On Improving Genetic Programming for Symbolic Regression

Steven Gustafson, Edmund K. Burke and Natalio Krasnogor
School of Computer Science & IT, University of Nottingham
Jubilee Campus, Wollaton Rd., Nottingham, NG81BB, UK
{ smg, ekb, nxk }@cs.nott.ac.uk

Abstract- This paper reports an improvement to genetic programming (GP) search for the symbolic regression domain, based on an analysis of dissimilarity and mating. GP search is generally difficult to characterise for this domain, preventing well motivated algorithmic improvements. We first examine the ability of various solutions to contribute to the search process. Further analysis highlights the numerous solutions produced during search with no change to solution quality. A simple algorithmic enhancement is made that reduces these events and produces a statistically significant improvement in solution quality. We conclude by verifying the generalisability of these results on several other regression instances.

1 Introduction

Our recent research has focused on improving the understanding of the concept of diversity in genetic programming (GP) and evolutionary algorithms (EAs)[1, 2, 3, 4, 5, 6, 7]. As population diversity plays a key role in the search process, we believe that to find principled ways of improving search, a measure and problem specific understanding of diversity is necessary. This nuanced view, recently supported in [8], is a deviation from previous work that supposes that the general increase of diversity is better for problem solving, and, conversely, that the loss of diversity is the cause of poor problem solving. In this paper, we continue researching the role of diversity by studying the recombination success of dissimilar solutions. This work builds directly upon the existing understanding of the canonical GP search process for the Ant, Parity and regression domains, recently summarised and extended in [5]. In that work, algorithmic improvements were explained by understanding population diversity issues for the Ant and Parity domain. In this paper, we carry out an in-depth analysis to find an empirically motivated way to improve search for the symbolic regression domain.

2 Background

In [5], Lineage selection was used to increase population edit distance diversity and the number of unique fitness values in a standard GP framework for the Ant and Parity domains. The results of that study clearly aligned with previous work on diversity methods and comparisons to hill-climbing. The Ant problem benefited from increased diversity, in terms of genotypes and phenotypes, using Lineage selection. These results agreed with previous work showing GP and similar methods outperforming hill-climbing strategies for this problem, e.g. [9]. Intuitively, this is due to the ability of diversity to escape the numerous local optima that achieve a similar fitness but in very different ways. The Parity problem search space was shown to be more amenable to hill-climbing-like strategies for GP than to standard GP search [10]. That is, increasing different aspects of population diversity using Lineage selection led to worse overall solution quality. The regression domain (using the Binomial-3 instance) proved deceptive in [5], with few concrete conclusions save that GP behaves differently than on the Ant and Parity domains. GP using Lineage selection on the Binomial-3 instance with increased diversity performed worse, but also without the benefit of previous work to justify these results. In fact, this domain also showed itself to be unique from the Ant and Parity in its lack of correlation between solution quality and various measures of diversity [2]. We believe that a more in-depth analysis of dissimilarity and recombination is required for a well-motivated improvement to GP search on this domain.

2.1 Dissimilarity and Mating

In EAs, the recombination operator (typically two-parent crossover) produces new solutions (offspring) during search. The success of this operator is ultimately defined by how well the overall search proceeds. However, we may wish to consider a successful recombination event as one that improves solution quality between parents and offspring solutions. The loss of dissimilarity, defined later using edit distance, in solutions typically signifies the end of the search process as it becomes unlikely that new and improving solutions are found. For this reason, many methods haven been proposed to maintain or increase diversity. In this paper we are concerned with the specific case of mating between dissimilar solutions as a means to understand and ultimately improve GP search. Most importantly, mating events provides new solutions during search, allowing progress to be made in solution quality. Additionally, the recombination success of dissimilar solutions is related to four significant lines of GP applications and research: multipopulation and distributed models, mate selection, diversity methods and operator design and application. Thus, analysing mating in terms of diversity and dissimilarity may not only assist in developing a nuanced understanding of GP search, but also facilitate improvements to other GP methods and models.

Multipopulation models allow easy distributed computing, often explicitly promoting diversity and recombination between dissimilar solutions. Grid topologies [11] encourage local mating that may promote more similar mates, but can also lead to more genetically distinct subpopulations that will be recombined later. Island models typically use migration events to distribute fit solutions between isolated

populations, where migrants are likely to be genetically different from their new subpopulation. Migrating fit solutions introduces another level of selection pressure.

