Diversity in Genetic Programming: An Analysis of Measures and Correlation With Fitness

Edmund K. Burke, Steven Gustafson, and Graham Kendall

***Abstract*—This paper examines measures of diversity in genetic programming. The goal is to understand the importance of such measures and their relationship with fitness. Diversity methods and measures from the literature are surveyed and a selected set of measures are applied to common standard problem instances in an experimental study. Results show the varying definitions and behaviors of diversity and the varying correlation between diversity and fitness during different stages of the evolutionary process. Populations in the genetic programming algorithm are shown to become structurally similar while maintaining a high amount of behavioral differences. Conclusions describe what measures are likely to be important for understanding and improving the search process and why diversity might have different meaning for different problem domains.**

***Index Terms*—Diversity, genetic programming, population dynamics.**

I. Introduction

THE AIM OF this paper is to develop a general understanding of diversity in genetic programming and to gain additional insight into the algorithm's search behavior. More specifically, we would like to understand how one could improve fitness by controlling diversity. Toward this goal, we survey previous measures and methods of diversity and apply them in an experimental study. The genetic programming algorithm can be difficult to reason about theoretically, as has been shown by numerous past attempts (see [30] for a review). Many experimental studies have been insightful in uncovering and addressing various aspects of the algorithm's properties; Daida *et al.*'s examination of problem difficulty and single node building block analysis [7], [9] is a good example. Additionally, previous investigations into measures of diversity have given the community a clearer view of populations and the evolutionary process of genetic programming [11], [19], [26], [30], [40], [44]. To assess how informative different types of diversity measures are, we address the relationship between population diversity and fitness.

This program of research was also motivated by the low level of research activity into identifying diversity measures which correlate with fitness. Conventional wisdom suggests that increasing diversity should be generally beneficial. However, there are many possible definitions of diversity in a representation like genetic programming. Identifying such measures could allow more prediction of run performance, improved understanding of populations and could enable the design of more efficient operators and genetic programming algorithms.

Three main questions are raised and addressed in this paper.

1) How informative are various measures of diversity (structural and behavioral measures)?
2) Does there exist a correlation between the best fitness and diversity of populations?
3) Does diversity play a more significant role at different stages of the evolutionary process?

As genetic programming is highly stochastic, we do not expect to obtain clear (and always applicable) rules about exact levels of diversity. We aim to draw general conclusions and "rules of thumb" from the investigation of evolving populations with different measures of diversity.

The genetic programming literature consistently cites the importance of maintaining diversity as being crucial in avoiding premature convergence toward local optima [18], [38], [40], [44], [46]. Diversity is a key element of the biological theory of natural selection and is used in genetic programming to describe structural or behavioral variety in the population. The term *diversity* is often used without definition and the implicit assumption is the diversity of genotypes, or structural diversity, as this is the common use in the genetic algorithm literature. Measures of diversity have, however, been defined as the number of different behaviors (fitness values or phenotypes) [45], the number of different structures (individuals, programs, or genotypes) [29], the edit distance between structures in the population [12], [18], and other complex or composite measures [15], [26], [39].

The previous uses and meanings of diversity from the literature are examined and applied to four standard problem instances (two with continuous fitness spaces and two with discrete spaces) to develop a fuller picture of diversity in genetic programming. This paper significantly extends our initial studies [4], [5] with additional experiments and new and more complete analysis. Genetic programming evolves solutions by means of a population. Thus, population diversity is related to nearly every aspect of the evolutionary process. Extending this line of research will lead to a deeper understanding of the algorithm.

II. Diversity Measures

Measures of diversity are concerned with the levels and types of variety in populations. Such measures can be defined over general features, including fitness values, structures, or a combination of the two. Diversity measures can also be defined with specific problem domains in mind, such as composite measures of behavioral types which the fitness function does not express

Manuscript received February 3, 2003; revised July 2, 2003.

The authors are with the School of Computer Science and Information Technology, University of Nottingham, Nottinghan NG8 1BB, U.K. (e-mail: smg@cs.nott.ac.uk).

Digital Object Identifier 10.1109/TEVC.2003.819263

1089-778X/04$20.00 © 2004 IEEE

[15]. Additionally, there are methods that attempt to control or promote diversity during evolution. Depending on the specific problem or representation being used, infinitely many diversity measures and methods could exist. In this paper, we focus on measures developed for general problems and measures that are commonly used, especially in the genetic programming literature with the tree representation. However, we also report and analyze significant measures from across the genetic programming field and other relevant genetic algorithm measures. This section provides a survey of the significant measures and the methods used to control diversity levels within genetic programming populations.

A. Population Measures

Biological diversity refers to the differences between individuals in a population, which, by the nature of biology, implies a structural and behavioral difference. In genetic programming, the standard use of diversity refers to structural differences only. This does not guarantee behavioral difference and usually only implies that two structures are not identical. Koza [28] used the term *variety* to indicate the number of different genotypes that were contained in a population. In a standard genetic programming population, this would be the number of structurally unique individuals, trees, or programs. While this measure is probably the least informative it is the most common due to its ease of use and understanding. Langdon [29] argued that genotypic diversity is a sufficient upper bound of population diversity. Due to the nature of most genetic programming systems and problems, two identical structures will produce the same behavior (fitness). Thus, a decrease in genotypic diversity (unique structures) will necessarily cause a decrease in unique behaviors. In his treatment of the *stack* problem [29], Langdon investigated the effects of the crossover operator on variety. The author noted that genetic programming loses some ability to improve fitness after 20–30 generations and it is most probably due to crossover causing a loss of variety. Langdon also noted that in the *stack* problem, runs with better fitness appeared to allow crossover to produce a larger number of fitter (and nonduplicate) children than their parents.

The standard program representation (tree structures) in genetic programming lends itself to more fine grain structural measures that consider nodes, subtrees, and other graph theoretic properties (rather than just entire trees). Keijzer [26] measured subtree variety as the ratio of unique subtrees over total subtrees and program variety as a ratio of the number of unique individuals over the size of the population. Keijzer also used a distance measure between two individuals as the number of distinct subtrees the individuals share. Tackett [51] also measured structural diversity using subtrees and schemata frequencies.

Problem specific measures can allow additional insight into population diversity, especially on novel and nontraditional problems. D'Haeseleer and Bluming [15] defined *behavior* and *frequency* signatures for each individual based on fitness and *gene* frequencies, respectively. The average correlation between every two individuals' respective signatures represents the *phenotypical* and *genotypical* diversity of the population. In addition, D'haeseleer and Bluming tag genetic code and evaluate the behavior of individuals with a "stimulus-response map" to gain further knowledge into the structure and behavior of populations in their robot tank problem. Using these measures, the authors witnessed emerging demes with neighborhood selection and mating.

Graph isomorphism could be applied to genetic programming tree structures as a measure of diversity. However, due to the nature of nodes used in genetic programming, the properties (associativity, commutativity, etc.) would require special, and possibly complex, implementations of isomorphism [44]. Also, determining graph isomorphism would be computationally expensive for an entire population. However, a measure of possible isomorphic trees could be found by noting simple properties (terminal, functions, depth, etc.) to determine the individuals which could be isomorphic without actually computing isomorphism.

McPhee and Hopper [40] investigated diversity at the genetic level by assigning numerical tags to each node in the population. The tags track the survival of nodes from the initial population and the change of context for nodes during recombination. The authors also tracked the genetic lineages from the initial population by noting the individuals selected for recombination, which child they produced and which of the parents provided the root portion of the child's tree. They found that populations in the final generation often descended from one single initial individual, and genetic lineages were effectively reduced to one surviving lineage early on in the evolutionary process.

Measuring the difference between two individuals based on string edit distances has been used several times in genetic programming. O'Reilly [42] used an edit distance based on string matching, which uses single node insertions, deletions and substitutions to transform two trees to be equal in structure and content. De Jong *et al.* [12] also used a similar edit distance in a multiobjective method. Ekárt and Németh [18] defined an edit distance specific to genetic programming parse trees, adapted from [41], which considered the cost of substituting between different node types (functions versus terminals and within these classes).

The diversity measures discussed above are based on structural differences (except [15]). The measure of success in evolutionary algorithms is typically the fitness of a solution or behavior in the problem's environment. Measures based on behavior compare differences among the populations' fitness values at a given time. Rosca [44] used the fitness values in a population to define an entropy and free energy measure. Entropy represents the amount of disorder of the population, where an increase in entropy represents an increase in diversity. Rosca found that populations appeared to be stuck in local optima when entropy did not change or decreased monotonically in successive generations.

B. Promoting Diversity

The canonical view of evolution and diversity is that more diversity will provide more opportunities for evolution. However, as noted in several diversity studies (see [19]), typical evolutionary algorithms contain a phase of exploration followed by exploitation. Promoting all kinds of diversity during the entire evolutionary process could be counterproductive to the exploitation phase. The type and amount of diversity required at

different evolutionary times remains rather unclear. However, several measures and methods have been used to promote diversity. These methods typically use a nonstandard selection, mating, or replacement strategy to increase or control diversity. Common methods are geographical distributions of individuals that define their interactions (neighborhoods [6] and islands [35]) and methods which consider the behavior similarities (sharing [24]) or structural similarities (crowding [14], or genotype sharing [13]) to define individual interactions. These common techniques were initially applied in genetic algorithms.

Eshelman and Schaffer [20] investigated the advantage of pair-wise mating in genetic algorithms. The authors used Hamming distances to select individuals for recombination and replacement to improve over hill-climbing-type selection strategies for genetic algorithms. Ryan's "Pygmie" algorithm [46] addressed premature convergence and elitism in small populations for evolving minimal sorting networks. The algorithm builds two lists based on fitness and length to facilitate selection for reproduction. Ryan's algorithm maintained more diversity, prevented premature convergence, and used simple measures to promote diversity. De Jong *et al.* [12] used multiobjective optimization for the n-parity problem to promote diversity and concentrate on nondominated individuals according to a three-tuple of $\langle fitness, size, diversity \rangle$. 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.

Keller and Banzhaf [27] described a structural difference measure based on the edit distances between two genotypes. The measure is more complicated than standard edit distance but is intended for explicitly controlling the diversity of populations. Brameier and Banzhaf [3] used a string edit distance on the effective portions of their *linear genetic programming* individuals, measuring the distance between the program code which contributes to fitness. They used their measure in a two-level tournament, selecting for fitness and then for diversity.

