The Roles of Diversity Preservation and Mutation in Preventing Population Collapse in Multiobjective Genetic Programming

Khaled M S Badran
Laboratory for Image and Vision Engineering
Department of Electronic and Electrical Engineering
University of Sheffield, Sheffield S1 3JD, UK
khaled.badran@shef.ac.uk

Peter I Rockett
Laboratory for Image and Vision Engineering
Department of Electronic and Electrical Engineering
University of Sheffield, Sheffield S1 3JD, UK
p.rockett@shef.ac.uk

ABSTRACT

It has been observed previously that genetic programming populations can collapse to all single node trees when a parsimony measure (tree node count) is used in a multiobjective setting. We have investigated the circumstances under which this can occur for both the 6-parity boolean learning task and a range of benchmark machine learning problems. We conclude that mutation is an important – and we believe a hitherto unrecognized – factor in preventing population collapse in multiobjective genetic programming; without mutation we routinely observe population collapse. From systematic variation of the mutation operator, we conclude that a necessary condition to avoid collapse is that mutation produces, on average, an increase in tree sizes (bloating) at each generation which is then counterbalanced by the parsimony pressure applied during selection. Finally, we conclude that the use of a genotype diversity preserving mechanism is ineffective at preventing population collapse.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.6 [**Artificial Intelligence**]: Learning—*induction*

General Terms

Algorithms

Keywords

Genetic programming, multiobjective optimization, bloat control, population collapse, diversity preservation.

1. INTRODUCTION

It is well-established that unless active measures are taken to prevent it, the trees in a genetic programming population will *bloat*; that is, there is a tendency for tree sizes in the population to grow without limit. The objections to bloat are well-rehearsed [3, 9] and considerable effort has been expended on both analyzing its causes and suppressing unnecessary tree growth.

Langdon and Poli [9] have summarized the three principal approaches to preventing bloat: i) Limiting tree depth to some maximum value, ii) Tailoring genetic operators and iii) Using parsimony pressure, including the use of multiobjective (MO) methods. Capping the tree depth is unsatisfactory since (paradoxically) it requires knowledge of the maximum necessary depth in advance of solving the problem. Tailoring the genetic operators has proved problematic.

A number of authors have used genetic programming (GP) – for example, [2, 12, 13, 14] – across a range of tasks with parsimony pressure incorporated in a multiobjective framework, where one of the objectives to be minimized is a measure of tree size. This approach has been found to be extremely effective at controlling bloat.

Our motivation for undertaking the present work has been the frequent comments by anonymous reviewers of our submitted papers that MO methods are "inappropriate" for controlling bloat and in one case, we have been told – quite contrary to all our experience – that MO methods "don't work". These reviewers have typically cited the work of de Jong and Pollack [3] who concluded that, without an explicit diversity-preserving mechanism, the population in MO-GP rapidly degenerates to just trees of a single node. (Hereafter, we refer to this phenomenon as *population collapse* or *collapse*, for short.) In all the MO-GP work we have ever done over two or three years, which must total many hundreds of GP runs under a wide range of (often, highly) experimental conditions, we have never once observed population collapse. We were thus interested to understand exactly what it is we (and others) are doing that prevents the population collapse seen by de Jong and Pollack [3].

The structure of this paper is that in Section 2 we carefully re-examine the paper of de Jong and Pollack [3] and based on this, we have conducted a set of systematic experiments which we report in Section 4; our methodology is set-out in Section 3. We conclude that mutation has a key – and we

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

believe, hitherto unrecognized – rôle in bloat control by MO methods which we discuss and expand upon in Section 5. Finally, we offer concluding remarks in Section 6.

2. DE JONG AND POLLACK REVISITED

De Jong and Pollack [3] concluded that an explicit diversity preserving mechanism is necessary to prevent population collapse in MO-GP and without such a modification, all the trees in the population degenerate to a single node within around ten generations. The argument developed by de Jong and Pollack was based principally on examining the behaviour of a "standard" multiobjective GP procedure on the even 6-parity benchmark problem; without diversity preservation, population collapse ensued. With a phenotypic diversity preserving operation, on the other hand, good sampling of the Pareto front was obtained. De Jong and Pollack then employed a second GP setup incorporating phenotype diversity preservation and mutation on a range of problems (a harder version of the 6-parity problem, the 11-multiplexer and symbolic regression problems). In this second series of experiments, no collapse was seen which they interpreted as evidence of the effectiveness of diversity preservation. A third series of experiments employed mutation together with both phenotype diversity and a genotype diversity measure as a third objective; again no collapse was seen.