Various mating methods have been shown to improve search, often producing smaller solutions with similar fitness. Disassortative mating in [12] selects differently fit and sized solutions as parents. Negative assortative mating in [13] selects mates based on maximal Hamming distances. In [14], a two-level mate selection tournament based on fitness and then diversity (using an edit distance) was used, again selecting for greater edit distance. In [5], Lineage selection encouraged mating between different genetic lineages that are likely to contain dissimilar solutions.

Diversity methods based on differences in the population have been previously used to control population diversity in GP. The sharing method requires solutions to receive a lower fitness if many solutions already exist with a similar genotype or phenotype [15, 16, 17, 18]. An adaptive method encourages the reduction of genetic diversity while solution quality continues to improve [19]. As the levels of similarity in the population can effect the recombination operator responsible for producing new search points, controlling population diversity is likely to have dramatic effect on search.

In contrast to the above methods, several recombination operators focus on the similarity between parents and attempt to preserve the context or similarity of exchanged genetic material [20, 9, 21, 22]. The intuition used in these studies is that preserving similarity between mates will increase the chance of producing offspring with a similar fitness or behaviour in the parents.

Dissimilarity concepts are used within GP in many contexts. Understanding how search works on a problem domain is critical for making algorithmic or representation improvements. The aim of the following study is to gain such an understanding to improve the search algorithm for the regression domain.

2.2 Definitions

We first need to define several concepts. Some of these make use of an edit distance measure between strings, or Levenshtein edit distance, to define the distance between trees (solution structure and content), as used in previous research [23, 19, 2]. The selection of parent solutions for recombination is based on fitness, and thus we focus on the better fit solutions. In GP, recombination is the most common operator used and is typically implemented as subtree crossover. This operator replaces a subtree in a solution (root-parent) with a subtree selected from another solution (non-root-parent or donor parent).

Fit and Unfit. Given a population p with m solutions, we define the *fit* population as those solutions with a fitness better or equal-to than more than half the other solutions in the population. The $unfit$ population is defined as $(p - fit)$. The criterion for determining this set can be adjusted for selection methods, here it is having fitness that is better or equal to more than half the population.

Dissimilar. Given the mean (μ) and standard deviation (σ) of a population's average pair-wise edit distance, and each solution i's average pair-wise distance v_i, we define the dissimilar population as:

$$dissimilar = \{\forall i \in P |\ v_i > (\mu + 2 \times \sigma)\}.$$

The dissimilar population are the solutions that have an average pair-wise distance greater than two standard deviations from the population mean pair-wise distance. We define the fit and dissimilar population as $(fit \bigcap dissimilar)$ and the fit and similar population as $(fit \bigcap (p - dissimilar))$.

Survival. Given a population p^t at time t, we let the predicate $Parent(i, j)$ be true if i is in p^t and is the root-parent of j, in p^{t+1}. Given a solution $i \in p^t$, the set $produced_i$ consists of those solutions that i produced (i.e. was a root-parent for) in p^{t+1}. Finally, given i and $produced_i$, the surviving offspring of i are defined as:

$$survived_i = \{\forall j \in produced_i |\ k \in p^{t+2} \wedge Parent(j, k)\}.$$

For a set of solutions D, that are a subset of population p^t, the rate of survival is defined as:

$$survival\_rate(D) = \frac{\sum_{i \in D} \#survived_i}{\sum_{i \in D} \#produced_i}$$

The survival rate of a set of solutions is the ratio of the number of their offspring that were later selected for recombination over the number of their offspring.

In the study that follows, we will also make use of some diversity measures, described next.

Edit Distance Diversity. Given two solutions represented by syntax trees (binary in this study), we overlap the trees at the root node and downwards. The edit distance between two solutions is the minimum number of node additions, deletions, and transformations needed to make the two trees equal in structure and content [24, 2]. The distance is normalised by the size of the trees. To represent the edit distance diversity of a *population of solutions*, we use the average edit distance between the current best solution and all other population members. This is less complex than a all-pairs distance measure and we have found it to be useful in previous work [2].

Fitness-Based Entropy. Given a solution i in the population with a fitness value f_i, we can represent the partition of the population also with fitness value f_i as k. The proportion of the population occupying a partition k is denoted as p_k, and the entropy is defined as

$$-\sum_k p_k \times \log p_k.$$

Entropy represents the amount of chaos in the population, according to the number of unique fitness values

and the distribution of the population over those values. Low entropy describes a population with few unique fitness values and many solutions with the same fitness. High entropy describes a population with many unique fitness values and a more evenly distribution over those fitness values by the population.

