Novelty-based Fitness: An Evaluation under the Santa Fe Trail*

John Doucette and Malcolm I. Heywood†

7–9 April 2010

Abstract

We present an empirical analysis of the effects of incorporating novelty-based fitness (phenotypic behavioral diversity) into Genetic Programming with respect to training, test and generalization performance. Three novelty-based approaches are considered: novelty comparison against a finite archive of behavioral archetypes, novelty comparison against all previously seen behaviors, and a simple linear combination of the first method with a standard fitness measure. Performance is evaluated on the Santa Fe Trail, a well known GP benchmark selected for its deceptiveness and established generalization test procedures. Results are compared to a standard quality-based fitness function (count of food eaten). Ultimately, the quality style objective provided better overall performance, however, solutions identified under novelty based fitness functions generally provided much better test performance than their corresponding training performance. This is interpreted as representing a requirement for layered learning/ symbiosis when assuming novelty based fitness functions in order to more quickly achieve the integration of diverse behaviors into a single cohesive strategy.

1 Introduction

A novelty-based fitness measure is one inspired by inter-species evolution, wherein individuals are awarded *not* for the *quality* of their behavior, but for discovering behaviors in which no/ few individual are presently engaged i.e., phenotypic behavioral diversity [5]. Thus, in a pure novelty-based fitness function individuals are rewarded based only on how different their observed phenotypic behaviors are from the rest of the population. Conversely, an objective or quality based measure of fitness would reward individuals for finding solutions that minimize some concept of 'error'; thus the population as a whole might converge to solutions that are behaviorally very similar e.g., all individuals returning the same

*EuroGP, LNCS 6021 – Copyright 2010 Springer-Verlag

†Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada

classification count.[1] Previous works have considered the utility of niching operators to provoke diversity maintenance, but the utility of such operators under Genetic Programming (GP) is not necessarily straight forward. However, recent work suggests that purely novelty-based evolutionary searches can be particularly effective [5, 1, 10, 9].

Recent research using novelty-based fitness measures for genetic programming include both the aforementioned novelty only formulation and combined novelty–quality fitness functions. Specifically, several authors have employed combined novelty–quality objectives under the domain of classification [7, 8]. This is very different from the standard approach to ensemble methods as the novelty objective makes explicit the desire to avoid solely 'cherry picking' the exemplars that are easy to classify. The resulting team of individuals exhibit explicitly non-overlapping behaviors [8]. Other works consider the effects of novelty-based fitness in detail, but have considered post training generalization [5, 9, 10]. This presents a problem, as the effects directly attributable to novelty-based fitness measures remain unknown, especially the effects on model generalization. With this in mind, we present a detailed empirical study of the effects of various combinations of quality and novelty-based fitness metrics on training and test performance using a classic GP benchmark, the Santa Fe Trail. Two novelty-based fitness measures are considered, as well as a simple combination novelty–quality measure. Particular attention is paid to the effects of novelty-based fitness on generalization performance.

2 Background Concepts

2.1 Generalization

Generalization measures the relative change in behavior of a candidate solution vis-a-vis the environment it was trained on versus an independent set of environments on which testing is performed [3, 4]. In maze navigation the navigator often starts in a fixed location and attempts to reach a fixed exit. A system which memorizes the correct sequences of movements and rotations to navigate a specific maze is nearly useless, since it would fail to navigate any maze with an even slightly different structure, starting place, or destination. In contrast, a navigator which has learned a true maze navigating behavior would be capable of 'generalizing' the training scenario to a wide range of previously unseen maze architectures. Needless to say, decisions made regarding representation, credit assignment and cost function all impact on the resulting generalization ability of solutions [4]. Several benchmark problems – e.g., parity and the multiplexer – are frequently deployed without assessing generalization at all [2]. In this work we focus on the contribution of the cost function alone, and keep representation and credit assignment processes constant.

[1] This might imply some diversity relative to the exemplars correctly classified, although this is generally not explicitly articulated in the fitness function.

2.2 Novelty Search

Novelty-based search heuristics are those which reward the discovery of unique behaviors, in contrast to quality-based heuristics which reward individuals that are believed to be closer to the domain goal. Thus quality/ goal style objectives tend to reinforce cherry picking of the scenarios that are easier to solve. The assumption being that this forms a learning gradient from which stronger behavior develops. Conversely, a novelty style objective is more effective at maintaining population diversity and as a consequence might lead to a better supply of building blocks for providing a more general solution. Recent work suggests that novelty-based fitness measures can outperform quality-based fitness measures in maze navigation [5, 10, 9] since solutions are under no pressure to cluster around local maxima in the search space. Other work has incorporated solution novelty into a Pareto multi-criteria objective function to promote problem decomposition [8], rewarding individuals not for correctly labeling all the data, but for labeling some unique subset of it correctly.