Careful analysis of [3], however, reveals a number of points which are significant for the work reported here. Firstly, the "standard" MO-GP procedure used in [3] to demonstrate collapse is not, in our judgement, typical of (MO-) GP methodologies described in the literature. Most significantly for what follows, this so-called "standard" MO-GP setup does not include any mutation operator.

The second series of experiments conducted in [3] on the set of three problems did, however, employ mutation as well as a diversity preserving mechanism in phenotype space.

The third series of experiments employed both phenotype and genotype diversity as well as mutation. It is worth noting that in terms of generality, phenotype diversity is only readily applicable to discrete problems.

Based on our analysis of the methodology of de Jong and Pollack,we have carried-out a comprehensive and systematic series of experiments, focusing especially on the rôles of mutation and diversity preservation in population collapse.

3. METHODOLOGY

3.1 Genetic Programming Procedure

We have employed generational GP with a fixed population size of 100; the initial population was produced by the ramped half-and-half method, with 50% of the individuals having a tree depth of 7 and the other 50% having a random, uniformly-distributed depth in the range $[1 \ldots 7]$. Selection was by Pareto ranking using the method of Fonseca and Fleming [6]. The tree function nodes used were:

- AND, OR, NAND and XOR for the boolean 6-parity problem.
- The basic binary arithmetic operators: $+, -, \times$ and $\div$ for the machine learning experiments.

In order to preserve the best individuals in the population from one generation to the next, the 34 top ranked individuals were copied, unaltered (This also retains some linkage to the evolutionary strategy of de Jong and Pollack.) The remaining 66 members of the new population were produced by genetic operations on individuals selected (with replacement) from the full population of 100. Crossover, which was always applied, was implemented using the depth-fair method of Ito et al. [7].

We have optionally employed mutation; exact details of when and where are set-out in Section 4. Where mutation is used, again it is always applied. Like the crossover operator, the mutation operator used the depth-fair mechanism of Ito et al. [7] where we first select a depth, in the tree at which to perform mutation. Then, one of the sub trees at this depth is selected biased by complexity – that is, we prefer to mutate the larger subtrees. Having selected a subtree, it is replaced with a new, randomly generated subtree. Note that as a consequence of using the depth-fair subtree selection mechanism in mutation, the root node of a tree is selected $1/2^1 = 50\%$ of the time. If the root node is selected for mutation, a whole new tree of fixed depth 7 is generated. If a proper subtree (as opposed to the root node) is selected, it is replaced by a new subtree of fixed depth, N_{mut} – see Section 4 for the various experimental values of N_{mut} investigated.

The two objectives comprising our MO fitness vector were: tree node count and a measure of fitness over the training set. In addition, we have optionally employed a genotypic diversity preserving mechanism identical to that of de Jong and Pollack [3]. Two trees were 'overlaid' geometrically and a 'distance' between the two trees defined by counting the number of spatially matching nodes which are dissimilar, and then normalizing by the size of the smaller tree. This 'distance' between the trees is used as a third objective to modify the ranking of the population - see [3]. See Section 4 for details.

Each GP run was continued for a fixed number of tree evaluations: 10,000 for the 6-parity problem and 20,000 for the machine learning tasks.

3.2 Datasets

Two problem domains were considered: Firstly, we report the results of exploring the 6-parity boolean learning task studied in [3] where the performance objective was the number of incorrect outputs over the (exhaustive) training set.

Second, we have investigated the feature extraction problem on benchmark machine learning tasks. For the machine learning portion of this work, four well-known datasets were taken from the UCI Machine Learning Repository [1]: Pima Indians Diabetes (PID), BUPA Liver Disorders (BUPA), Glass, reduced to a two-class problem to differentiate between float and non-float glasses (GLASS) and Wisconsin

Breast Cancer (WBC). Half of each dataset was randomly selected as the training set and the remaining half used as an independent validation set.

GP was used to evolve a feature extraction stage for the above classification problems by projecting each n-dimensional pattern vector into a 1D decision space and selecting an optimal threshold in the decision space by separate search. The resulting classification error (0/1 loss) over the training set was used as a fitness objective alongside the tree's node count.