3 Experimental Procedure

We use a canonical GP system with the binary syntax tree representation. Solutions (trees) consist of functions (internal nodes with two arguments) and terminals. Trees are initialised using the ramped half-n-half method, with tree depth between 2 and 4. The subtree crossover operator is used with internal node selection probability of 90% and bounded to trees less than depth 10. A population of size 500 is used, where maximum number of generations of 51 is defined for the generational algorithm using a tournament size of 4 for tournament selection. We begin our analysis of the symbolic regression domain using the Binomial-3 function and an instance from previous work [25, 4, 5]. These instances and this methodology are very common in the literature.

The symbolic regression problem uses the Binomial-3 function ($(1+x)^3$) with 50 equidistant points in the range [0,1]. The fitness is the mean squared error over all the test points (minimisation). The function set consists of arithmetic operators +, *, -, p/, where protected division returns 0 if denominator is close to 0. The terminal set consists of the x value for each of the 50 equidistant point set and ephemeral random constants, initialised in the range [-10,10] (which was found to produce instances with medium difficulty for GP in [25]).

4 Results

Note that in Fig. 1, the survival rate is defined in Section 2.2 for a group of solutions. This value represents the ability of solutions to produce offspring that then produce solutions themselves in the future. A survival rate of 1.0 denotes the case where, on average, every offspring produced by a population goes on to produce one offspring itself.

Fig. 1 shows a population composed of approximately half *similar and fit* and half *unfit* solutions. On average, three percent of the population are *dissimilar and fit*. These averages are fairly consistent throughout the runs and are reasonable values to expect. Fig. 1 also shows the survival rates of the different subpopulations. In an evolutionary search process, it is reasonable to expect that fit solutions would have a higher survivability than unfit ones, which is the result seen for the Ant and Parity problems in [1]. That is, we would hope that the fitness function, representation and operators create a search space where *fit* solutions produce new and fit solutions that contribute heavily toward the search (i.e. survive). In this study, it is surprising to see that the search process is not favouring the fit solutions (either similar or dissimilar). Also, after approximately 10 generations, as seen in Fig. 1, the fit and dissimilar have a much

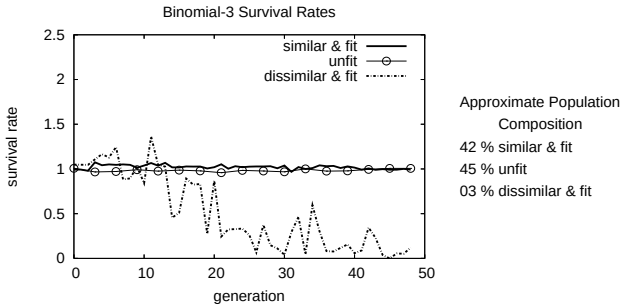

Figure 1: Binomial-3 Problem average survival rates per generation and the approximate breakdown of the population into three sub populations, average over 30 random runs.

lower survival rate that even the unfit.

Based on Fig. 1, the GP search process relies on fit solutions only a fraction more than the unfit solutions to produce new search points. Instead of search being mostly influenced by fit solutions, the search process appears to be equally influenced by fit and unfit solutions. Also, whereas intuition might suggest that population search relies on dissimilar solutions to escape local optima, these results show the search process being influenced by the dissimilar and fit solutions only in the first few generations. However, solution quality is being improved right up until the last generations, where the mean generation where the runs stop improving is 41, with a standard deviation of 10 generations.

These results resonate with the previous inability to characterise the GP search process for regression domain. To better understand GP search on the regression domain in the light of on these results, we next look at the mating success and dissimilarity of all parents. As fit and similar solutions had a similar survival rate as unfit solutions, and also as the dissimilar and fit solutions had a much lower survival rate, we are particularly interested in understanding the affect of solution similarity on the production of new solutions. In the next section, we focus on the search process before generation 30 when the survival rate of the dissimilar and fit is transitioning the most in Fig. 1.

4.1 Mating Success and Dissimilarity

We calculate the change of fitness from the *average fitness of the parents* to the offspring and note the dissimilarity between the parents. Fig. 2 shows the probability density functions (PDF) for parent dissimilarity that resulted in improving fitness, worsening fitness and no change in fitness in offspring before generation 30. The PDFs in Fig. 2 show that the parents that are more dissimilar are more likely to produce a change in solution quality (versus making no change). The probability of making no change in solution quality increases significantly when solutions are very similar, as does the probability of worsening fitness. Overall, the search process improves upon the parents' fitness 21.71% of the time and worsens parents' fitness 65.96% of the time. However, 12.26% of all mating events results in no change in fitness. Even when parents are highly dissimilar, there is