McKay [38] applied the traditional fitness sharing concept from the work of Deb and Goldberg [13] to test its feasibility in genetic programming. The fitness sharing technique is credited with maintaining population diversity that allowed performance improvements and population size reductions for the multiplexer and recursive list membership problems. 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. Also, McKay studied negative correlation [31] and a *root quartic negative correlation* [37], [39] to preserve diversity on the multiplexer problem with mixed results. Ekárt and Németh [18] apply fitness sharing with a novel tree distance definition to a symbolic regression instance and suggest that it may be an efficient measure of structural diversity. Their results showed promise for controlling the size of programs without initially improving performance. The authors then apply their measure between every pair of individuals in a weighted arithmetic mean to develop a population diversity measure [19]. This measure is used to adaptively control diversity for broad and more focused search phases as it was noted that a conflict between fitness improvement and high diversity was observed in their previous work. The authors find that on their symbolic regression instances, fitness sharing is able to improve accuracy and maintain population diversity.

Bersano–Begey [1] tracked how many individuals solve specific fitness cases, where a pressure was added to promote diversity and the discovery of different or less popular solutions. This is similar to the *Stepwise Adaptation of Weights* [17] technique for constraint satisfaction and symbolic regression instances [16]. Smith *et al.* [48] investigated diversity within their immune system algorithm for classifier systems, based on a standard genetic algorithm. Their task is not concerned with *traditional* optimization and requires diverse populations to be successful. A speciation tree using Euclidean distance is applied by Bessaou *et al.* [2] in their study on multimodal optimization with island models. Their algorithm splits individuals into species, evolves them with a genetic algorithm, and then redistributes them into new species. Geard and Wiles [23] counted unique genotypes while studying recombination and diversity for a genetic algorithm solving their "royal staircase" problem.

Fernandes and Rosa [21] looked at varying population sizes and nonrandom mating to maintain diversity for the Royal Road problem. Their negative assortative mating looks for genotypes with maximal Hamming distances. Darwen and Yao [11] studied cooperation in the iterated prisoner's dilemma problem and found that increasing behavioral diversity, not genetic diversity, can improve cooperation and performance. The authors also comment on the dogma surrounding diversity and some previous methods to maintain diversity [10]. Ursem [52] cited the importance of high and low diversity phases in an evolutionary strategy framework. The author used a "distance-to-average-point" diversity measure for his real-value encoded individuals. Depending on whether the diversity is in a predefined high or low phase, different recombination operators are used which allow diversity to fall or which promote more diversity, respectively.

C. Studying Diversity

Low diversity is often mentioned as the reason for poor performance in evolutionary algorithms. Some methods mentioned above have attempted to improve, control or maintain diversity to improve their algorithms, while others have noticed unusual behavior while studying diversity issues in their research.

O'Reilly [42] noted the importance of using structural distance measures on genetic programming populations to understand the underlying dynamics. An edit distance measure is used here to study the effects of crossover and the differences between individuals and better individuals. While no clear results are found, the ability to understand genetic programming populations with edit distance measures is suggested. Keijzer [26] noted that his distance measure of distinct subtrees between two individuals could be used to predict when subtree crossover will fail to provide improvements due to loss of diversity. Langdon [29] found that the loss of diversity caused a decrease of unique terminals which, due to subtree crossover, led to further diversity loss. Langdon and Poli [30] later noted that measuring variety with only unique genotypes fails to consider the ancestral history of individuals, the degree of difference between nonunique individuals and their behavioral similarities. Our initial research examined common measures of diversity [4] and

measures based on edit distance [5]. These studies also briefly measured the correlation between diversity and fitness, noting that traditional measures based on unique genotypes had very low correlation with fitness.

In conclusion, measures of diversity, and studies using those measures, can provide different levels of knowledge about the evolving populations. The more detailed a measure is, the more computation, implementation, and analysis expense there is likely to be. Therefore, there is a need to find informative and inexpensive measures which can capture detailed information about populations (such as the ability to improve or get out of local optima). The focus of genetic programming is usually driven by a performance goal (i.e., fitness improvement or generality of populations) and not by the level of diversity. The level of diversity is not seen in itself as a goal. Thus, identifying the measures of diversity that are correlated with fitness is crucial.

The focus of this paper is to more thoroughly investigate different measures of population diversity, especially with respect to edit distance measure and the correlation of different measures *during* evolution. This paper builds significantly on [4] and [5]. Two problems with discrete fitness spaces and two instances of the regression problem with continuous fitness spaces are considered in a quantitative study. We hope to better understand how measures of diversity perform in these different fitness spaces. It should be mentioned that diversity is studied in other areas of evolutionary algorithms (neural network ensembles [31] for example) but is out of the scope of this paper.

D. Correlation Measures

An objective of this paper is to quantify the importance and levels of diversity, recorded by different measures, on typical problems. In this paper, we collected 1000 independent runs for each problem. As correlation measures, especially the nonparametric one used here, are not particularly appropriate for extremely large samples, we generally use sample sizes of 100. Larger and smaller samples were tried with no useful benefits seen.

Our primary test of the relationship between diversity and fitness is the Spearman correlation measure [47]. The Spearman measure ranks two sets of variables and tests for a linear relationship between the variables' ranks. Initially, we are interested in whether two runs can be distinguished by their diversity in terms of which run is better. Interesting relationships could easily exist but not necessarily be linear. We also evaluate a range of scatter plots which can show linear relationships in addition to others, as will be seen with edit distance measures later.

The Spearman correlation coefficient is computed (from [47]) as follows:

$$1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N^3 - N}$$

where N is the number of items, and d_i is the distance between each population's rank of fitness and rank of diversity. A value of -1.0 represents negative correlation, 0.0 denotes no correlation, and 1.0 demonstrates positive correlation. For our measures, if we see ideal low fitness values, which will be ranked in ascending order ($1 = \text{best}, \ldots, 50 = \text{worst}$) and high diversity, ranked in ascending order ($1 = \text{lowest diversity}$ and $50 = \text{highest diversity}$), 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.

III. Experiments

Four common problem instances and parameter values are used (see [8], [32], [34], [40], [50]). As previous studies into the dynamics, code growth, recombination, and theoretical foundations in genetic programming use similar problems and parameter settings, we felt it appropriate to use them here as well. 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 ramped half-n-half algorithm, standard subtree crossover, and internal node selection probability of 0.9 for crossover is used. Each run consists of 51 generations.

The crossover probability is set to 1.0 (no mutation is used), the tournament size is 4, and the Mersenne Twister random number generator [36] is seeded with the current time in milliseconds for each run. *Evolutionary Computation in Java*, version 7.0, [33] is used, where each problem (except Rastrigin, which was modified from the regression problem) is available in the distribution. The setup of all experiments are summarized in a parameter file which allows the exact same run to be rerun for verification. Note that the measures of diversity (and necessary modification to accommodate those measures) are not available in the *Evolutionary Computation in Java* framework, but implementation detail can be acquired from the authors.

A. Problems

The artificial ant, even-5-parity, and symbolic regression problems (with the quartic polynomial and Rastrigin function) are used. All four problem instances are common in the genetic programming literature and can be found in many studies, including [28], [30], [40], and [50]. The functions and terminals of each problem are summarized in Table I along with other experiment parameters. Each of the problem instances can be summarized as follows.

1) Artificial Ant: The artificial ant problem (with the Sante Fe trail) consists of finding the best strategy for picking up pellets along a trail in a grid. The Sante Fe trail contains 89 food elements on a two-dimensional surface. The ant problem uses the *if_food_ahead, progn2*, and *progn3* functions and *left, right*, and *move* terminals. The function *if_food_ahead* tests for a food pellet and executes one of its two arguments. The other two functions (*progn2, progn3*) execute their arguments in succession. The terminals *left* and *right* turn the ant, and the *move* terminal moves the ant forward. The fitness for this problem is measured as the number of pellets missed. The artificial ant problem is investigated in several studies. Recently Langdon and Poli [30] report an in-depth investigation.

2) Quartic and Rastrigin Regression: The quartic regression instance (using the quartic polynomial) 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

TABLE I
EXPERIMENT AND PROBLEM PARAMETERS. NOTE THAT BOTH REGRESSION INSTANCES' FUNCTION SET INCLUDES THE SAME FUNCTIONS (SIN,COS,EXP,LOG) AND THAT "P/" IS PROTECTED DIVISION IN BOTH THE QUARTIC AND RASTRIGIN INSTANCES, RETURNING 1.0 IF THE DENOMINATOR IS 0.0

Parameter	Value
evolutionary model	generational genetic algorithm
population size	500
fitness functions	see Section III-A
stop criterion	maximum generations (51)
function sets	
ant	{if_food_ahead,progn2,progn3}
parity	{and,or,nand}
quartic, Rastrigin	{+,-,*,p/,sin,cos,exp,log}
terminal sets	
ant	{left,right,move}
parity	{D0,D1,D2,D3,D4}
quartic, Rastrigin	{x} (as defined in Sec. III-A.2)
tree generation	ramped half-n-half
initial depth	4
maximum depth	10
subtree crossover probability	1.0
mutation probability	0.0
internal node selection probability	0.9
maximum generations	51
parent selection	tournament, size 4
ECJ version	7.0

the objective function and the function produced by the individual. The Rastrigin instance is similar to the quartic instance where the function is

$$f(x) = 3.0n + \sum_{i=1}^{n} x_i^2 - 3.0\cos(2\pi x_i).$$

For the Rastrigin instance, x is in the range $[-5.12, 5.12]$ and for the quartic instance, x is in $[-1.00, 1.00]$, while $n = 20$ for both instances. Both problems use the same function and set of addition, subtraction, multiplication, protected division (returning 1.0 if the denominator equals 0), sine, cosine, exponentiation, and logarithm. Their common terminal set includes the original functions' x values, 20 randomly sampled points for both problems. The function set used here is typical for the Rastrigin instance, whereas the quartic instance occasionally uses only addition, subtraction, multiplication, and division. We use the same for both to be consistent and do not use any ephemeral random constants. Note that by keeping the function and terminal sets the same for both regression instances, the Rastrigin problem is likely to be more difficult to solve without using ephemeral random constants. Also, in this paper, we will often refer to the "quartic problem" and "Rastrigin problem" when they are indeed instances of the same problem domain: the regression problem.

3) Even-5-Parity: The even-5-parity problem takes an input of a random string of 0s and 1s and outputs whether there are an even number of 1s. 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 ($0 = \text{best fitness}$). The function set consists of the binary *and, or nand* functions and there are the five terminals *D0, D1, D2, D3,* and *D4* representing the boolean inputs. The parity problem has also been investigated in detail in [30].

B. Diversity Measures Used

Our experimental study uses several measures of diversity that were introduced in Section II. With the following measures, we attempt to asses their relationship with performance and use them as a way to view population dynamics. The measures are collected for each population in every generation.