That said, the fact that novelty-based search places less emphasis (or none at all) on finding goal orientated solutions raises concerns about its effects on generalization error. It might be supposed that solutions created without any emphasis on correctness might be correct only coincidentally rather than by virtue of having learned a particular task, and that novelty-based search would produce solutions that fail to generalize.

2.3 The Santa Fe Trail

The Santa Fe Trail is a widely used benchmark in genetic programming [2, 3, 4]. The problem consists of evolving a controller for an "artificial ant" on a toroidal grid such that the ant correctly follows a trail of food. An ant solves the problem if it eats all of the food on the grid within a certain number of time steps. At each time step the ant can either change the direction it faces by 90 degrees, or move forward one square. As discussed in [4], the ant problem is "deceptive", meaning that there are many local maxima in the space of possible controller programs. These local maxima result from ants which loose the trail and stumble into more food at a later point in the trail. Ants of this type are not very close to solving the problem i.e., they are very unlikely to find the stretch of trail that they missed within the time limit. In contrast, ants that eat more of the trail in order may not eat as much food in total. Consequently, GP does not perform significantly better than random search on this problem, since the most commonly used heuristic (more food = higher fitness) is deceptive [4]. Previous work suggests that novelty-based search should be more effective in deceptive problems [5]. Consequently, the Santa Fe Trail may be a good choice for determining effects of novelty-based search on generalization.

In addition to being a deceptive problem, the ant trail has several advantages. First and foremost is a previously established method of measuring generalization error [3]. This entails generating a set of random trails which share certain properties with the Santa Fe trail, including maximum distance between food,

shapes of corners in the trail, and density of food in the trail. The trails may be of different lengths, and may have differing amounts of food. An ant which has successfully learned a general solution to the Santa Fe trail should do well on these trails, while one which has learned specialized strategies (memorizing the turns in the trail) will not. Additionally, the Santa Fe Trail has an easily representable space of program behaviors, namely the order in which the food on the trail is eaten. An ant which follows the trail diligently will thus end up with a very different behavior from one that tessellates the grid, and those behaviors may be concisely represented and quickly compared.

3 Methodology

To test the effects of novelty-based fitness on generalization error, we considered two different methods for determining novelty, hereafter denoted Methods 1 and 2. In this context a novel solution is one in which the ant consumes food on the trail in a sequence that differs from all previously observed behaviors as summarized in terms of a pair-wise similarity metric. Needless to say, the metric employed for the pairwise comparison has a significant impact on the quality of the resulting evaluation [1], with Hamming Distance being assumed in this work i.e., one of the two recommended metrics identified by the earlier study. This also raises the question as to how dissimilar individuals need to be before they are considered novel. Two methods are considered. In Method 1 (Algorithm 1), an archive of fixed size stores "archetypes"; or solutions which represent broad classes of behaviors. Individuals are added to the archive if the behavioral difference between them and all archetypes is larger than the difference between the closest pair of archetypes presently in the archive (line 7). In this case the individual will replace one of the two archetypes which are most similar to each other, causing the inter-archetype difference to increase monotonically over the course of a run (line 11). Archives were fixed at a size of 100 archetypes in all runs. In Method 2 (Algorithm 4), an archive of variable size stores archetypes that differ from each other by at least some constant amount Δ_{min} (as in [5, 10]). New individuals are added to the archive if their behavior differs from that of every archetype presently in the archive by at least Δ_{min} (line 10). In the case of both methods, individuals are awarded fitness as a function of how far their behavior is from that of any archetype presently in the archive, with radically different behaviors receiving the highest fitness and those whose behavior is identical to that of some archive member receiving the lowest fitness.