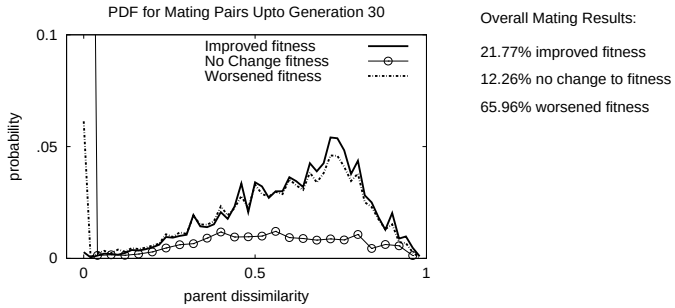

Figure 2: Binomial-3 problem mating dissimilarity versus probability of improving, worsening, or making no change to fitness in offspring for generations up-to generation 30.

a relatively good chance of not producing a change in fitness. The fitness space is continuous, and in all but one or two of these cases, offspring with the same fitness as their parents' average fitness came from *parents with the same fitness*. The average number of 'no change to fitness' events occurred 2,500 times a run, with a minimum of 366, a maximum of 7,100 and a standard deviation of 1,700. Over all mating events, approximately 18% are between parents with the same fitness. However, of those 18%, the resulting offspring has the same fitness as the parents 56% of the time. This percentage decreases by around 3% when we do not consider genetically-identical parents. In the other 82% of mating events, only 28% result in offspring with the same fitness as its root-parent. By focusing on the dissimilarity between parents and their mating success, an unexpected property of search has been highlighted: dissimilar and similar parents with equal fitness produce, on average, new solutions without a change in their fitness over 50% of the time.

4.2 On Improving GP

Previously, a variety of operators and hill-climbing methods were tested on the regression instance $x^4 - 3x^3 + 9x^2 - 27x$ in [26]. Stochastic hill-climbing performed significantly worse than a steady-state GP system using mostly subtree crossover with two forms of mutation. In this case, hill-climbing accepted a randomly chosen, non-worsening solution. These results, taken together with this study, suggest that GPs less conservative acceptance strategy (as compared to hill-climbing) produces better search, possibly due to a general difficulty in producing improving solutions. In [2], the only diversity measure that showed a correlation with fitness in the regression domain was the number of unique fitness values in the population (more values correlated positively with better fitness).

Increasing the number of unique fitness values in the population may be a way to improve search. This is based on the idea that reducing the number of 'no change to fitness' events would increase the number of differently fit solutions visited during search. Producing more differently fit solutions might increase the chance of finding improving solutions simply because there are more solutions. Visiting more solutions seems to be a desirable improvement that would produce better search. Note that this idea is different from other methods that suppose the operator will work better with differently fit solutions, or that mating differently fit solutions will increase diversity – both believed to improve search.

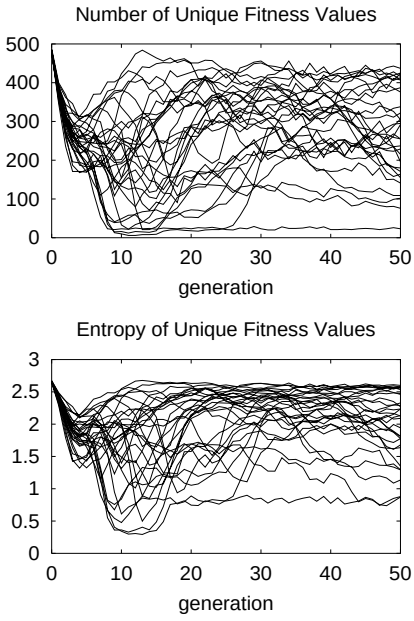

Figure 3: The number of unique fitness values in each population is shown in the top graph, and the fitness-based entropy of each population is shown in the bottom graph.

Figure 3 shows the number of unique fitness values and the fitness-based entropy of each population for the GP runs. In most runs, there is a large number of unique fitness values. Also, the population distribution over the unique fitness values tends to remain relatively high throughout the run, according the entropy measure. Reducing the number of 'no change to fitness' events is likely to increase the number of unique fitness values in the population and increase the population entropy. According to our previous work in [2], where increases in these measures were positively correlated with better fitness, we would then expect better solution quality by increasing the number of unique fitness values and fitness-based entropy. Additionally, at least intuitively, reducing the number of 'no change to fitness' might reduce the time required to achieve good solution quality as the search process spends less time on fitness plateaus (i.e. areas of equally fit solutions with little information to guide search). Of course, if those fitness plateaus are *beneficial*, the search process may be impaired.