- **Genotype** diversity counts the number of unique trees [29]. Genotype diversity does not consider the fitness or behavior of the trees. Two trees are equal only if they contain the exact same structure and content.
- **Phenotype** diversity counts the number of unique fitness values in a population [45]. This measure is quite important, as we will see later, as the selection mechanism which must choose individuals to produce the next generation selects individuals based on their fitness. Different problem domains define the number of possible fitness values differently. For instance, in the Parity problem, there is a finite number of possible fitness values that an individual can have. However, the fitness space is continuous in regression problems, but due to the precision of numbers, wrappers around operators (protected division for instance), and the presence of nonfunctional code it is common for different trees to have the same fitness value.
- **Entropy** diversity is calculated for the population as in [44], 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. This would be most appropriate for the continuous fitness space problems. However, for these problems, it would be equally valid to define the phenotype measure in this way, but both tasks would require a deeper understanding of the possible fitness values. Entropy represents the amount of chaos in the system, where high entropy describes the presence of many unique fitness values where the population is more evenly distributed over those values. Low entropy describes a population which contains fewer unique fitness values and many individuals have the same fitness.
- **Pseudo-isomorphs** are found by defining a three-tuple of $\langle terminals, nonterminals, depth \rangle$ for each individual and the number of unique three-tuples in each population is the diversity measure. Two identical three-tuples represent trees which could be isomorphic and two nonidentical three-tuples are not isomorphic. To determine if the trees are indeed isomorphic would be too computationally expensive.

- **Edit distance 1 and 2** diversity is based on the edit distance between individuals used by de Jong *et al.* [12] (referred to as "ed 1" in the graphs) and an adapted version of the approach used by Ekárt and Németh [18] ("ed 2"). Every individual in the population is measured against the best fit individual found so far in the run. This measure is then divided by the population size. The first measure (denoted "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 score 0. The edit distance is then the sum of all different nodes which is normalized by dividing it by the size of the smaller tree. The second measure (denoted "ed 2") is adapted back to its original formulation in [41] where the difference, $d(p, q)$ between any two nonequal nodes p and q is 1. The difference between two trees is then (defined in [18])

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

where T_1, T_2 are trees with roots p, q and possible children (m total) subtrees s, t, and $\mathrm{K} = 1/2$. The constant K is set to 1/2 but can be adjusted, as done by Ekárt and Németh [18], to weight the depth of tree differences differently. 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. This is possibly a very convenient description for genetic programming as it has been noted that programs converge quickly to a fixed root portion [25], [40]. Also, note that our edit distance diversity measures the population against the individual with the best fitness in the run so far, not the one with the best fitness in the current population, a distinction that was less clear in [5]. The reason for this is that it is common for researchers to consider this individual rather than the best in each generation for analysis. Additional experiments using edit distance based on the current generation's best of run individual yielded little variation.

IV. Results and Analysis

First, we examine the primary results of the experiments, focusing on trends that populations exhibit when viewed with best fitness and diversity measures. We then attempt to present a more general analysis of how effective our diversity measures are, what diversity tells us about evolving populations, and how these results support previous results and conjectures.

A. General Comments on Sample Runs

We begin by viewing 50 of the random independent runs, with one graph for each problem and for selected diversity measures.

Fig. 1 shows the best fitness of each generation during the evolutionary process, and Fig. 2 shows the evolution of size and depth for all problems. Many runs stop improving after 15–20 generations, with the exception being the Parity problem which continues to make improvements. Previous research by Luke [32] showed that it is better to carry out short runs (above a *critical point*) than fewer long runs for the ant and quartic problem. Luke also found that with the parity problem (even-10), one long run was actually better, because of the difficulty of the problem and the ability of genetic programming to consistently make improvements. This *critical point* was around generation 8 for the quartic problem and slightly higher for the ant problem. With this period in mind, we now look at several measures of diversity for the same runs.

An early period of higher activity in the runs also exists with respect to diversity measures. Note that in Figs. 3–5 there is typically a lot of activity in the early generations and not too much after generation 30. In these graphs of diversity measures, populations begin with similar values, and during the initial generations branch off to lower and higher diversity values with generally lower fitness.

The phenotype diversity in Fig. 3 of the quartic and Rastrigin instances (which have continuous fitness spaces) shows an initial decrease followed by a sharp increase, whereas the ant and parity problems show only an increase in initial populations. This behavior was also seen with genotype diversity and entropy, an initial sharp decrease followed by an increase within the quartic and rastrigin problems and in all problems with genotype diversity. Intuitively, the cause of this initial fluctuation is due to the population *settling* after the selection and recombination of initial populations, where differences are due to problem representations. This initial phase highlights these differences. Also, note that phenotype diversity for the parity problem continues to increase until the final generation.

The edit distance in Fig. 4, for all problems, generally decreases after the initial generation. Also, in Fig. 4, the populations measured with edit distance 2 behave similarly (note that only the averages are graphed in Fig. 5). With this in mind, and because the edit distance 2 measure places more importance on the root and higher portions of trees, we can conclude the following: While trees are changing (according to edit distance 1) to be more like the best fit tree in each population, the differences between the roots and top portions of the tree also become more similar (according to the edit distance 2 measure). This supports previous conclusions [25], [40], [49] that roots become fixed early on in the evolutionary process. Structural convergence is important when considering using a method to control diversity. If structural convergence is beneficial to genetic programming search, then encouraging or forcing structural diversity (edit distance in this case) could have negative consequences. However, the loss of edit distance diversity does not necessarily mean a loss of phenotypic diversity or the worsening of fitness, as seen in Figs. 1 and 3.

The last comment on these figures is the observable behavior that in some runs (most notably in the ant problem) fitness continues to increase until the final population. Identifying the properties of these populations that allowed for this continued increase is critical for genetic programming practitioners. And this is one of the goals of this paper: understanding how to make populations more amenable to improvement. Given the wide range of fitness and diversity, we would like to know if these measures

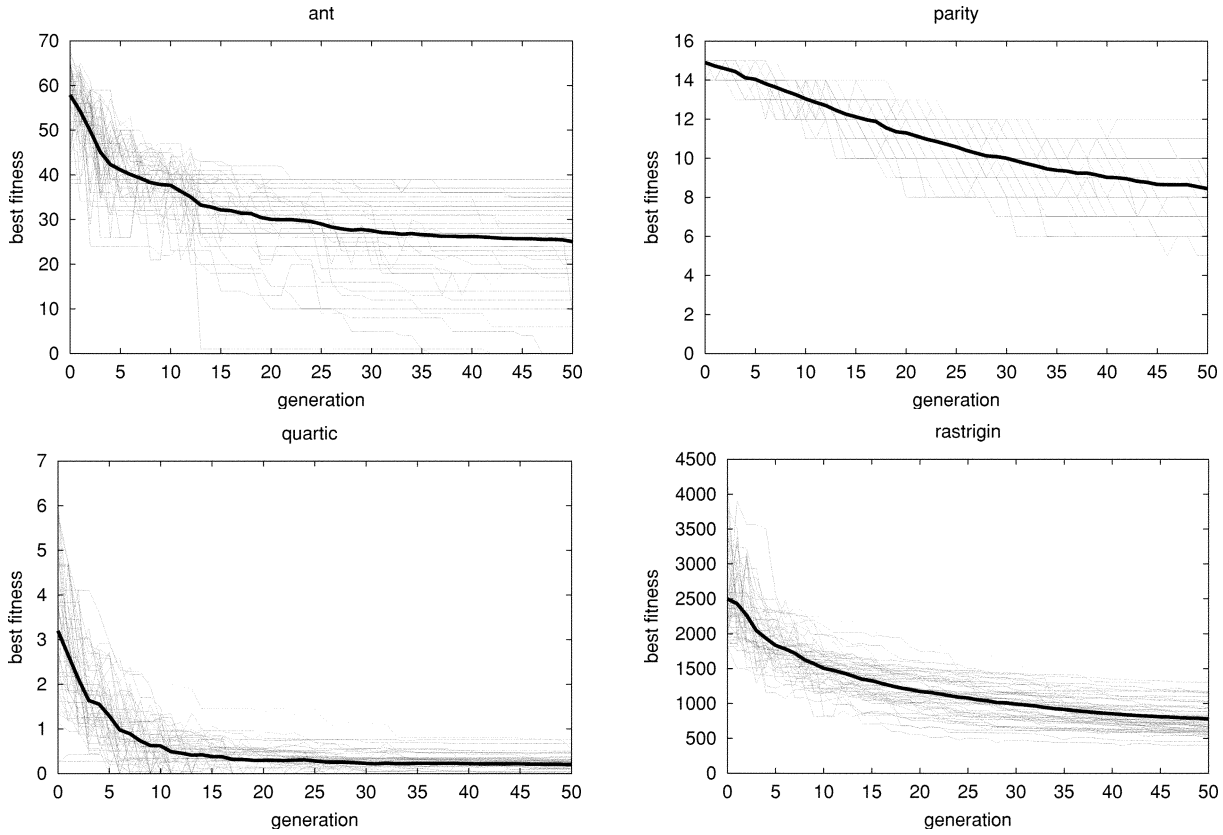

Fig. 1. Ant, parity, quartic, and Rastrigin best fitness per population, plotted against the generation number. Fifty independently random runs of each problem are shown.

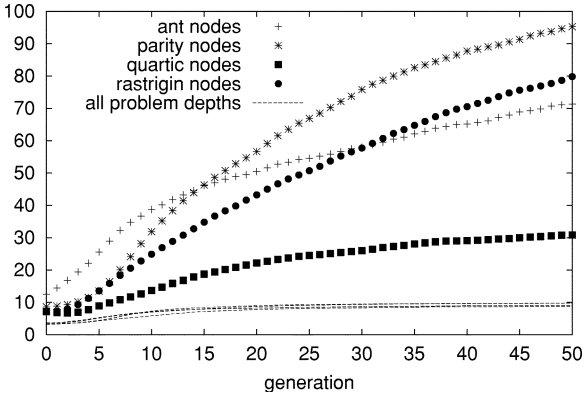

Fig. 2. Average depth and average number of nodes in an individual in each generation, averaged over 50 random independent runs. Note that all problems evolved individuals which quickly reached the maximum depth of 10 around generation 15. The quartic instance showed the largest variance and, thus, the lowest average number of nodes. Since the quartic instance is the easiest to solve, we suspect that this also leads to smaller trees.

correlate in any way. Addressing this question is key to understanding if controlling diversity is likely to be effective and how it should be applied on different problem domains.

B. Correlations in Final Populations

We initially look at the correlation of diversity and fitness in the final generation of each run. We limit our analysis to samples of size 100. Table II (with four, problem specific subtables) summarizes the Spearman correlation coefficients between fitness and diversity, and also between diversity measures.

In the ant problem (Table II) negative correlation is seen between phenotypes and fitness, and also between entropy and fitness. As one might expect, good (low) fitness is seen with high phenotype diversity and entropy. There is a positive correlation of edit distance with fitness and also between pseudo-isomorphs and fitness. Only very weak correlation is seen between genotypes and fitness on the ant problem, which is the trend for all the problems. In this case, a positive correlation between fitness and edit distance and fitness and pseudo-isomorph correlation with fitness suggests that low (good) fitness is seen with low diversity. As we know from Figs. 4 and 5, edit distance generally decreases during the run. While runs tend to structurally converge for the ant problem, and with respect to the edit distance 1 measure in the parity problem, those which converge more often have better fitness.