Two additional methods are now introduced to provide a relative baseline on the performance of the purely behavioral performance functions. Method 3 uses the finite archive of Method 1 in a combined equally weighted contribution from novelty and quality objectives, or the average of the fitness returned from the archive method and the fitness returned by the typical "eat most food" fitness evaluation, i.e. $Fitness = \frac{F_{novelty}+F_{quality}}{2}$. Thus, a solution that eats all the pieces of food in a completely unique order will have a fitness of 1; whereas

Algorithm 1 Novelty-Based Fitness Evaluation with finite archive. Returns the fitness of an individual and adds it to the archive if it qualifies.

```
1: Let A be an archive storing > 0 individuals.
2: Let I be an individual.
3: procedure FITNESS(A,I)
4:     mindiff=+∞
5:     for all a ∈ A do
6:         if mindiff > ham(a, I) then mindiff = ham(a, I)
7:         end if                                  ▷ ham(. , .) (Algorithm 2)
8:     end for
9:     if mindiff > A.current_mindiff then
10:        insert I replacing A.minidx
11:        recompute_mindiff(A)          ▷ recompute_mindiff(.) (Algorithm 3)
12:    end if
        return mindiff
13: end procedure
```

Algorithm 2 Pairwise Hamming Distance Estimation.

```
1: Let i and j be individuals
2: Let {i, j}.foodvect be vectors showing the order in which the individual ate
   food
3: Let {i, j}.foodvect(n) = (x, y) iff the n^th piece of food eaten was at (x, y)
4: procedure HAM(I,J)
5:     if |i.foodvect| < |j.foodvect| then swap i and j
6:     end if
7:     hamsum = 0
8:     for z = 1to|j.foodvect| do
9:         if i.foodvect ≠ j.foodvect then hamsum++
10:        end if
11:    end for
12:    hamsum + = |i.foodvect| − |j.foodvect|
        return hamsum
13: end procedure
```

Algorithm 3 Recomputing of the minimum difference between any two members of the archive.

```
1: Let A be an archive storing > 0 individuals.
2: recompute_mindiff(A)
3: if |A| < A.maxsize then
4:     A.current_mindiff = 0
5:     A.minidx = A.currentsize
6: end if
7: mindiff=+∞
8: for all i, j ∈ A; where i ≠ j do
9:     if mindiff > ham(i, j) then                ▷ ham(. , .) (Algorithm 2)
10:        A.current_mindiff = ham(i, j)
11:        A.minidx = i
12:    end if
13: end for
```

Algorithm 4 Novelty-Based Fitness Evaluation with Infinite Archiving. Returns the fitness of an individual and adds it to the archive if it qualifies. $\Delta_{min} = 10$ for all our runs.

```
1: Let A be an archive storing > 0 individuals.
2: Let I be an individual
3: Let Δ_min be a constant s.t. for i and j ∈ A, ham(i, j) >= Δ_min
4: procedure FITNESS(A,I, Δ_min)
5:     mindiff = +∞
6:     for all a ∈ A do
7:         if mindiff > ham(a, I) then mindiff = ham(a, I)
8:         end if                                 ▷ ham(. , .) (Algorithm 2)
9:     end for
10:    if mindiff > Δ_min then insert I → A
11:    end if
       return mindiff
12: end procedure
```

Table 1: GP Parameters, based on [4, 2]

Parameter	Value
Terminal Set	Left, Right, Move_Ahead
Function Set	If_Food_Ahead, Prog2, Prog3
Selection Method	Stochastic Elitism
Max Time steps	400
Max Program Depth	17
Initialization	"Ramped half and half", max depth 6
Reproduction Operators	90% Mutation, 10% Reproduction
Population Size	1000
Maximum Generations	50

an individual that eats half the food in a previously observed order will have a fitness of 0.25 (0 for having the same behavior as some archive member, 0.25 for eating half the food).

All three of the above methods were implemented with a modified version of the lilgp package [11]. This provides us with the original code for the Santa Fe Trail and therefore a quality-based method for fitness evaluation or items of food eaten (Method 4). The only substantial modification made to the code other than changing the fitness functions was to allow solutions to be run on test environments after the completion of training.

To compare the four fitness methods, at the end of each run, the individual who had eaten the most food was selected as the champion. We evaluated the champions on a fixed set of 100 test trails generated according to the algorithm in [3], with results measured in terms of the percentage of food eaten on each trail. Each method was run with 500 unique random seeds.

We selected parameters to avoid optimizing any method at the expense of the others. The archive size for Method 1 and Δ_{min} for Method 2 were selected by trying 10 values on single runs with the same random seed and adopting the best performing parameterization. The 10 values were selected at even intervals over (10,200) for archive size, and (5,40) for Δ_{min}. The values of the other parameters are the defaults found in the Santa Fe Trail implementation provided with [11], with the single change of swapping crossover for mutation in the reproduction operators, as in [4]. Table 1 summarizes the complete parameterization.