5 A Principled Improvement

Fig. 2 showed a surprising number of 'no change to fitness' events from dissimilar parents with equal fitness. We suggest that decreasing these events is likely to lead to more new solutions visited and one could hypothesise that this would improve search. While our study was motivated by

understanding the role of solution dissimilarity and population diversity during the search process, it is the many 'no change to fitness' events produced by equally-fit parents that suggests the most direct and simple way of improving GP search for symbolic regression: *We prevent mating between two solutions with the same fitness.* As 'no change to fitness' events occurred when parents, similar and dissimilar, had the same fitness, we carry-out the same experiment as above, only we do not allow two parents with the same fitness to mate. Note that this method is similar in principle to other selection techniques that consider fitness values for mate selection. However, it is uniquely motivated by the reduction of time spent on producing solutions with no change in quality from parents. Also, it is fundamentally easier to implement and less likely to make an overwhelming change to other algorithm dynamics. To prevent mating between equally fit parents, we simply select the first parent followed by the repeated selection of another parent until we find one with different fitness.

Table 1 reports that the new method significantly improves solution quality (*Fitness*). Significance was tested using a Student's T-test with p=0.05. The new method also significantly increases the average number of the 50 test points (*Hits*) correctly classified by a solution. The effects of the new mating method are seen by the significant increase of unique fitness values (*Phenotypes*) (and *Entropy*) in the population due to the decrease of the 'no change to fitness' events. That is, it shows that mating solutions with different fitness does, as intended, produce more offspring with different fitness values. Note that the new runs themselves are not significantly different according to changes in average solution *Size*, average *Depth* of solution, population *Diversity* according to edit distance and the number of genetically unique solutions (*Genotypes*) in the populations.

By reducing the number of 'no change to fitness' events, more improved fitness events (and worsened fitness events) were created, as reported in Table 2. Fig. 4 reports the performance of the GP search process as a function of the cost (node evaluations) required to achieve a solution quality. Confidence bars are plotted at the 95% level. It takes about

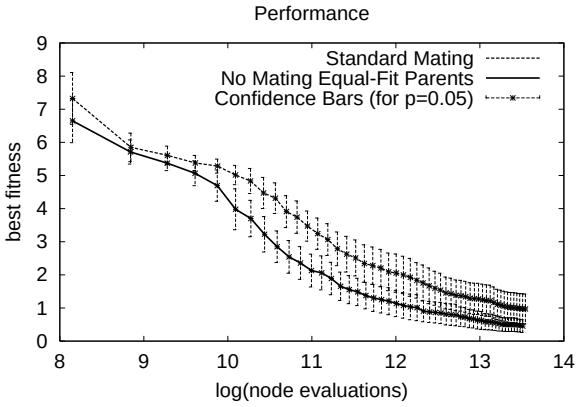

Figure 4: The best fitness of the run (at the end of each generation) is plotted against the node evaluations taken by the run so far.

Table 1: Various measures taken from last generation of runs with (Old) and without (New) mating between equal-fit parents. A ***** next to the *p-value* indicates a statistically significant difference between Mean values.

Measure (p-value)	Exp.	Min.	Max.	Mean	Stdev.
Fitness	Old	0.068	4.669	0.970	1.211
**2.144*	New	0.008	2.194	0.454	0.521
Hits	Old	2.000	50.0	28.333	16.449
**2.463*	New	5.000	50.0	38.133	14.299
Size	Old	5.164	112.58	49.729	20.457
0.644	New	24.98	71.120	46.811	14.056
Depth	Old	3.050	9.904	8.809	1.471
0.563	New	8.002	9.800	8.969	0.514
Diversity	Old	0.207	0.657	0.369	0.085
1.237	New	0.265	0.554	0.392	0.057
Genotypes	Old	25.0	473.0	353.73	115.18
1.453	New	263.0	468.0	387.4	53.168
Phenotypes	Old	23.0	437.0	268.6	117.51
**2.737*	New	213.0	464.0	337.43	71.844
Entropy	Old	0.872	2.60	1.996	0.524
**3.503*	New	1.882	2.65	2.356	0.206

Table 2: Overall mating results for standard mating (Old) and using the new mating method (New) that prevents parents from having the same fitness. Note the reduction of the 'no change to fitness' events and the increased percentage of 'improved fitness' events.

Up-to Gen. 30	Old	New
improved fitness	21.77%	**25.83%**
no change to fitness	12.26%	**01.50%**
worsened fitness	65.96%	72.66%

$x = 1.0 \times 10^{10}$ node evaluations (or $log(x) = 10$), before a fitness value less than 4.0 is found using the new mating method. In contrast, the standard mating method requires about $x = 4.0 \times 10^{10}$ node evaluations (or $log(x) = 10.6$) to achieve the same solution quality. The new method appears to allow GP to improve solution quality and reach similar solution quality with fewer node evaluations (or with a smaller computational complexity).