The last table of correlation coefficients in Table II gives the Rastrigin problem results. This table shows the lack of strong correlations between diversity and fitness (the same effect is partially seen in the quartic instance as well). It may be the case that a correlation did exist between fitness and diversity, but final populations have lost any correlation due to the repeated application of selection and recombination without change in fitness.

The importance of phenotypic diversity is now seen with the parity problem in the second part of Table II, where a strong negative correlation exists with fitness and phenotype diversity. Fig. 3 shows that phenotype diversity tends to increase in the parity problem. With only 32 possible fitness values in

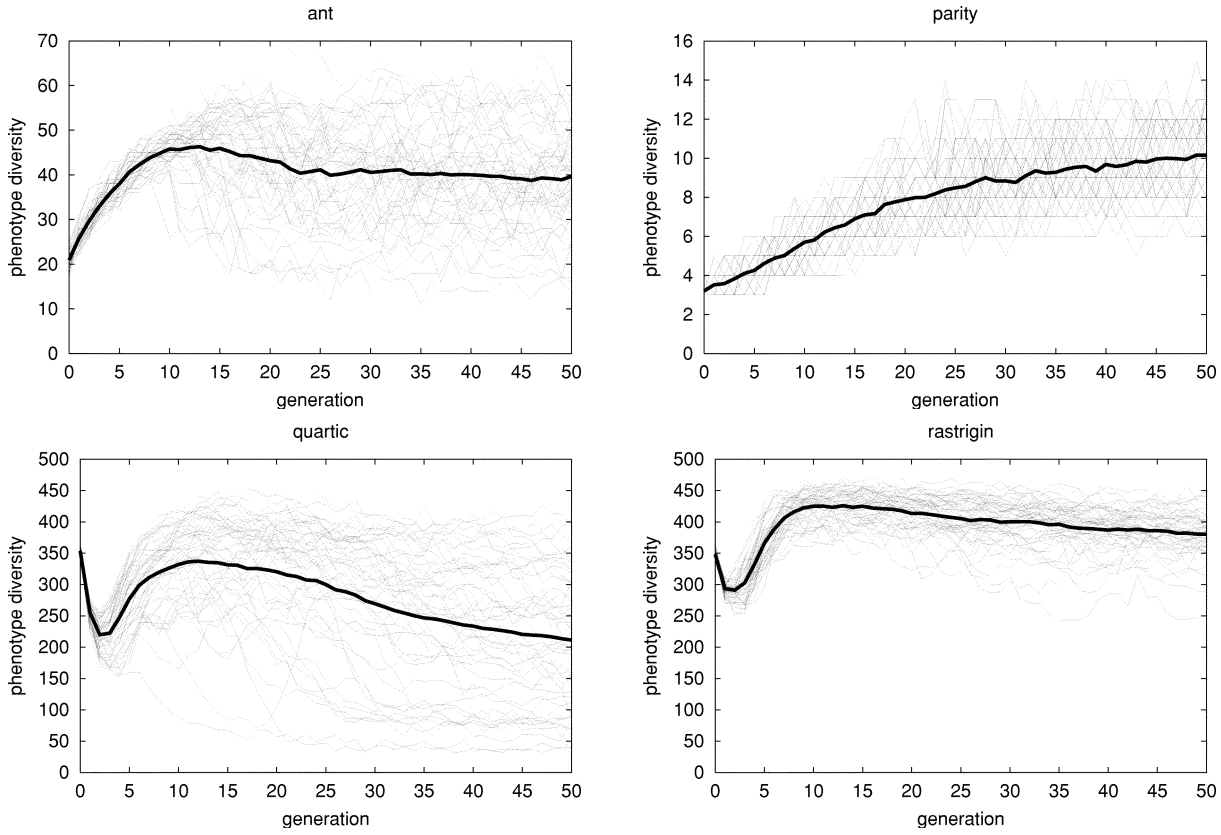

Fig. 3. Ant, parity, quartic, and Rastrigin phenotype diversity, plotted against the generation number. Fifty independently random runs of each problem are shown.

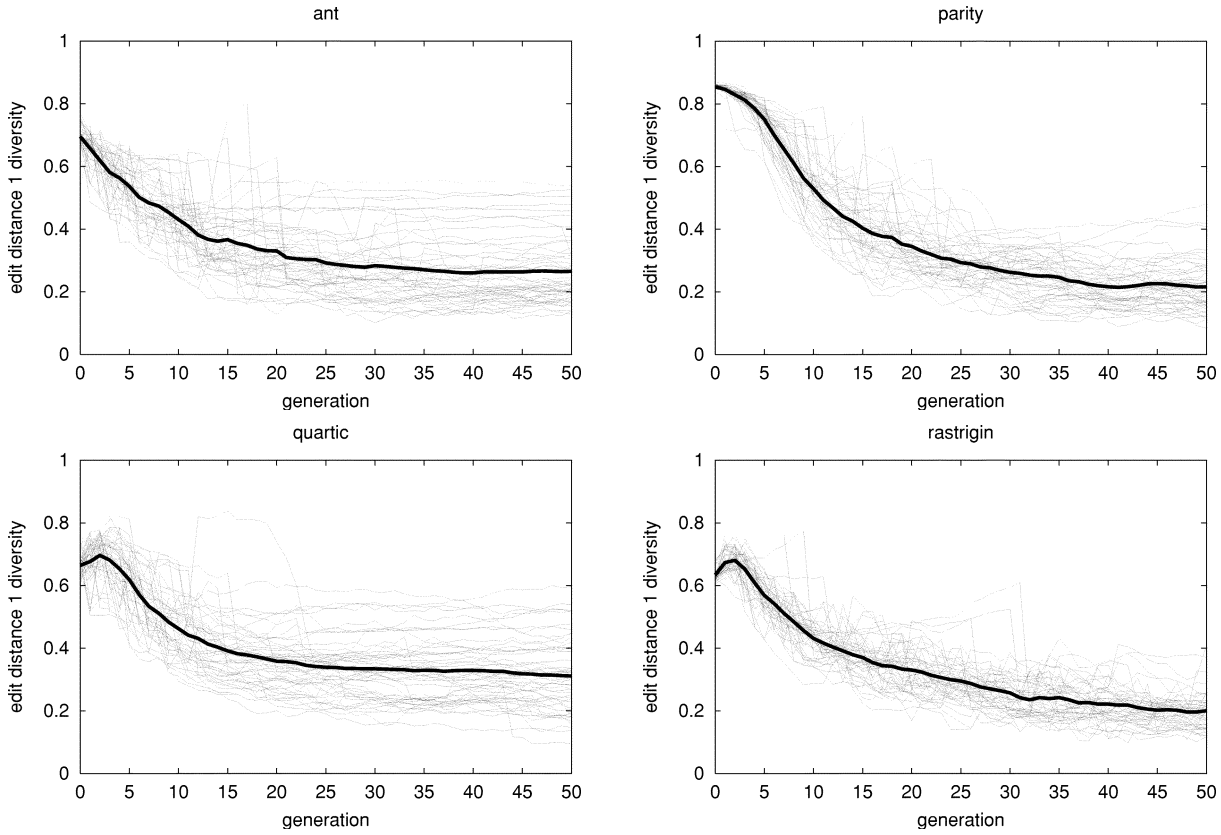

Fig. 4. Ant, parity, quartic, and Rastrigin edit distance 1 diversity plotted against the generation number. Fifty independently random runs of each problem are shown.

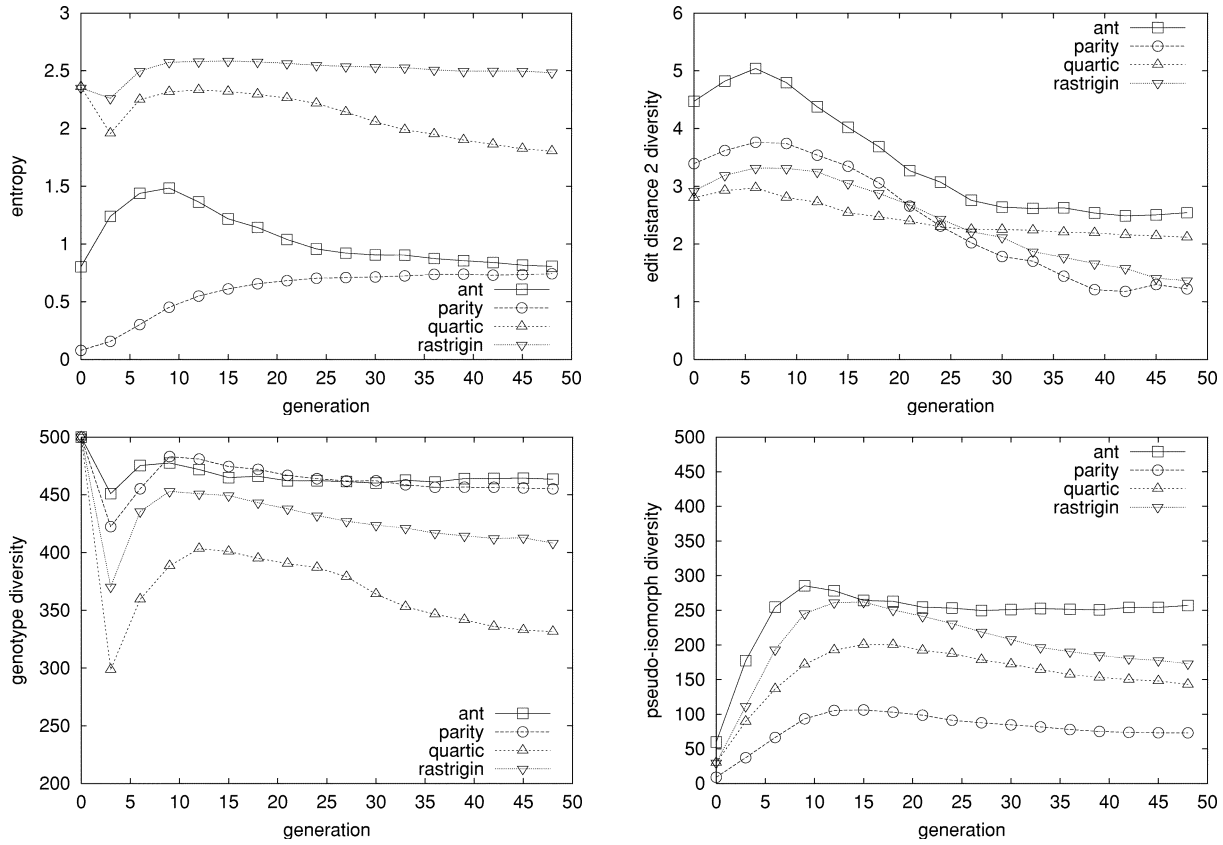

Fig. 5. Average of 50 runs of the entropy, edit distance 2, genotype, and pseudo-isomorph diversity measures for ant, parity, quartic, and Rastrigin.

this problem, the population begins with random guesses with approximately a fitness of 15. As populations undergo selection and recombination, the number of unique fitness values increases from 3–4 to 6–13. Without some increase in phenotypic diversity, genetic programming cannot distinguish between good individuals and bad ones. Thus, it is critical that the number of unique fitness values is increased for the parity problem.

