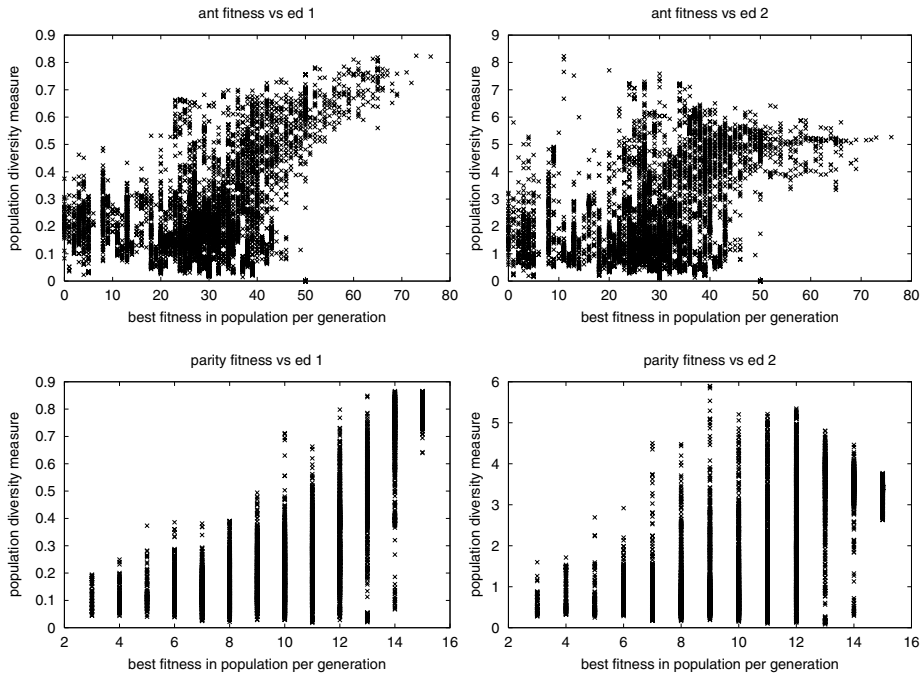

Fig. 2. The best fitness per population is plotted (x axis) against that population's edit distance diversity (ed1, ed2). Here, low fitness is better for both the artificial ant and even-5-parity problems.

with our previous results seem to suggest that populations converge to a similar structure but keep high diversity.

Moreover, if we consider the curves for edit distance diversity (ed1, ed2) *together* with those for phenotype diversity (phenes, entropy) in Figure 3, during most of evolution edit distance diversity correlates positively and phenotype diversity negatively with high fitness. These results taken *together* show that the fitness landscapes defined by the genetic operators chosen and the fitness function used for the problems studied are uncorrelated, i.e., individuals with low edit distance have very different phenotype characteristics (e.g. fitness). This in turn suggests that the search capabilities of the algorithms studied in this paper might be impaired.

While the edit distance measures are expensive, if they prove useful in predicting successful runs we could attempt to find accurate approximations or limit their use to defined generational intervals. Finally, results showed that evolving populations have diversity values which fluctuate between positive and negative correlation with best fitness and this behaviour varied among the studied problems. This paper also indicates the need to carefully define diversity measures and the goal of those measures (high or low values) when using diversity to assess or alter genetic programming evolution.