With respect to the populations produced using the new mating method, Fig. 5 shows the correlation between the fitness-based entropy of a population and its current best solution quality (restricted to fitness less than a value of 1.0). We can see some evidence for a positive correlation, where better solution quality is achieved when the entropy is higher. This correlation supports the hypothesise that increasing the number of differently fit solutions visited during search will have a positive impact on the search process. Based on the above study, we were able to achieve a principled GP algorithm improvement due to an improved nuanced understanding of GP in the regression domain.

6 Other Regression Instances

To understand how preventing mating between parents with the same fitness effects search on other domains, we tested the method on the Quartic polynomial ($x^4 + x^3 + x^2 + x$), the Sextic polynomial ($x^6 - 2x^4 + x^2$) with ERC ranges of $[-1, 1]$, two more instances of the Binomial-3 problem (from [25], with ERC ranges of $[-1, 1]$ and $[-100, 100]$)) and three instances of random polynomials of degree 3, 7 and 11 (described fully in [4]). Both the Binomial-3 instances and the random polynomials were shown to be tunably difficult for GP. The Quartic polynomial can be classified as a relatively easy instance for GP, and the Sextic polynomial, in comparison to the Quartic polynomial, as a harder instance for GP. We use the same algorithmic setup as used above. Each instance was tested using the new mating method and the standard method for 30 runs, and Table 3 reports the improvements resulting from preventing parents from having equal fitness.

On all three Binomial-3 instances, a statistically significant improvement to fitness was found using the new mating method. On the most difficult random polynomial instance (degree 11), and the harder Sextic polynomial, fitness was also statistically improved. On the easier random polynomials (degree 3 and 7) and the Quartic instance, no statistical change occurred in fitness. The new mating method was innocuous on these easier instances. However, there was some decrease in the time required to achieve the best fitness using the proposed method in these latter instances. For example, Figure 6 shows that better solution quality is achieved with fewer node evaluations in the earlier stages of the search process for the Quartic polynomial, an easy instance for GP. However, according to the confidence bars shown in Figure 6, the improvement was not enough in our runs to be significant.

Preventing mating between parents with equal fitness appears to be a sound improvement to GP search as it improves solution quality on harder instances and is harmless on easier ones, highlighted in Table 4. Also, while preventing mating of equal-fit parents seems to have little affect on other aspects of the search dynamics, the new method appears to require fewer node evaluations to achieve good fitness. It is likely that this method is suitable for other problems with continuous fitness spaces or very large discrete ones.

Table 3: Average Best fitness from last generation of runs with (Old) and without (New) mating between equal-fit parents. A ***** next to the *p-value* indicates a statistically significant difference at the 95% level between Mean values. The Sextic improvement was at the 90% level.

Instance (p-value)	Exp.	Min.	Max.	Mean	Stdev.
Binomial-3 $ERC_{[-1,1]}$					
	New	0.031	0.721	0.211	0.151
**2.49311*	Old	0.075	3.114	0.497	0.609
Binomial-3 $ERC_{[-10,10]}$					
	New	0.008	2.194	0.454	0.521
**2.14416*	Old	0.068	4.669	0.970	1.211
Binomial-3 $ERC_{[-100,100]}$					
	New	0.095	3.512	1.706	0.978
**2.18641*	Old	0.352	4.164	2.241	0.915
Random Polynomial Degree 3					
	New	0.027	2.271	0.728	0.658
0.284501	Old	0.027	2.315	0.681	0.623
Random Polynomial Degree 7					
	New	0.175	1.136	0.393	0.215
0.933727	Old	0.208	0.655	0.352	0.099
Random Polynomial Degree 11					
	New	0.036	0.110	0.070	0.024
**3.12846*	Old	0.038	0.153	0.093	0.033
Quartic Polynomial					
	New	0.000	0.000	0.000	0.000
n/a	Old	0.000	0.000	0.000	0.000
Sextic Polynomial					
	New	0.254	2.455	2.002	0.461
**1.80146*	Old	1.476	2.592	2.180	0.284

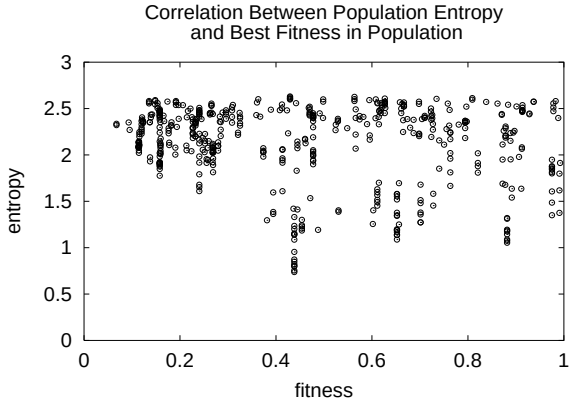

Figure 5: The correlation between the fitness-based entropy of a population and its best fit solution. Each point represents a population produced using the new mating method.