Another effect of high phenotype diversity and entropy is the relationship it has with the selection pressure. As tournament selection uses the fitness values of an individual to decide tournaments, the less unique phenotypes in the population (and the lower the entropy) will make selection more *random*. That is, selection will be faced with many individuals that have the same fitness. Therefore, if a high phenotypic diversity and entropy is maintained, selection pressure remains at the preset level. The lowering of phenotype diversity and entropy might actually benefit some problems where less selection pressure is suitable, but negatively affect others where higher selection pressure is better.

Table II also gives the correlation between the measures of diversity. In the ant problem, note that more phenotype diversity negatively correlates with the structural measures (genotypes, pseudo-isomorphs, and the edit distances). An increase (or decrease) of unique fitness values in the population corresponds with a decrease (or increase) in the structural diversity. This seems counterintuitive as *more* unique structures should correspond with *more* unique fitness cases. We expect this behavior with the edit distance measures as we know that these measures generally decrease during evolution, while phenotype diversity increases. In this problem, the discovery of different fitness values appears to be aided by less structural diversity. That is, if the population is structurally similar, it is easier to find more unique fitness values.

C. Evolving Populations' Correlation

Does diversity play a more significant role at different times of evolution? The fact that several methods have been previously used to adaptively control the level of diversity would suggest so. Fig. 6 shows the correlation between diversity and fitness for each generation. Note that each point represents the correlation between 100 populations, sampled from 100 runs. Thus, there is a dependency of later generations on preceding ones, but this is what we are interested in observing. We would expect to see less activity in changes in correlation between fitness and diversity toward the end of runs, as fitness usually stops improving before this point. Also, Fig. 5 uses only random runs, where as a similar graph in [5] considered populations from nonrandom experiments, ones that were predictively poor.

Both the ant and parity problems contain varying levels of correlation for edit distance with fitness (ant) and also for phenotype diversity with fitness. The quartic problem contains a period of early fluctuation, followed by an increase in positive correlation between entropy (and phenotype diversity) and fitness. As runs typically achieve the best fitness early, we think this effect is due to many copies of the best fit individual accumulating in the population. That is, populations which achieve good local optima begin to have lower entropy.

TABLE II
ANT, PARITY, QUARTIC, AND RASTRIGIN PROBLEMS. CORRELATION BETWEEN BEST FITNESS IN LAST GENERATION AND THAT POPULATION'S DIVERSITY MEASURE IN THE FIRST COLUMN. THE OTHER COLUMNS SHOW THE CORRELATION BETWEEN THE LAST POPULATION'S DIFFERENT DIVERSITY MEASURES. THE SAMPLE SIZE IS 100 INDEPENDENT RUNS

Ant	fitness	phenes	genes	p-isom	entropy	ed 1
phenes	-.3936	–	–	–	–	–
genes	.1962	-.4950	–	–	–	–
p-isom	.4009	-.6389	.6949	–	–	–
entropy	-.3615	.9039	-.5724	-.7569	–	–
ed 1	.4205	-.5040	.2991	.3998	-.4891	–
ed 2	.4606	-.4537	.4702	.5603	-.4949	.7504

Parity	fitness	phenes	genes	p-isom	entropy	ed 1
phenes	-.7803	–	–	–	–	–
genes	-.0641	.0510	–	–	–	–
p-isom	.0773	.0646	.5132	–	–	–
entropy	-.7146	.7048	-.0379	.0204	–	–
ed 1	.3235	-.2156	.1178	.4483	-.3062	–
ed 2	.0148	-.0087	.2656	.5377	-.0626	.7265

Quartic	fitness	phenes	genes	p-isom	entropy	ed 1
phenes	.4345	–	–	–	–	–
genes	-.1363	-.0353	–	–	–	–
p-isom	-.0300	.1588	.8408	–	–	–
entropy	.3924	.9730	-.1712	.0070	–	–
ed 1	-.1640	.0045	.2290	.3150	-.0191	–
ed 2	-.0881	-.0273	.1554	.2182	-.0461	.6891

Rastrigin	fitness	phenes	genes	p-isom	entropy	ed 1
phenes	-.0616	–	–	–	–	–
genes	-.1305	.7089	–	–	–	–
p-isom	-.2262	.5521	.6163	–	–	–
entropy	-.0402	.9688	.7324	.5525	–	–
ed 1	-.0530	-.0365	.2056	.2014	.0460	–
ed 2	-.0762	.1185	.3265	.3828	.1750	.6514

The Rastrigin problem contained an early period of varying correlation between diversity and fitness before most measures lost correlation with fitness. In this problem and representation, the relationship between fitness and diversity becomes less important, probably due to other, more critical relationships like node-to-node dependencies [9]. As we noted in [5], a positive correlation between fitness and edit distance occurs together with a negative correlation between fitness and phenotype diversity. This behavior is seen to some degree in all problems, most notably in the ant and parity problems. These results suggest that the fitness landscape induced by the representation and operators is uncorrelated. Small differences between individuals are still capable of expressing a wide range of behaviors. However, this statement should be considered in the light of the operator not being used to define distance and the actual difference between behaviors is not being measured. The measures used here can only approximate the fitness landscape.

D. Scatter Plots of Diversity and Fitness

The Spearman correlation coefficient only describes linear relationships, so we now examine a series of scatter plots. Figs. 7 and 8 plot a population's performance (best fitness found in the population is plotted along the x axis, where values to the left are better) versus that population's diversity (on the y axis). Each point represents a population sampled from a different run, where no run is used twice and 10 populations are sampled for each generation, requiring 500 runs. Also, note that all points for the parity problem have their fitness values randomly offset in the range of $[-0.2,0.2]$ so the number of populations at each fitness value can be seen.

A few general comments can be made about the scatter plots in Figs. 7 and 8. There are clear trends of fitness occurring with lower edit distance and with higher entropy. However, many populations with low fitness also have a wide range of entropy (Rastrigin and quartic) and edit distance (quartic). The ant problem, in particular, shows a transition from high to low fitness with populations in the middle containing a wide range of entropy and edit distance values. The populations which achieve the lower fitness also have lower entropy and edit distance. It is likely that this problem suffers the most from local optima, where populations stuck with suboptimal individuals also have suboptimal diversity. Too high edit distance diversity and either too-low or too-high entropy would appear to be suboptimal for the ant problem.

An important observation is that better populations tend to occur near the end of evolution and resulting populations will be less diverse simply because of our search and selection mechanisms. We can see in Fig. 8 that when populations have large edit distances they are unlikely to have better fitness values. The reason for this is that in our experiments, large edit distances only occur at the beginning of runs. As we are attempting to understand genetic programming populations better, the question of whether these populations always occur late in evolutionary process is analyzed next.

For Fig. 9, we use the same populations from Fig. 8, except now the z axis shows a vertical line representing the generation in which that population occurred. A common trend is that the worse fit populations occur in early generations, which is to be expected as Fig. 1 showed fitness to always improve (decrease in value) initially. In general, as we move from right to left in fitness values (from worse to better), the lines get taller on the z axis. However, it is not the case that the best populations are at the end of runs for all problems. We can see many populations where good fitness occurs early and in the middle of runs. Furthermore, Fig. 9 emphasises that populations have different diversity at similar times in the evolutionary process. Later evolutionary periods do not always imply high or low values of diversity and fitness.

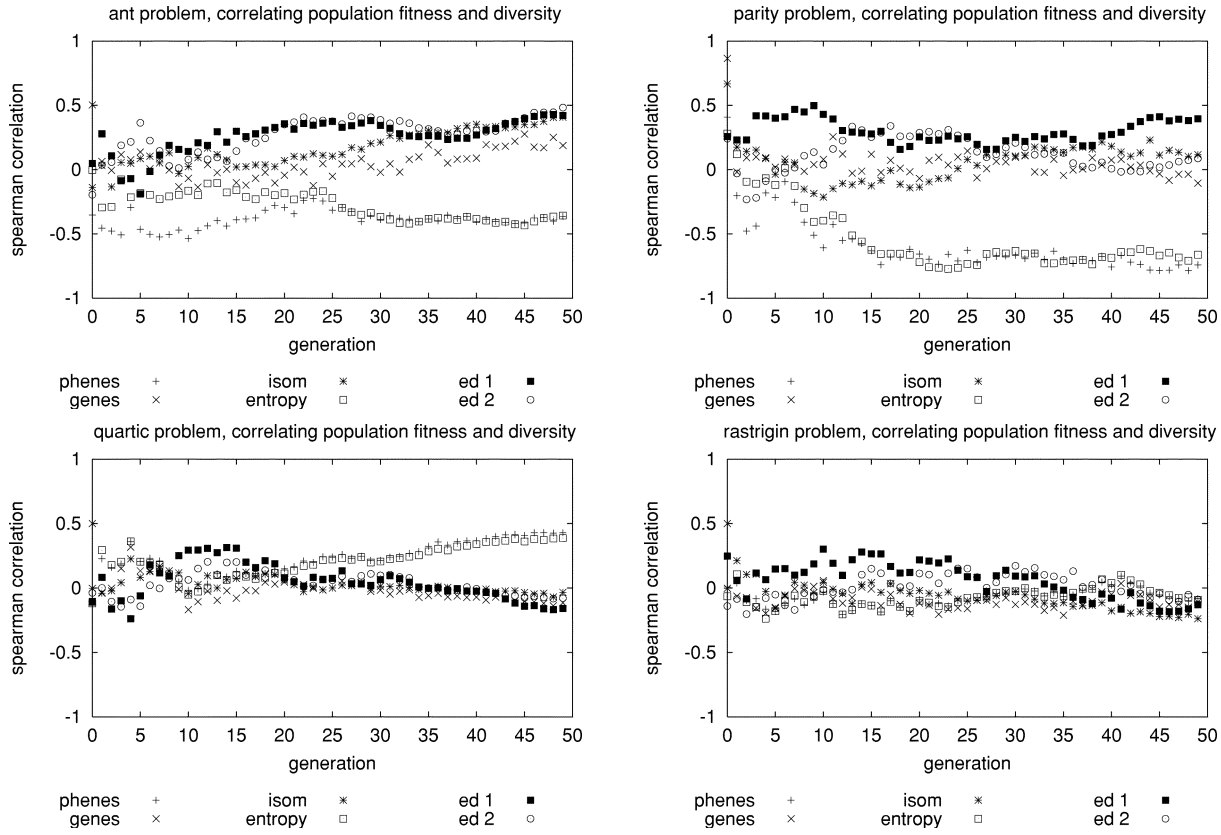

Fig. 6. Evolving populations' correlation between best fitness in each population and different diversity measures. Each point represents the correlation between 100 populations from a 100 runs, each 50 generations are represented.