4 Results and Analysis

The results have been separated into training, testing, and generalization performance. For simplicity, all results were tested for statistical significance at a confidence level of 95%, with a Bonferroni correction used to compare the means of each pair of samples. A Jarque-Bera test was used to determine whether data were normally distributed. One-way ANOVA tests followed by student t-tests were used to compare normally distributed data, while Kruskal-Wallis tests fol-

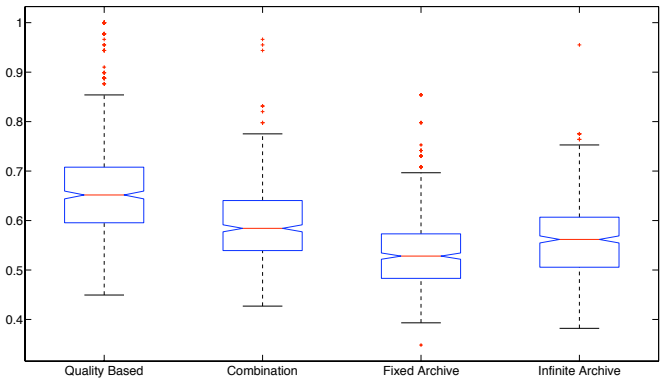

Figure 1: Plot showing 1st, 2nd, and 3rd quartiles for the distribution of training performance. Whiskers identify the limit of points within 1.5 times the inter quartile distance with crosses marking any outliers.

lowed by Wilcoxon Rank-Sum tests were used to compare data which was not normally distributed. In the graphs presented, the limited archive approach of Method 1, the infinite archiving approach of Method 2, the combination novelty–quality formulation of Method 3, and the standard quality-based approach of Method 4 are denoted by the labels "Fixed Archive", "Infinite Archive", "Combination", and "Quality-Based" respectively.

4.1 Training Performance

We gathered training data by measuring the proportion of food eaten by each champion individual in the training enviroment (the Santa Fe Trail). Data from all four methods were normally distributed, and statistically significant differences were returned between all pairs of methods (Figure 1). The quality based function performed best, eating 66% of the food on average, compared with 59%, 56%, and 53% for the combination, infinite archive and fixed archive respectively. Relative to the original Santa Fe study of Koza [2] we note that the level of performance is generally lower. However, this is in part likely due to adopting the 400 time step limit reported by Koza whereas this was apparently 600 in his experiments (see the commentary in [4]).

4.2 Testing Performance

We produced test data by measuring the proportion of food eaten by the champion from each run on 100 test environments. The champions are compared using summary statistics of the collected data, in particular the median, maximum, and minimum proportion of food eaten by each champion on the test environments. The median performance of the champions was normally distributed for all 4 methods. We did not find a statistically significant difference

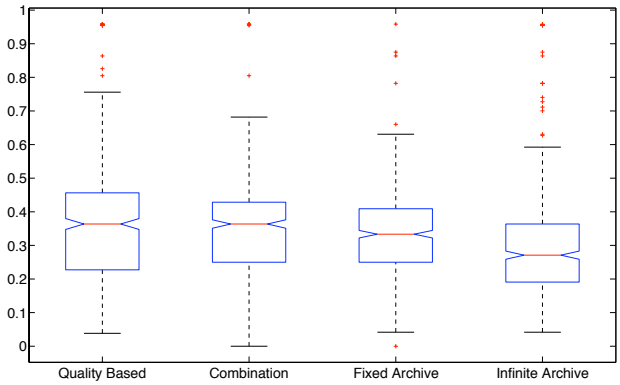

Figure 2: Plot showing first, second and third quartiles for the distribution of median test performance. See Figure 1 caption for interpretation of the whiskers.

between the performance of the quality-based and combination methods, but did find significant differences between the performance of all other pairs of samples (Figure 2). Both the combination and quality-based methods consumed an average of 35% of the food or more in at least half their runs, while the average was only 32% for the fixed archive method, and 29% for the infinite archive method.

Results for maximum and minimum performance were not normally distributed (Figure 3). We found significant differences between all pairs of samples for maximum performance, with no more than 78% of food being eaten on any test enviroment, on average, for the quality-based method, 77% for the combination method, and only 73% and 71% for the fixed and infinite archive methods. For minimum performance, we found a statistically significant difference between the combination and infinite archive methods, and between the quality-based method and both the fixed and infinite archive methods. On average, champions ate no less than 8% of the food on any test enviroment using the quality-based method, 7% using the combination method, and 6% using the fixed or infinite archiving methods.