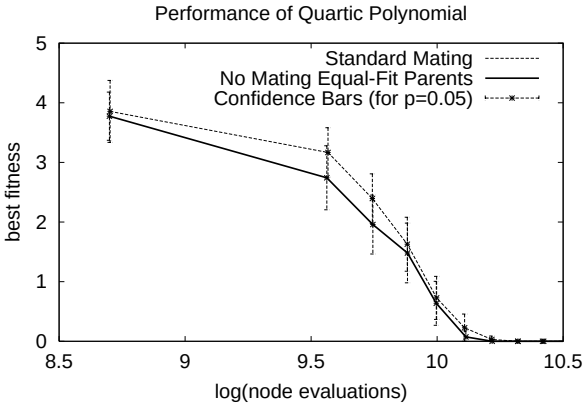

Figure 6: Performance on the Quartic Polynomial as a function of computational cost (node evaluations) using standard mating and the new method that prevents mating between equal-fit parents.

Table 4: Instances where fitness was improved, due to the new mating method, are shaded - otherwise, no change occurred.

	GP Hardness		
Instances	Easy		Hard
Binomial-3 (ERCs)	[-1,1]	[-10,10]	[-100,100]
Random Polynomial	degree 3	degree 7	degree 11
Polynomial	Quartic		Sextic

7 Conclusions

An improvement to the GP search process was reported in this paper for the symbolic regression domain. GP search has been generally difficult to characterise for symbolic regression, e.g. seen by the lack of a clear correlation between diversity and fitness in [2, 6, 7] or an understandable response to increasing diversity in [5]. While GP search for some other domains can be characterised and algorithmic improvements justified, a lack of such evidence in the regression domain prevents well motivated algorithmic improvements. Our study proceeded as follows:

- An analysis of survival rates of the population highlighted the unexpected lack of influence of fit solutions that were similar and dissimilar.
- A study of dissimilarity and mating showed the frequency of which new solutions have the same fitness as their parents, even when parents are dissimilar.
- Taken together with previous work, the results supported the idea that GP search may be improved by producing more differently fit solutions during search, a hypothesis suggested by the positive correlation between solution quality and fitness-based entropy in [2, 6, 7].
- Finally, a new, simple and well motivated method to prevent the mating between two parents with equal fitness values was tested. *This new method increased the number of new solutions with different fitness values which led to a statistically significant improvement in solution quality.*

Overall, we tested this method on three Binomial-3 instances (that are increasingly difficult for GP), three random polynomials with increasing degree (also increasingly difficult for GP) and the Quartic and the Sextic polynomial, which are frequently used in the literature. Solution quality was improved for all the Binomial-3 instances, for the most difficult random polynomial instance and the GP-harder Sextic polynomial. Significantly, for the other instances that are easier for GP to solve, the method was innocuous. When the instance is easy enough for GP, the new mating method is not required and was harmless to performance. Similar to the use of interval arithmetic and scaling [27], this paper makes a significant contribution toward improving the GP algorithm by means of mating and solutions produced, specifically for the symbolic regression domain, but with possibilities for generalisation to other domains.

Acknowledgments

This work was supported by EPSRC Grant GR/S70197/01.

Bibliography

[1] S. Gustafson. *An Analysis of Diversity in Genetic Programming*. PhD thesis, School of Computer Science and Information Technology, University of Nottingham, Nottingham, England, February 2004.

[2] E.K. Burke, S. Gustafson, and G. Kendall. Diversity in genetic programming: An analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation*, 8(1):47–62, 2004.

[3] S. Gustafson, E.K. Burke, and G. Kendall. Sampling of unique structures and behaviours in genetic programming. In M. Keijzer et al., editors, *Genetic Programming, Proceedings of the 6th European Conference*, volume 3003 of *LNCS*, pages 279–288, Coimbra, Portugal, April 2004. Springer-Verlag.

[4] S. Gustafson, A. Ekárt, E.K. Burke, and G. Kendall. Problem difficulty and code growth in genetic programming. *Genetic Programming and Evolvable Hardware*, 5(3):271–290, 2004.