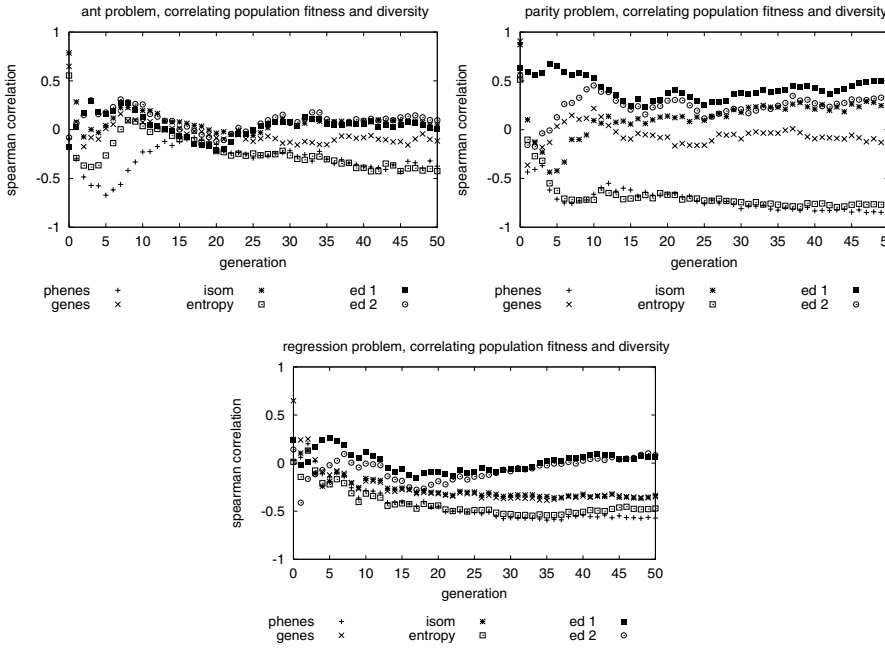

Fig. 3. For each problem, the Spearman correlation between a populations diversity and best fitness is calculated across all runs. Note the fluctuations between negative, no and positive correlation as the populations change during evolution.

6 Future Work

Current research includes studying new problems, tracking root ancestors and other measures during evolution, and applying methods to promote diversity while using different measures to determine their effects. Fitness landscape distance correlation is also being investigated.

References

1. N.F. McPhee and N.J. Hopper. Analysis of genetic diversity through population history. In W. Banzhaf et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, Florida, USA, 1999. Morgan Kaufmann.
2. C. Ryan. Pygmies and civil servants. In K.E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 11, pages 243–263. MIT Press, 1994.
3. A. Ekárt and S. Z. Németh. A metric for genetic programs and fitness sharing. In R. Poli et al., editors, *Proceedings of the European Conference on Genetic Programming*, volume 1802 of *LNCS*, pages 259–270, Edinburgh, 15-16 April 2000. Springer-Verlag.
4. R.I. McKay. Fitness sharing in genetic programming. In D. Whitley et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 435–442, Las Vegas, Nevada, USA, 10-12 July 2000. Morgan Kaufmann.

5. J.P. Rosca. Entropy-driven adaptive representation. In J.P. Rosca, editor, *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, pages 23–32, Tahoe City, California, USA, 9 July 1995.
6. J.P. Rosca. Genetic programming exploratory power and the discovery of functions. In J.R. McDonnell et al., editors, *Proceedings of the Fourth Conference on Evolutionary Programming*, pages 719–736, San Diego, CA, 1995. MIT Press.
7. W.B. Langdon. Evolution of genetic programming populations. Research Note RN/96/125, University College London, Gower Street, London WC1E 6BT, UK, 1996.
8. E.D. de Jong, R.A. Watson, and J.B. Pollack. Reducing bloat and promoting diversity using multi-objective methods. In L. Spector et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, CA, 7-11 July 2001. Morgan Kaufmann.
9. R.I. McKay and H.A. Abbass. Anticorrelation measures in genetic programming. In *Australasia-Japan Workshop on Intelligent and Evolutionary Systems*, 2001.
10. M. Keijzer. Efficiently representing populations in genetic programming. In P.J. Angeline and K.E. Kinnear, Jr., editors, *Advances in Genetic Programming 2*, chapter 13, pages 259–278. MIT Press, Cambridge, MA, USA, 1996.
11. P. D'haeseleer. Context preserving crossover in genetic programming. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, volume 1, pages 256–261, Orlando, FL, USA, June 1994. IEEE Press.
12. E. Burke, S. Gustafson, and G. Kendall. Survey and analysis of diversity measures in genetic programming. In *(Accepted as a full paper) Proceedings of the Genetic and Evolutionary Computation Conference*, 2002.
13. J.R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
14. W.A. Tackett. *Recombination, Selection, and the Genetic Construction of Computer Programs*. PhD thesis, University of Southern California, Department of Electrical Engineering Systems, USA, 1994.
15. L.J. Eshelman and J.D. Schaffer. Crossover's niche. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 9–14, San Mateo, CA, 1993. Morgan Kaufman.
16. K. Deb and D.E. Goldberg. An investigation of niche and species formation in genetic function optimization. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, Washington DC, 1989.
17. T.F. Bersano-Begey. Controlling exploration, diversity and escaping local optima in GP. In J.R. Koza, editor, *Late Breaking Papers at the Genetic Programming Conference*, Stanford University, CA, July 1997.
18. S. Siegel. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill Book Company, Inc., 1956.
19. S. Luke. ECJ: A java-based evolutionary comp-utation and genetic programming system, 2002. http://www.cs.umd.edu/projects/plus/ecj/.
20. S.-H. Nienhuys-Cheng. Distance between Herbrand interpretations: a measure for approximations to a target concept. In N. Lavraĉ and S. Dẑeroski, editors, *Proceedings of the 7th Internations Workshop on Inductive Logic Programming*. Springer-Verlag, 1997.