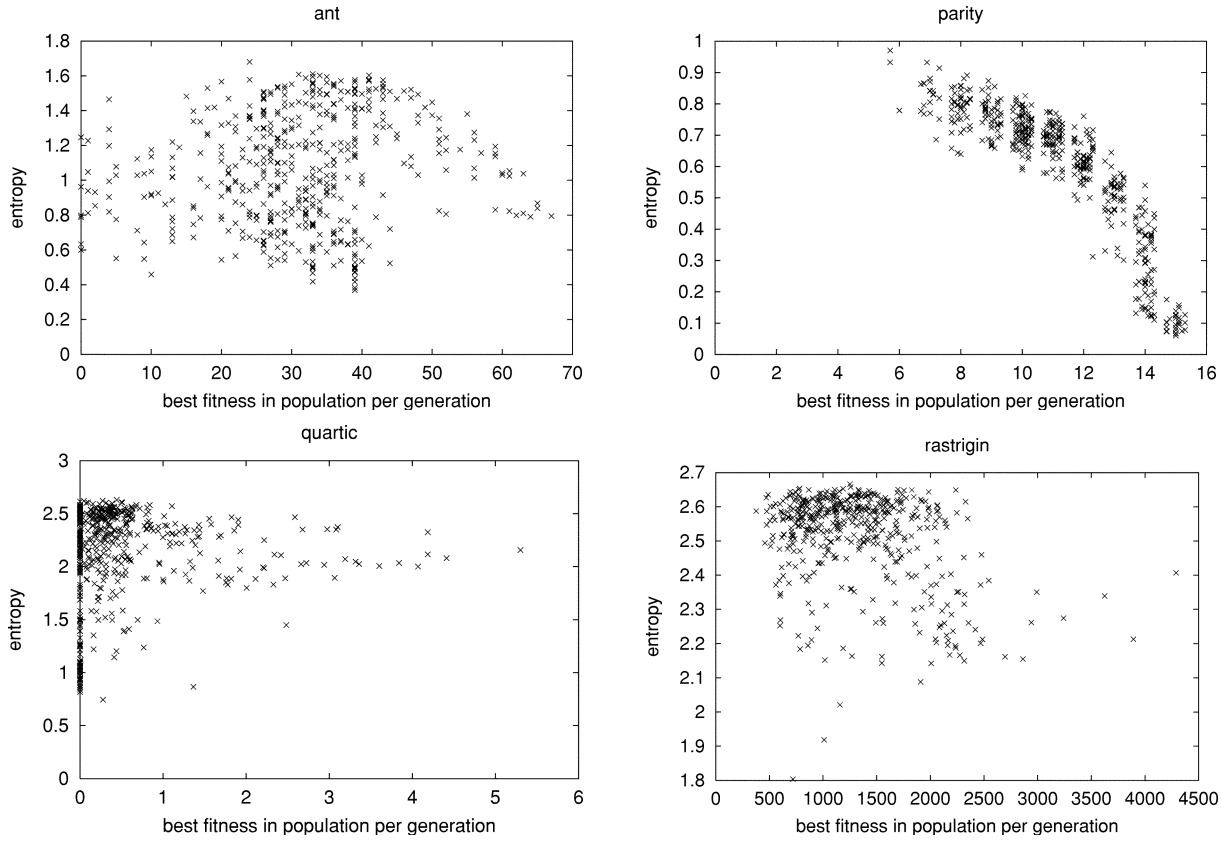

Fig. 7. Ant, parity, quartic, and Rastrigin best fitness per population plotted against that population's entropy diversity. Note that each point represents one population from each run. We sample ten different runs for each population at generation g, requiring $50 \times 10 = 500$ runs for all 50 generations.

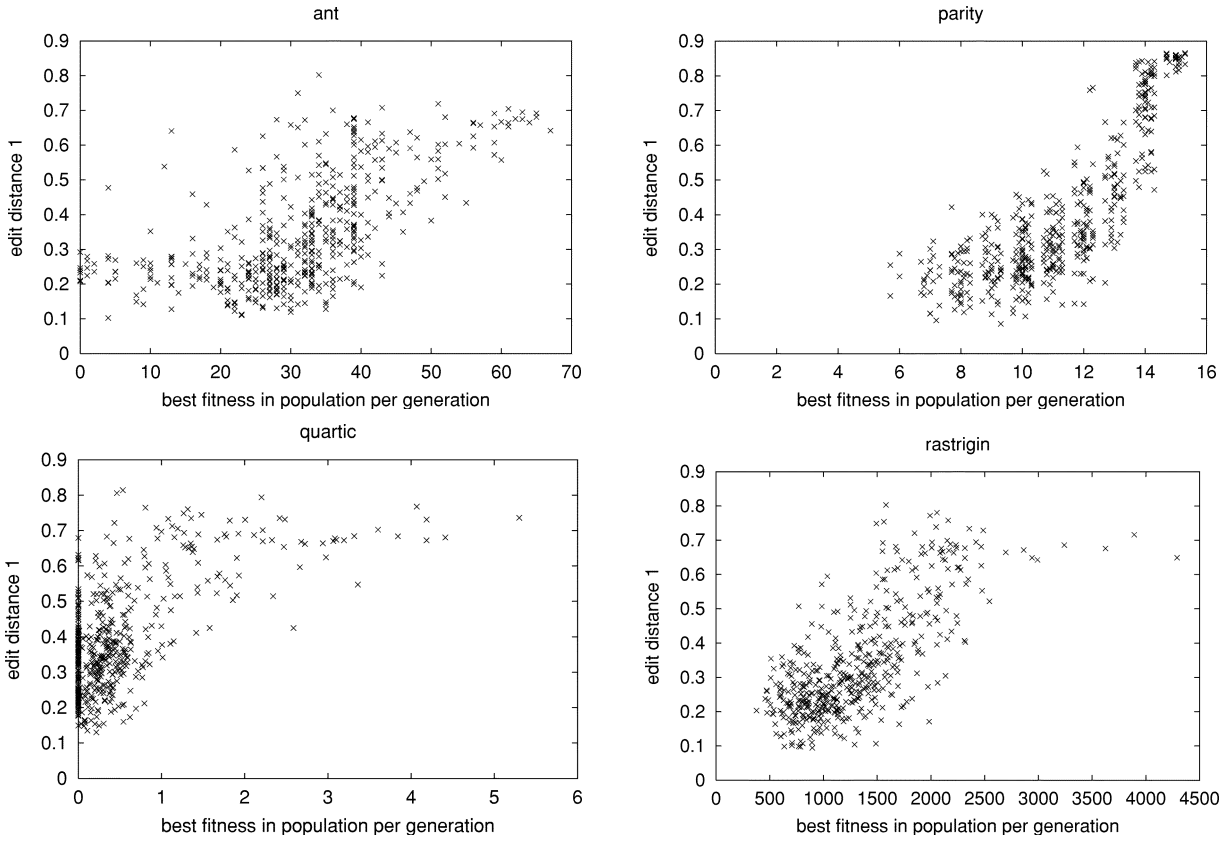

Fig. 8. Ant, parity, quartic, and Rastrigin best fitness per population plotted against that population's edit distance diversity. Note that each point represents one population from each run. We sample ten different runs for each population at generation g, requiring $50 \times 10 = 500$ runs for all 50 generations.

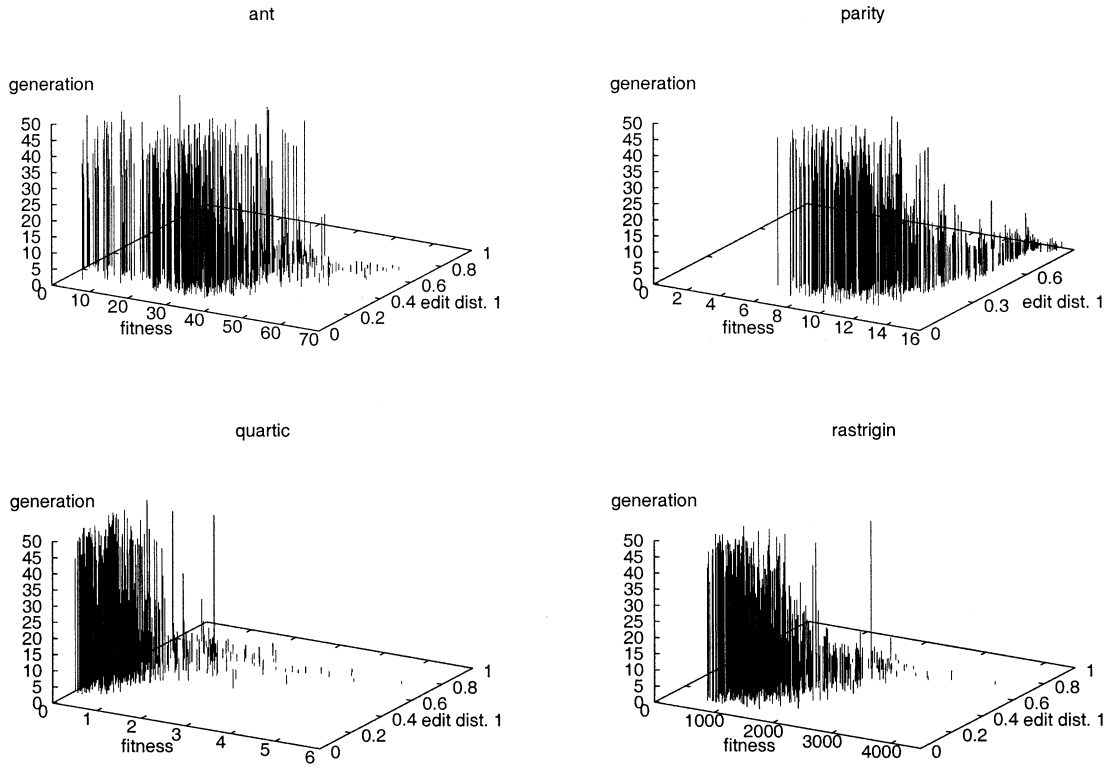

Fig. 9. Ant, parity, quartic and Rastrigin best fitness per population (x axis) plotted against that population's edit distance 1 diversity, (y axis) and the generation the population occurred (z axis). Note that each point represents one population from each run.