[5] E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor. Is increasing diversity in genetic programming beneficial? An analysis of the effects on fitness. In B. McKay et al., editors, *Congress on Evolutionary Computation*, pages 1398–1405, Canberra, Australia, December 2003. IEEE Press.

[6] E. Burke, S. Gustafson, and G. Kendall. A survey and analysis of diversity measures in genetic programming. In W. B. Langdon et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 716–723, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[7] E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor. Advanced population diversity measures in genetic programming. In J.J. Merelo Guervós et al., editors, *Parallel Problem Solving from Nature*, volume 2439 of *LNCS*, pages 341–350, Granada, Spain, September 2002. Springer.

[8] J.M. Daida, D.J. Ward, A.M. Hilss, S.L. Long, M.R. Hodges, and J.T. Kriesel. Visualizing the loss of diversity in genetic programming. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 1225–1232, Portland, Oregon, 20-23 June 2004. IEEE Press.

[9] W.B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer-Verlag, Berlin, 2002.

[10] A. Juels and M. Wattenberg. Stochastic hillclimbing as a baseline method for evaluating genetic algorithms. Technical Report Technical Report CSD-94-834. Computers Science Department, University of California at Berkeley, USA, 1995.

[11] R.J. Collins. *Studies in Artificial Evolution*. Ph.D. dissertation, Department of Computer Science, University of California at Los Angeles, 1992.

[12] C. Ryan. Pygmies and civil servants. In K.E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 11, pages 243–263. MIT Press, Cambridge, MA, 1994.

[13] C. Fernandes and A. Rosa. A study on non-random mating and varying population size in genetic algorithms using a royal road function. In *Proceedings of the Congress on Evolutionary Computation*, pages 60–66. IEEE Press, 27-30 2001.

[14] M. Brameier and W. Banzhaf. Explicit control of diversity and effective variation distance in linear genetic programming. In A.G.B. Tettamanzi et al., editors, *Genetic Programming, Proceedings of the 5th European Conference*, volume 2278 of *LNCS*, pages 162–171, Kinsale, Ireland, April 2002. Springer-Verlag.

[15] 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*, pages 42–50, San Mateo, CA, USA, 1989. Morgan Kaufmann.

[16] J. Hu, K. Seo, S. Li, Z. Fan, R.C. Rosenberg, and E.D. Goodman. Structure fitness sharing (SFS) for evolutionary design by genetic programming. In W.B. Langdon et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 780–787, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[17] J. Hu, E.D. Goodman, and K. Seo. Continuous hierarchical fair competition model for sustainable innovation in genetic programming. In R.L. Riolo and B. Worzel, editors, *Genetic Programming Theory and Practice*, chapter 6, pages 81–98. Kluwer, 2003.

[18] 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, NV, USA, 10-12 July 2000. Morgan Kaufmann.

[19] A. Ekárt and S. Németh. Maintaining the diversity of genetic programs. In J. Foster et al., editors, *Genetic Programming, Proceedings of the 5th European Conference*, volume 2278 of *LNCS*, pages 162–171, Kinsale, Ireland, 3-5 April 2002. Springer-Verlag.

[20] 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.

[21] R. Poli and W.B. Langdon. On the search properties of different crossover operators in genetic programming. In J.R. Koza et al., editors, *Proceedings of the Third Annual Genetic Programming Conference*, pages 293–301, Madison, WI, USA, 22-25 July 1998. Morgan Kaufmann.

[22] M.D. Platel, M. Clergue, and P. Collard. Maximum homologous crossover for linear genetic programming. In C. Ryan et al., editors, *Genetic Programming, Proceedings of the 6th European Conference*, volume 2610 of *LNCS*, pages 200–210, Essex, UK, 14-16 April 2003. Springer-Verlag.

[23] 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*, pages 11–18, San Francisco, CA, 7-11 July 2001. Morgan Kaufmann.

[24] A. Ekárt and S. Németh. A metric for genetic programs and fitness sharing. In R. Poli et al., editors, *Genetic Programming, Proceedings of the 3rd European Conference*, volume 1802 of *LNCS*, pages 259–270, Edinburgh, 2000. Springer-Verlag.

[25] 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 Programming and Evolvable Machines*, 2(2):165–191, June 2001.

[26] K. Harries and P. Smith. Exploring alternative operators and search strategies in genetic programming. In J.R. Koza et al., editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 147–155, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[27] M. Keijzer. Improving symbolic regression with interval arithmetic and linear scaling. In C. Ryan et al., editors, *Genetic Programming, Proceedings of the 6th European Conference*, volume 2610 of *LNCS*, pages 71–83, Essex, UK, 14-16 April 2003. Springer-Verlag.