4. RESULTS

4.1 6-Parity Boolean Problem

As an initial investigation, we have repeated the first experiment of de Jong and Pollack [3] on the 6-parity boolean problem. Using the GP setup in Section 3.1 but *without* mutation or the genotypic diversity mechanism, we always observed population collapse within five generations. Thus we were readily able to reproduce the results of de Jong and Pollack.

Next, we considered the performance of the basic GP setup but *with* mutation ($N_{mut} = 3$) but again, without the genotypic diversity mechanism. The effect of adding mutation was to prevent population collapse; repeated runs showed no sign whatsoever of collapse. These initial two experiments are summarized in Figure 1 which shows the fraction of population individuals of single node as a function of the number of tree evaluations. The influence of mutation is very apparent from this figure. Thus our first conclusion is that although the *phenotypic* diversity preservation reported in [3] was indeed able to prevent population collapse, the more common operation of mutation is able to achieve exactly the same result. (Furthermore, evolution with mutation is able to repeatedly find the known optimal solution of the problem comprising 13 nodes [3].)

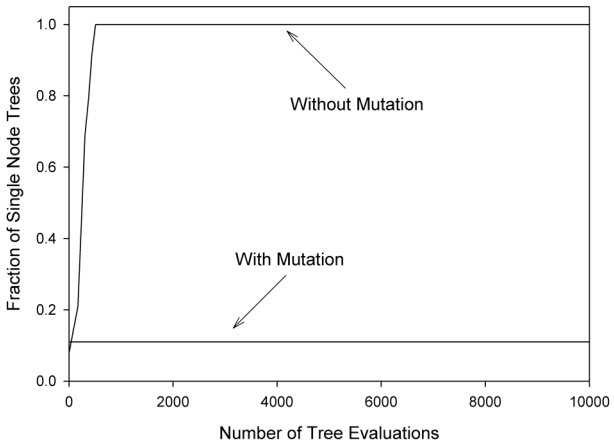

Figure 1: Typical fraction of single node trees in the population as a function of the Number of Tree Evaluations; 6-parity problem.

4.2 Machine Learning Problems

Although we have examined four datasets in this work, we show only the results from the BUPA dataset; all four datasets yield very similar results and exactly the same conclusions would be drawn from the results on any of the four. For brevity, we show only typical results – qualitatively, all the results are more-or-less identical.

As a first step, we took the GP setup described in Section 3.1 *with* the mutation operator but *without* any diversity preservation mechanism. The N_{mut} parameter, the depth of replacement trees was set to 3; that is, the mutation operator replaced non-root subtrees with a new, random subtree of fixed depth, 3. When selected for mutation, the root node was replaced with a tree of depth 7 – see Section 3.1. We take this configuration as our baseline GP setup for the machine learning tasks.

In complete accord with all our previous experience, this baseline GP procedure converged on every one of ten runs; we saw absolutely no sign of population collapse. Typical training and validation errors averaged over the ten independent runs are plotted in Figure 2; reassuringly, the validation error values are quite close to those of the training error, although as expected, somewhat larger. This implies reasonable generalization performance. (The minimum value of error in Figure 2 compares to a value of 0.28, the lowest error obtained by Lim et al. [10] for the BUPA dataset in a comparative study of thirty three conventional classification algorithms. The statistical significance of this difference, however, remains to be established and since it is not central to this work, we have ignored such factors. The best validation errors we observed on the other three datasets are similarly numerically smaller than the best results reported in [10]. We thus infer that the classification errors we have obtained are, at very least, proximate to the state-of-the-art.)

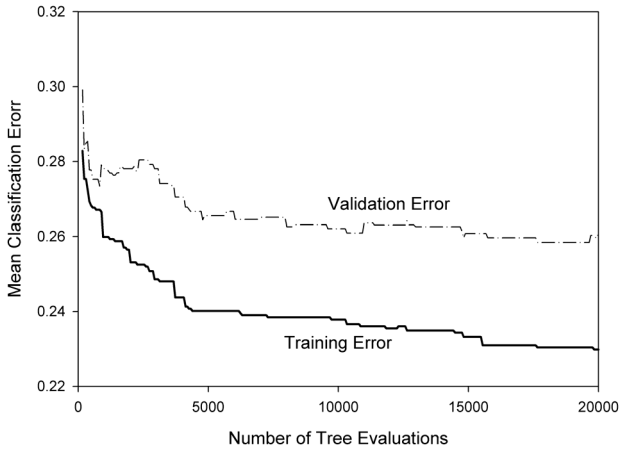

Figure 2: Typical mean test and validation classification errors as a function of the Number of Tree Evaluations. BUPA dataset.