4.3 Paired Generalization Error

The final measure considered is paired 'generalization error' or the relative normalized[2] difference between training and test performance of the *same* individual. A drop in performance is generally assumed to appear between training and test performance. However, as this difference increases lack of generalization is a more likely candidate. Hence, higher positive differences are taken as indicating that the model has learned to memorize the Santa Fe Trail in

[2]By 'normalized' we imply that the number of food items can vary under test conditions [3], hence both training and test performance are normalized relative to the total of food items available in that scenario.

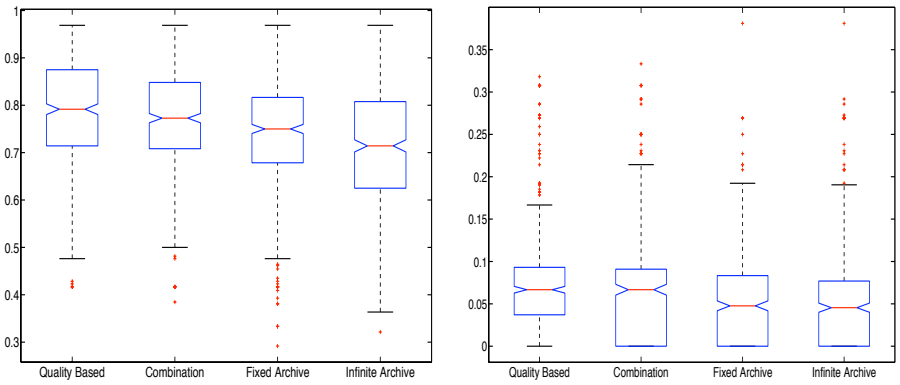

Figure 3: Plot showing first, second and third quartiles for the distribution of maximum (left) and minimum (right) test performance. See Figure 1 caption for interpretation of the whiskers.

particular; whereas negative values indicate that more food is eaten under the test condition than in training. We found the median generalization error for champions from all four methods to be normally distributed, with statistically significant differences between all pairs of models (Figure 4). The fixed archive method has the lowest median generalization error, with a difference in the proportions of food eaten in the training and median test environments being 21% on average. The combination method has a mean difference of 24%, while the infinite archive has 27%, and the quality based method 31%.

This distribution is further emphasized by considering performance from the perspective an interquartile distance function. Letting training and test performance as a whole be two 'clusters' and comparing the normalized inter cluster distance illustrates the degree to which test and training performance diverge. Thus, given the standard inter cluster distance metric of,

$$\frac{\mu(test) - \mu(train)}{\sqrt{\sigma^2(test) + \sigma^2(train)}} \tag{1}$$

where μ and σ^2 are the mean and variance of normalized 'training' and 'test' performance. Figure 5 summarizes the corresponding inter cluster distance for each fitness function. The strong correlation between training and test performance under the Fixed archive version of novelty objective is immediately apparent. Conversely, the Quality and Infinite archive schemes experience in the region of a 40% decline in performance from training to test; whereas the combined quality–novelty metric returned an intermediate decline in performance (in the order of 20%).

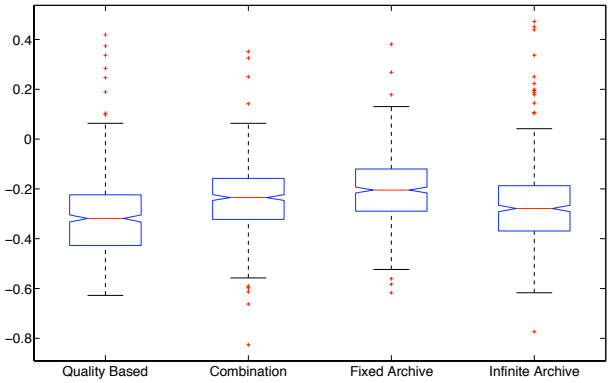

Figure 4: Plot showing first, second and third quartiles for the distribution of median generalization error. See Figure 1 caption for interpretation of the whiskers.

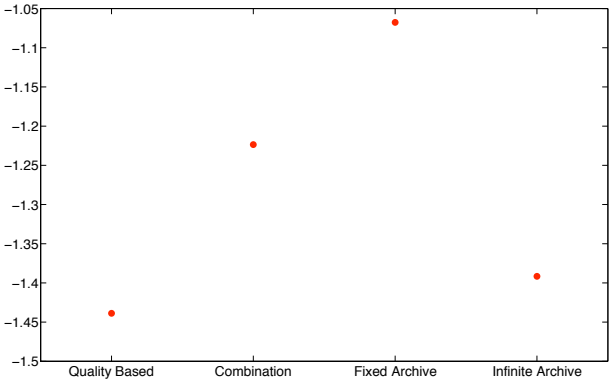

Figure 5: Plot of inter cluster distance (defined in Eq. (1)) between normalized training and test performance). Negative values express the percent by which food counts under training exceeded that under test.