V. Discussion

The measures chosen to study here (and in [4] and [5]) are in a sense related hierarchically with respect to the amount of information they contain about the population. The edit distance measures provide a fine grain description of population structural differences, pseudo-isomorphs give a more abstract view of the population and the genotype diversity measure simply describes the number of absolutely different trees. Entropy and phenotype diversity are similarly related. Entropy not only describes the number of unique phenotypes, but also how the population is distributed over the existing phenotypes. Also, the experimental study presented here shows the most consistent correlation between edit distance and fitness and between entropy and fitness (suggesting they capture an important element in the genetic programming search process). The pseudo-isomorph diversity measure was used to capture a level of information that is more specific than genotype diversity, but less expensive than edit distance. Our initial investigation of this measure in [4] and the results here show that it can express stronger correlations than genotype diversity and is generally more correlated to edit distance measures than genotype diversity.

The experiments used different measures of diversity and have enabled us to analyze not only the measures and how they correlate with fitness, but also the behavior of standard genetic programming on commonly used problems. Results showed additional evidence that the roots of trees become fixed very early on in the genetic programming evolutionary process and are unlikely to change. This has been demonstrated by previous research [25], [40], [49] and is supported here by the edit distance diversity measures.

We have previously mentioned the importance of phenotypic diversity and entropy due to the ability of selection to distinguish between individuals better and maintain a constant level of pressure. Depending on the problem and behavior of the current run, the increase and decrease of phenotypic and entropy diversity is likely to be crucial at different stages of evolution. This emergent change of selection pressure due to the loss of entropy could be beneficial in helping to avoid local optima for some problems. The constantly fluctuating values of phenotype diversity in Fig. 3 could be demonstrating this behavior. However, based on our experiments and analysis, it is not clear if this is necessarily the case.

The Spearman correlation coefficient [47] shows a positive correlation between fitness based diversity and fitness, and a negative correlation between edit distance diversity and fitness. We hypothesise that this is the result of the following: More structurally similar populations create a neighborhood in which crossover is likely to find better neighbors. Crossover initially works with very *unlike* structures until a significantly good one is found. Then, combined with the selection pressure, the population begins to resemble this good individual as crossover repeatedly combines more and more like individuals. Success at this point suggests that crossover is able to work within this population structure to find better solutions. We have seen here and in [4] and [5] how quickly diversity is lost. It appears that this crossover-friendly neighborhood occurs early in the evolutionary process, but might also be responsible for leading the search toward inescapable local optima rather quickly. The point here is not to argue that crossover is (or is not) a sufficient operator for search in tree-based genetic programming, but to show (in cases where genetic programming is solving problems) how populations and recombination operators may be working together.

However, just as the correlation coefficient suggests associations between diversity and performance, it should not be used to infer causation between variants, i.e., higher diversity does not necessarily *cause* better performance but better performance is seen *with* higher diversity (phenotypic diversity here). This should apply to all conclusions about diversity. Caution should also be taken considering that the search mechanism's recombination and selection methods play an extremely important role in shaping individuals and populations. Very simple implementation differences can drastically increase or decrease diversity measures. Models of causation based on diversity results should be defined carefully.

Standard genetic programming is often compared with a blind local search or a hill-climber, due to the loss of diversity and the attraction to local optima [22], [40], [43]. The results presented here with diversity also support this phenomenon with lopsided exploration and exploitation phases. After an initial period of adjustment to different problem representations and selection, the populations appeared to converge toward less structural diversity. These initial few generations of each run appear to represent the exploration phase, while the latter part of the run is concerned with exploiting the better individuals found. Adaptive controls of diversity, selection pressure or mutations could be used to extend the exploration phase to allow more global search. However, they should also be aware of the initial *settling* behavior observed here, which might be the process of vetting poor individuals.

Researchers have shown that encouraging different amounts of diversity can lead to better performance (for example [46]). Based on our results, we hypothesise that the strong exploitation of structures occurs in almost all runs (populations consistently converge on a common structure), but not all runs exploit good structures. Thus, genetic programming may be exploiting structures which are not amenable to further improvements with respect to the existing population and the algorithm. If our algorithm backtracked upon finding a bad structure, or made a concentrated effort to find a good structure, it could be argued that we would be more likely to exploit the better structures which lead to better performance. In essence, by either increasing the length of exploration or adaptively exploring in later phases, local-optima may be avoided more effectively. This is the effect that we think has been achieved in previous work, while improvements are being made with fitness, populations are allowed, or forced, to exploit that structure. However, when no improvements are made, then populations are pushed to become more diverse and try other structures. Increased population sizes, higher levels of mutation, and models which prevent the overall convergence of populations (such as islands, demes or distributed models) could achieve this effect.

VI. Conclusions and Some Future Directions

This paper has provided a survey of measures used to capture diversity in genetic programming and of the methods employed to control diversity. An experimental study enabled an analysis of the correlation between selected measures of diversity and fitness. The results showed three important behaviors.

1) The generation to generation behavior of specific diversity measures is problem specific. In fact, representation changes of the same problem are likely to have different diversity behaviors. Thus, the pursuit of a single measure with which to control diversity in order to improve fitness is likely to be difficult.
2) Entropy and edit distance diversity showed strong correlation with fitness. This is likely to be related to an emergent change of selection pressure and the level of structural convergence which allows a form of hill-climbing search.
3) Regression problems had the weakest correlation between any measure of diversity and fitness overall, suggesting that the things that make these populations achieve good fitness may not be captured by any of the measures used here.

The introduction of different recombination operators, large changes in parameter values and applications on different problem domains are all likely to effect the results and interpretations made here. However, the methodology of using several informative and complimentary measures of diversity should allow one to gain a deeper understanding of the search space and algorithm. As search spaces become larger and more complex, fine grain measures will become too inefficient. Therefore, using measures which capture the right level of information while still being efficient will be critical. Based on these results, we make the following recommendations. Before applying new methods to control diversity in order to improve fitness the correlation between fitness and diversity should be investigated. Knowledge of this correlation can help to enhance the diversity measure and method and give insight into results, taking care to distinguish between correlation and causation. Also, when a many-to-one relationship exists between the genotype and phenotype encoding, measures which are based on genotype uniqueness will probably not be as useful as those which capture phenotype uniqueness.

Our future research is looking at various methods used to control diversity and the effects of higher and lower diversity on different problem domains. Additionally, we are attempting to incorporate more knowledge of operators (subtree crossover) into existing diversity measures while preserving their efficiency.

Acknowledgment

The authors would like to thank N. Krasnogor, W. Hart, A. Ekárt, and D. Gustafson for conversations about diversity. We would also like to thank the anonymous reviewers for their insightful comments. S. Gustafson would like to thank K. Gustafson.

References

[1] T. F. Bersano–Begey, "Controlling exploration, diversity and escaping local optima in GP," in *Proc. Late Breaking Papers at the Genetic Programming Conf.*, J. R. Koza, Ed., Stanford Univ., Stanford, CA, July 1997, pp. 7–10.

[2] M. Bessaou, A. Pétrowski, and P. Siarry, "Island model cooperating with speciation for multimodal optimization," in *Proc. 6th Int. Conf. Parallel Problem Solving From Nature*, M. Schoenauer *et al.*, Eds., Paris, France, 2000, pp. 437–446.

[3] M. Brameier and W. Banzhaf, "Explicit control of diversity and effective variation distance in linear genetic programming," in *Proc. 5th Eur. Conf. Genetic Programming*, vol. 2278, LNCS, Kinsale, Ireland, Apr. 2002, pp. 162–171.

[4] E. Burke, S. Gustafson, and G. Kendall, "A survey and analysis of diversity measures in genetic programming," in *Proc. Genetic and Evolutionary Computation Conf.*, W. B. Langdon *et al.*, Eds., New York, July 9–13, 2002, pp. 716–723.

[5] E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor, "Advanced population diversity measures in genetic programming," in *Proc. 7th Int. Conf. Parallel Problem Solving From Nature*, vol. 2439, LNCS, J. J. M. Guervós *et al.*, Eds., Granada, Spain, Sept. 2002, pp. 341–350.

[6] R. J. Collins, "Studies in artificial evolution," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Los Angeles, CA, 1992.

[7] J. M. Daida, R. R. Bertram, J. A. Polito II, and S. A. Stanhope, "Analysis of single-node (building) blocks in genetic programming," in *Advances in Genetic Programming 3*, L. Spector *et al.*, Eds. Cambridge, MA: MIT Press, 1999, ch. 10, pp. 217–241.

[8] J. M. Daida, R. R. Bertram, S. A. Stanhope, J. C. Khoo, S. A. Chaudhary, O. A. Chaudhri, and J. A. Polito II, "What makes a problem GP-hard? analysis of a tunably difficult problem in genetic programming," *Genetic Program. Evol. Mach.*, vol. 2, no. 2, pp. 165–191, June 2001.

[9] J. M. Daida, J. A. Polito II, S. A. Stanhope, R. R. Bertram, J. C. Khoo, and S. A. Chaudhary, "What makes a problem GP-hard? analysis of a tunably difficult problem in genetic programming," in *Proc. Genetic Evolutionary Computation Conf.*, vol. 2, W. Banzhaf *et al.*, Eds., Orlando, FL, July 13–17, 1999, pp. 982–989.

[10] P. J. Darwen and X. Yao, "Does extra genetic diversity maintain escalation in a co-evolutionary arms race," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 4, no. 3, pp. 191–200, 2000.

[11] ——, "Why more choices cause less cooperation in iterated prisoner's dilemma," in *Proc. 2001 Congress on Evolutionary Computation*, 2001, pp. 987–994.

[12] E. D. de Jong, R. A. Watson, and J. B. Pollack, "Reducing bloat and promoting diversity using multi-objective methods," in *Proc. Genetic Evolutionary Computation Conf.*, L. Spector *et al.*, Eds., San Francisco, CA, July 7–11, 2001, pp. 11–18.

[13] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed., Washington, DC, 1989, pp. 42–50.

[14] K. A. DeJong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Dept. Comput. Commun. Sci., Univ. Michigan, Ann Arbor, 1975.

[15] P. D'haeseleer and J. Bluming, "Effects of locality in individual and population evolution," in *Advances in Genetic Programming*, K. E. Kinnear, Jr., Ed. Cambridge, MA: MIT Press, 1994, ch. 8, pp. 177–198.

[16] J. Eggermont and J. I. van Hemert, "Adaptive genetic programming applied to new and existing simple regression problems," in *Proc. 4th European Conf. Genetic Programming*, vol. 2038, LNCS, J. F. Miller *et al.*, Eds., Lake Como, Italy, Apr. 18–20, 2001, pp. 23–35.

[17] G. Eiben and J. van Hemert, "SAW-ing EAs: Adapting the fitness function for solving constrained problems," in *New Ideas in Optimization*, D. Corne *et al.*, Eds. New York: McGraw-Hill, 1999, pp. 389–402.