We have noted in Section 2 that the "standard" GP algorithm of de Jong and Pollack [3] omitted mutation. We have

therefore repeated our baseline experiments reported above but this time, *without* mutation. The result in each of 10 independently initialized runs was rapid and complete population collapse, exactly as observed in [3]. To characterize this, Figure 3 shows the typical fraction of single node trees in the population as a function of the cumulative number of tree evaluations. The curve for the baseline algorithm with mutation shows that a small – but non-zero – fraction of the population comprises a single node; this is reasonable if the MO optimisation is properly sampling the Pareto front. The upper curve in Figure 3, however, is for the baseline algorithm without mutation. Under these conditions, there is rapid and complete population collapse, as reported in [3].

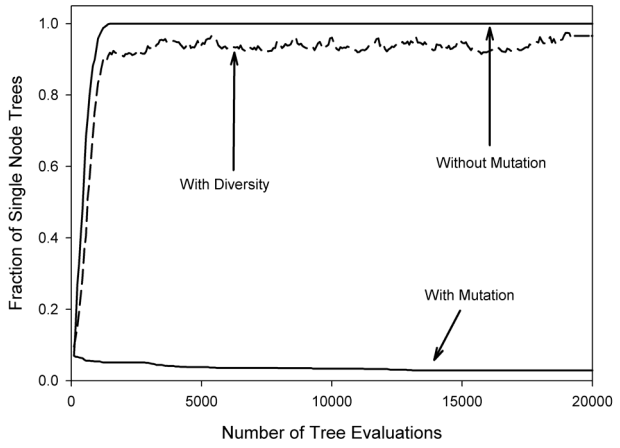

Figure 3: Typical fraction of single node trees in the population as a function of the Number of Tree Evaluations; BUPA dataset.

As far as we are aware, this critical rôle of mutation in preventing collapse in MO-GP has not previously been identified. To reinforce this point, Figure 4 shows the average tree size as a function of the cumulative number of tree evaluations for the baseline algorithm. With mutation, the mean tree size rapidly settled to a fairly constant value of ≈60 nodes. Without mutation, the whole population rapidly collapses to single node trees. We therefore conclude that mutation has a vital rôle in preventing collapse. Further, we infer that the reason de Jong and Pollack observed collapse in their initial experiments was due to the somewhat unconventional omission of mutation from their "standard" algorithm.

To gain greater insight into the means by which mutation prevents collapse, we have varied the mutation operator in our baseline algorithm. The results reported above were obtained by replacing non-root subtrees with new, randomly-generated subtrees of fixed depth, $N_{mut} = 3$. Initially, we varied the value of N_{mut} in the range $[1 \ldots 5]$ to no effect. Similarly, replacing a selected subtree with a new subtree of exactly the same size did not produce collapse. Even replacing the subtrees with a single (obviously, terminal) node did not trigger collapse. Consequently, we infer the depth of the non-root mutating subtree is not critical.

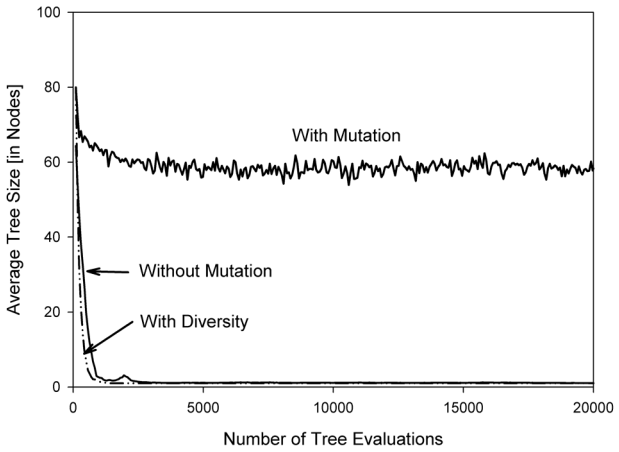

Figure 4: Typical average population tree size as a function of the of the Number of Tree Evaluations; BUPA dataset.

To further vary the mutation operator, if it selected the the root node of a tree for replacement, we ignored the operation. Under our depth-fair mutation scheme, each level in a tree of depth, d_{max}, is assigned a probability of being selected, with the root node being selected 50% of the time [7]. Rejecting the possibility of selecting the root node led to rapid population collapse, regardless of the value of the N_{mut} parameter. This rather surprising observation led us to further investigate the effects of mutation on the population.

Since we have established that i) removing mutation altogether leads to collapse and that ii) disallowing mutation of the root node also leads to collapse, we have isolated the influence of mutation on the size of the trees in the population.

Considering the n-th generation, the requisite number of children are generated to complete the $(n+1)$-th generation by repeatedly selecting pairs of parents, performing crossover to produce two offspring, C_1 and C_2, and then mutating each offspring independently to produce C_1' and C_2' which are finally added to the $(n+1)$-th generation. We can define the difference in tree sizes produced by mutation, Δ_i as:

$$\Delta_i = Size(C_i') - Size(C_i)$$

where $Size(X_i)$ is the node count of the tree, X_i and $i \in [1, 2]$. Note that Δ_i can be either positive or negative. By averaging Δ_i over all 66 mutation operations which produce new individuals for the $(n+1)$-th generation, we can assess the impact of the various mutation schemes on the growth of the trees between generations.

The results of measuring the mean changes in population tree size against generation number for a typical GP run are shown in Figure 5 where mutation is allowed to replace the root node. Considering the effects of mutation on its

own (upper curve), there is a mean change of 20-30 nodes at each generation; thus mutation is tending to *increase* the size of trees. The lower plot in Figure 5 shows the *net* mean change in tree size between successive generations which is effectively zero. Thus with root node mutation, the mean size of individuals in the population remains effectively constant: there is neither bloat nor collapse.

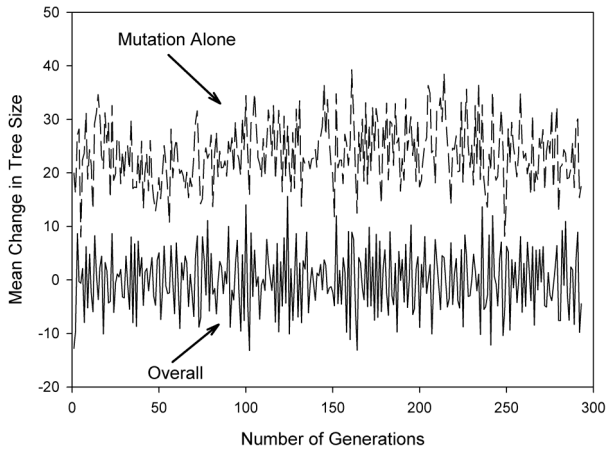

Figure 5: Typical mean change per generation in tree size as a function of the Number of Generations allowing root node mutation. Upper curve: Mean change due to mutation alone. Lower curve: Overall. BUPA dataset.

Repeating the above experiment but ignoring root node mutations produces the results in Figure 6; this is the situation that leads to population collapse. (Note the change in both scales in Figure 6 compared to Figure 5.) It is clear that here, mutation results in a *decrease* in mean tree size which thus falls between successive generations leading to population collapse. (The mean change in tree size falls to zero in Figure 6 since if the population has collapsed and all trees are of minimum size, one, then there can be no further reduction in tree size.)

We therefore conclude that the maintenance of a legitimate population in MO-GP is achieved by keeping a balance between the parsimony pressure exerted by the fitness-based selection, and the tendency of mutation (including mutation of the root node) to increase tree sizes. (The crossover operation, of course, conserves total node numbers in the population and therefore produces identically zero mean growth in the trees.) If the mutation operation does not tend to exert a bloating pressure on the population and therefore counteract the parsimony pressure, population collapse ensues. Conversely, if the mutation operation tends to increase tree sizes and this is not offset by a suitable parsimony pressure, bloat results. For example, Langdon and Poli [9] have shown that mutation alone can produce bloat.

Interestingly, although mutation tends to grow trees by a fixed (average) amount, the multiobjective parsimony pressure is able to adapt to apply sufficient influence to prevent bloat but not so much as to cause collapse. We suggest this