[18] A. Ekárt and S. Németh, "A metric for genetic programs and fitness sharing," in *Proc. European Conf. Genetic Programming*, vol. 1802, LNCS, R. Poli *et al.*, Eds., Edinburgh, U.K., 2000, pp. 259–270.

[19] ——, "Maintaining the diversity of genetic programs," in *Proc. 5th European Conf. Genetic Programming* , vol. 2278, LNCS, J. Foster *et al.*, Eds., Kinsale, Ireland, Apr. 3–5, 2002, pp. 162–171.

[20] L. J. Eshelman and J. D. Schaffer, "Crossover's niche," in *Proc. 5th Int. Conf. Genetic Algorithms*, S. Forrest, Ed., San Mateo, CA, 1993, pp. 9–14.

[21] C. Fernandes and A. Rosa, "A study on nonrandom mating and varying population size in genetic algorithms using a royal road function," in *Proc. 2001 Congress on Evolutionary Computation*, 27–30, 2001, pp. 60–66.

[22] C. Gathercole and P. Ross, "An adverse interaction between crossover and restricted tree depth in genetic programming," in *Proc. 1st Annual Conf. Genetic Programming 1996*, J. R. Koza *et al.*, Eds., CA, July 28–31, 1996, pp. 291–296.

[23] N. Geard and J. Wiles, "Diversity maintenance on neutral landscapes: An argument for recombination," in *Proc. IEEE Congress Evolutionary Computation*, 2002, pp. 211–213.

[24] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodalfunction optimization," in *Proc. 2nd Int. Conf. Genetic Algorithms and their Applications*, J. J. Grefenstette, Ed., Cambridge, MA, July 1987, pp. 41–49.

[25] C. Igel and K. Chellapilla, "Investigating the influence of depth and degree of genotypic change on fitness in genetic programming," in *Proc. Genetic Evolutionary Computation Conf.*, W. Banzhaf *et al.*, Eds., Orlando, FL, July 13–17, 1999, pp. 1061–1068.

[26] M. Keijzer, "Efficiently representing populations in genetic programming," in *Advances in Genetic Programming 2*, P. J. Angeline and K. E. Kinnear, Jr., Eds. Cambridge, MA: MIT Press, 1996, ch. 13, pp. 259–278.

[27] R. Keller and W. Banzhaf, "Explicit maintenance of genetic diversity on genospaces, Internal Rep.," Univ. Dortmund, Dortmund, Germany, 1995.

[28] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[29] W. B. Langdon, *Data Structures and Genetic Programming: Genetic Programming + Data Structures = Automatic Programming!*. Norwell, MA: Kluwer, 1998, vol. 1, Genetic Programming.

[30] W. B. Langdon and R. Poli, *Foundations of Genetic Programming*. New York: Springer-Verlag, 2002.

[31] Y. Liu, X. Yao, and T. Higuchi, "Evolutionary ensembles with negative correlation learning," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 380–380, Nov. 2000.

[32] S. Luke, "When short runs beat long runs," in *Proc. Genetic Evolutionary Computation Conf.*, L. Spector *et al.*, Eds., San Francisco, CA, 7–11, 2001, pp. 74–80.

[33] ——, (2002) ECJ: A Java-based evolutionary computation and genetic programming system. [Online]. Available: http://www.cs.umd.edu/projects/plus/ecj/

[34] S. Luke and L. Spector, "A revised comparison of crossover and mutation in genetic programming," in *Proc. 3rd Annual Genetic Programming Conf.*, J. Koza *et al.*, Eds., San Francisco, CA, 1998, pp. 208–213.

[35] W. N. Martin, J. Lienig, and J. P. Cohoon, "Island (migration) models: Evolutionary algorithms based on punctuated equilibria," in *Evolutionary Computation 2*, T. Back, D. B. Fogel, and Z. Michalewicz, Eds. Bristol, U.K.: Inst. Physics Publishing, 2000, ch. 15.

[36] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Trans. Model. Comput. Simulation*, vol. 8, no. 1, pp. 3–30, Jan. 1998.

[37] R. McKay and H. A. Abbass, "Anti-correlation: A diversity promoting mechanisms in ensemble learning," *The Australian J. Intell. Inform. Processing Syst.*, no. 3/4, pp. 139–149, 2001.

[38] R. I. McKay, "Fitness sharing in genetic programming," in *Proc. Genetic Evolutionary Computation Conf.*, D. Whitley *et al.*, Eds., Las Vegas, NV, July 10–12, 2000, pp. 435–442.

[39] R. I. McKay and H. A. Abbass, "Anticonelation measures in genetic programming," presented at the Australasia-Japan Workshop on Intelligent and Evolutionary Systems, Dunedin, New Zealand, 2001.

[40] N. F. McPhee and N. J. Hopper, "Analysis of genetic diversity through population history," in *Proc. Genetic Evolutionary Computation Conf.*, W. Banzhaf *et al.*, Eds., FL, 1999, pp. 1112–1120.

[41] S.-H. Nienhuys-Cheng, "Distance between herbrand interpretations: A measure for approximations to a target concept," in *Proc. 7th Int. Workshop Inductive Logic Programming*, N. Lavraĉ and S. Dzeroski, Eds., 1997, pp. 213–226.

[42] U.-M. O'Reilly, "Using a distance metric on genetic programs to understand genetic operators," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics, Computational Cybernetics and Simulation*, vol. 5, Orlando, FL, 1997, pp. 4092–4097.

[43] R. Poli and W. B. Langdon, "On the search properties of different crossover operators in genetic programming," in *Proc. 3rd Annu. Genetic Programming Conf.*, J. R. Koza *et al.*, Eds., Wisconsin, July 22–25, 1998, pp. 293–301.

[44] J. P. Rosca, "Entropy-driven adaptive representation," in *Proc. Workshop Genetic Programming: From Theory to Real-World Applications*, J. Rosca, Ed., Tahoe City, CA, July 9, 1995, pp. 23–32.

[45] ——, "Genetic programming exploratory power and the discovery of functions," in *Proc. 4th Conf. Evolutionary Programming*, J. R. McDonnell *et al.*, Eds., San Diego, CA, 1995, pp. 719–736.

[46] C. Ryan, "Pygmies and civil servants," in *Advances in Genetic Programming*, K. E. Kinnear, Jr., Ed. Cambridge, MA: MIT Press, 1994, ch. 11, pp. 243–263.

[47] S. Siegel, *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill, 1956.

[48] R. E. Smith, S. Forrest, and A. S. Perelson, "Searching for diverse, cooperative subpopulations with genetic algorithms," *Evol. Comput.*, vol. 1, no. 2, pp. 127–149, 1993.

[49] T. Soule and J. A. Foster, "Effects of code growth and parsimony pressure on populations in genetic programming," *Evol. Comput.*, vol. 6, no. 4, pp. 293–309, 1998.

[50] T. Soule and R. B. Heckendorn, "An analysis of the causes of code growth in genetic programming," *Genetic Program. and Evol. Mach.*, vol. 3, no. 3, pp. 283–309, Sept. 2002.

[51] W. A. Tackett, "Recombination, selection, and the genetic construction of computer programs," Ph.D. dissertation, Dept. Elec. Eng. Syst., Univ. Southern California, Los Angeles, 1994.

[52] R. K. Ursem, "Diversity-guided evolutionary algorithms," in *Proc. 7th Int. Conf. Parallel Problem Solving From Nature*, vol. 2439, LNCS, J. J. M. Guervós *et al.*, Eds., Granada, Spain, Sept. 2002, pp. 462–471.

Edmund K. Burke leads the Automated Scheduling, Optimization and Planning Research Group at the University of Nottingham, Nottingham, U.K. He is a Member of the Strategic Advisory Team for Information and Communications Technology of the U.K. Engineering and Physical Sciences Research Council (EPSRC). He is also a Member of the EPSRC Peer Review College. He is Editor-in-Chief of the *Journal of Scheduling*, Area Editor (for Combinatorial Optimization) of the *Journal of Heuristics*, and Associate Editor of the *INFORMS Journal on Computing*. He is Chairman of the Steering Committee of the international series of conferences on the Practice and Theory of Automated Timetabling (PATAT). He was Co-Chair of the Program Committee of the 1st International Conference on Multidisciplinary Scheduling: Theory and Applications (MISTA, 2003). He is also Co-Chair of the 8th International Conference on Parallel Problem Solving from Nature (PPSN, 2004). He Chaired the Special Program Committee on Evolutionary Scheduling and Routing at the Genetic and Evolutionary Computation Conferences (GECCO) in 2001, 2002, and will do so again in 2004. He has been a member of the program committees of over 40 international conferences in the last few years. He has edited/authored seven books (with a further four in preparation) and has published over 90 refereed papers.

Prof. Burke is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He is also a Member of the IEEE Neural Networks Society Technical Committee on Evolutionary Computation and he Chairs the IEEE Neural Networks Society Task Force on Evolutionary Scheduling and Timetabling. He has also been awarded 25 externally funded grants worth over 3M pounds from a variety of sources including EPSRC, ESRC, BBSRC, EU, and commercial organizations.

Steven Gustafson received the B.S. and M.S. degrees in computer science from Kansas State University, Manhattan, in 1999 and 2000, respectively. He is currently working toward the Ph.D. degree in computer science at the University of Nottingham, Nottingham, U.K.

He is a former Research Assistant with the Knowledge Discovery in Databases Laboratory, Kansas State University and is currently a Member of the Automated Scheduling, Optimization, and Planning Research Group, University of Nottingham. His research interests include evolutionary computation, machine learning, artificial intelligence, robotics, and software engineering. He has served on the program committees for the European Conference on Genetic Programming and the Genetic and Evolutionary Computation Conference, Genetic Programming track.

Mr. Gustafson has won The Best Poster Award for his previous paper at the European Conference on Genetic Programming in 2002, and had two papers receive nominations for the Best Paper at the Genetic and Evolutionary Computation Conference in 2002.

Graham Kendall is a Senior Lecturer in the School of Computer Science and Information Technology, University of Nottingham, Nottingham, U.K. Prior to his appointment in 1999, he spent over 15 years in the computer industry. He is a Member of the Automated Scheduling, Optimization, and Planning Research Group and is a Member of the U.K. EPSRC Peer Review College. He was Chairman of the Organizing Committee and Co-Chair of the Program Committee of the MISTA'03 Conference (1st Multidisciplinary International Conference on Scheduling: Theory and Applications, published by Kluwer). He was an editor of the selected papers volume of that conference. He was the co-editor of the INtroductory TutoRials in Optimization and Search Methodologies (INTROS'03) Workshop. He has co-edited or is associate editor of three other books. Since 2000, he has served on 23 international program/technical committees. He has published over 40 papers in international journals and conferences. His research areas include meta and hyperheuristics, evolutionary and adaptive computation, artificial life, stock cutting, scheduling and timetabling, and game playing.

Dr. Kendall has attracted external funding totaling over 1.6M pounds since 1999.