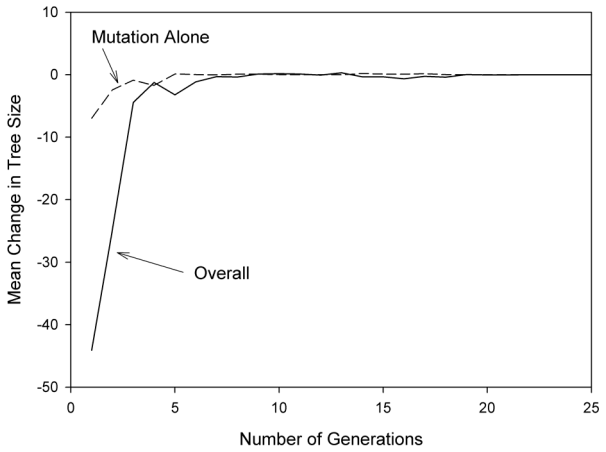

Figure 6: Typical mean change per generation in tree size as a function of the Number of Generations *not* allowing root node mutation. Upper curve: Mean change due to mutation alone. Lower curve: Overall. BUPA dataset.

is related to the parameter-less nature of the MO parsimony – it prefers *smaller* solutions rather than solutions below a specified size[1].

Finally, we have investigated de Jong and Pollack's use of genotypic diversity preservation to prevent collapse [3]. Although these authors report no collapse when a *phenotypic* diversity measure is used, this mechanism is only really appropriate for problems with discrete objectives. For a continuous objective it is necessary to define some *scale* over which two solutions are considered to be 'identical' – this need to have prior knowledge of a suitable scale also makes crowding/sharing techniques in conventional genetic algorithms problematic. We have consequently only investigated *genotypic* diversity which was added as a third objective in the form of an 'edit distance' between two trees – see Section 3.1. Including this third diversity objective (and omitting mutation) resulted in population collapse, as can be seen from the chained plots in Figures 3 and 4 although there was some minor but unimportant improvement compared to using crossover alone. We conclude that a genotypic diversity objective, at least in the form implemented in [3], is ineffective at preventing collapse.

5. DISCUSSION

Our starting point in this study was a re-appraisal of the work of de Jong and Pollack [3]. We have shown that although these authors were able to avoid population collapse in the 6-parity boolean problem in their first series of experiments by using a phenotypic diversity preservation mechanism, they could probably have achieved exactly the same

[1]In fact, we have data – which will be published elsewhere – that suggest changing the *size* of the new tree which replaces the root node under mutation does affect the mean tree size in the population. This greatly influences the time required for a GP run although not, it appears, the *quality* of the final solutions.

end by using mutation. De Jong and Pollack then demonstrated the effectiveness of their diversity preservation approach for preventing population collapse on a series of other problems on which they used not only diversity preservation but mutation as well. Finally, they employed a phenotypic diversity measure, mutation and a third, genotypic diversity objective to argue for the effectiveness of diversity preservation; we have shown the genotypic diversity measure to be of little use in preventing collapse on any of the machine learning problems considered here. All our work suggests that had de Jong and Pollack conducted a series of experiments which introduced the additional operations *one-at-a-time*, they might well have found that mutation alone can preserve diversity. We therefore have misgivings about the methodology of de Jong and Pollack which casts some doubts over their conclusions.

De Jong and Pollack [3] are not, of course, the only authors to have reported population collapse. Langdon and Nordin [8] have observed collapse despite using mutation. Similarly, Ekárt and Németh [4] have seen population collapse which they suppressed by biasing the Pareto domination relation to give (arbitrarily) greater preference to better performing individuals compared to smaller individuals. One possible factor could be these authors' use of tournament selection rather than proper Pareto ranking [3, 6]. We have conducted no tests with tournament selection since the small sample effects of the comparison set are well-known to produce highly variable selection. A more likely explanation is the detailed implementations of the mutation operators used in [8, 4] which may not have provided the necessary growth tendency in tree size which the present work suggests is key to preventing collapse. It is noteworthy that in order to obtain useful results, Langdon and Nordin abandoned their second, parsimony objective very early in their work [8], thus, we suggest, obviating the need to maintain the internal balance mechanism.

The key issue that seems to arise from Section 4 is that the mutation operator needs to produce (on average) a net *increase* in tree complexity which counterbalances the parsimony objective. (Conversely, for single objective GP with no parsimony objective, this is highly undesirable since it will lead to bloat.) Certainly a number of other authors have noted the importance of mutation in maintaining diversity (e.g. Poli and Langdon [11]) although we believe the present paper is the first report which specifically addresses the pivotal rôle of mutation in *multiobjective* genetic programming. In this work we have selected the mutation point in the tree using the depth-fair method of Ito et al. [7]. A worthy question is: Whether other implementations of mutation, such as point mutation, give different results? Certainly other workers such as [4, 8] have seen population collapse despite using mutation whereas ourselves and at least one other group working in MO-GP [5] have never seen this. This suggests that a more detailed consideration of the design of mutation operators for multiobjective environments should be carried-out; this is currently the subject of further research.

6. CONCLUSIONS

In this paper we have investigated the factors surrounding the collapse to all single node trees of a population in multiobjective genetic programming. In particular, we have re-examined the work of de Jong and Pollack [3] on the use of diversity preservation to prevent population collapse.

We observe that mutation alone is able to prevent collapse, specifically, a mutation operator which tends to produce a positive mean increase in tree size per generation. Under these circumstances, mutation produces a tendency in the population to bloat which is counterbalanced by the parsimony pressure exerted by the fitness-based selection process. Further results on this work will be published elsewhere.

We have also explored the use of the genotypic diversity objective of de Jong and Pollack to prevent collapse; we find this to be ineffective at preventing population collapse. Coupled with the failure of genotypic diversity measures, we have also pointed-out a number of methodological shortcomings in the work of de Jong and Pollack [3] which may cast doubt on their conclusions.

7. ACKNOWLEDGMENTS

We are indebted to Dr. Yang Zhang for providing much of the computer code used in this study and for invaluable discussions.

8. REFERENCES

[1] C. L. Blake and C. J. Merz. UCI Repository of Machine Learning Databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.

[2] S. Bleuler, M. Brack, L. Theile, and E. Zitzler. Multiobjective genetic programming: Reducing bloat using SPEA2. In *Congress on Evolutionary Computation*, pages 536–543, Seoul, Korea, 2001. IEEE.

[3] E. D. de Jong and J. B. Pollack. Multi-objective methods for tree size control. *Genetic Programming and Evolvable Machines*, 4(3):211–233, 2003.

[4] A. Ekárt and S. Z. Németh. Selection based on the Pareto nondomination criterion for controlling code growth in genetic programming. *Genetic Programming and Evolvable Machines*, 2(1):61–73, 2001.

[5] P. J. Fleming. Personal communication, 2007.

[6] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *5^{th} International Conference of Genetic Algorithms*, pages 416–423, San Mateo, CA, 1993. Morgan Kaufmann.

[7] T. Ito, H. Iba, and S. Sato. Non-destructive depth-dependent crossover for genetic programming. In *1^{st} European Workshop on Genetic Programming*, pages 14–15, Paris, France, 1998. Springer-Verlag.

[8] W. B. Langdon and J. P. Nordin. Seeding GP populations. In R. Poli, W. Banzhaf, W. B. Langdon, J. F. Miller, P. Nordin, and T. C. Fogarty, editors, *3^{rd} European Conference on Genetic Programming (EuroGP'2000)*, pages 304–315, Edinburgh, 2000. Springer-Verlag.

[9] W. B. Langdon and R. Poli. Fitness causes bloat: Mutation. In *1^{st} European Workshop on Genetic*

Programming, pages 37–48, Paris, France, 1998. Springer-Verlag.

[10] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40(3):203–229, 2000.

[11] R. Poli and W. B. Langdon. Genetic programming with one-point crossover and point mutation. Technical Report CSRP-97-13, Department of Computer Science, University of Birmingham, Birmingham, UK, 1997.

[12] K. Rodríguez-Vázquez, C. M. Fonseca, and P. J. Fleming. Identifying the structure of non-linear dynamic systems using multiobjective genetic programming. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 34(4):531–547, 2004.

[13] Y. Zhang and P. I. Rockett. Evolving optimal feature extraction using multi-objective genetic programming: A methodology and preliminary study on edge detection. In H.-G. Beyer, U.-M. O'Reilly, D. V. Arnold, W. Banzhaf, C. Blum, E. W. Bonabeaum, E. Cantu-Paz, D. Dasgupta, K. Deb, J. A. Foster, E. D. de Jong, H. Lipson, X. Llora, S. Mancoridis, M. Pelikan, G. R. Raidl, T. Soule, A. M. Tyrrell, J.-P. Watson, and E. Zitzler, editors, *Genetic and Evolutionary Computation Conference (GECCO 2005)*, pages 795–802, Washington, DC, 2005. ACM Press.

[14] Y. Zhang and P. I. Rockett. Feature extraction using multi-objective genetic programming. In Y. Jin, editor, *Multi-Objective Machine Learning*. Springer, Heidelberg, 2006.