5 Conclusions and Future Work

Taking as a starting point the Santa Fe trail as a benchmark with known deceptive properties [4] and introducing the test for generalization methodology of Kushchu [3], we evaluated a cross-section of novelty only, combined novelty and quality, and quality only fitness functions. Several general trends are apparent. The classical quality based performance metric provided the strongest training and test performance of all methods; thus, reinforcing the view that if a goal orientated objective can be defined for the domain in question, then assessing performance with such an objective is still very important. A simple linear combination of novelty and quality objectives provided the next best performance under training and test conditions. In the case of the novelty only fitness functions, imposition of a finite archive was generally beneficial under test conditions, but was the worst performing approach under training conditions. We suspect the poor training performance to be the result of rapidly increasing novelty requirements for entry into the archive, such that a solution which follows the trail slightly further than its parent will receive a low fitness value, i.e., on account of having eaten most of its food in the same order as its parent. The higher test performance is likely due to the same factor. Since individuals cannot gain entry into the archive for following the trail, they develop an extremely diverse set of strategies for eating the food, potentially generalizing strategies for any trail. In particular, tessellating strategies were often observed among the solutions from the finite archive method, but rarely observed among solutions from the other methods.

As highlighted above, both the fixed archive and infinite archive methods produce lower quality solutions than the traditional quality only fitness measure. In the case of training performance, this may be in part due to the deceptive nature of the problem. No method produced a true solution which managed to follow the entire trail, but the quality based solutions may not have been any closer to finding a true solution despite eating more of the food. While previous work [5, 9, 10] suggests novelty-based search may find better solutions sooner than quality based objective approaches in deceptive landscapes, our work suggests that intermediate solutions produced by novelty-based approaches may be of lower quality in terms of both training and test performance. In problems where finding a true solution is impossible or prohibitively expensive, this may be a concern. Indeed, purely novelty based schemes may encounter an overhead in the time necessary to recombine independent solutions (individuals) into a single solution that subsumes the behaviors from multiple individuals.

Future work will continue to look at the role of novelty in evolution. Earlier work using an explicitly Pareto multi-criterion objective composition of novelty and quality objectives indicates that such paradigms are effective at problem decomposition as opposed to forcing solutions to take the form of a single solution [8]. Indeed, evolution through novelty only fitness functions might support multi-level symbiotic/ teaming style models of evolution in GP. In particular, a novelty based phase of evolution might be followed/ combined with a combinatorial style search for the best combinations of solutions from the novelty based

search i.e., behaviors can exist symbiotically as independent entities within a 'host' individual at a higher level of representation. Models of this nature in which fitness is shared over a quality style objective have already appeared [6], however, doing so under purely novelty based fitness has not as yet been demonstrated. Likewise, the use of schemes such as NEAT – as was in the case in [5] and [10] – that explicitly support the identification and incorporation of traits from parent individuals into the children may provide a better basis for incorporating initially independent behaviors into a single model. Thus, frameworks such as NEAT and GP teaming – as opposed to canonical GP – might well be in a better position to make use of properties developed under novelty only style fitness functions.

Acknowledgements

J. Doucette was supported in part through an NSERC USRA scholarship and M. Heywood was supported under research grants from NSERC and MITACS.

References

[1] F. J. Gomez. Sustaining diversity using behavioral information distance. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 113–120. ACM, 2009.

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

[3] I. Kushchu. Genetic programming and evolutionary generalization. *IEEE Transactions on Evolutionary Computation*, 6(5):431–442, 2002.

[4] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, 2002.

[5] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the International Conference on Artificial Life XI*. MIT Press, 2008.

[6] P. Lichodzijewski and M. I. Heywood. Managing team-based problem solving with symbiotic bid-based Genetic Programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 363–370, 2008.

[7] Y. Liu, X. Yao, and T. Higuchi. Evolutionary Ensembles with Negative Correlation Learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380–387, 2000.

[8] A. R. McIntyre and M. I. Heywood. Cooperative problem decomposition in pareto competitive classifier models of coevolution. In *Proceedings of the European Conference on Genetic Programming*, volume 4971 of *LNCS*, pages 289–300. Springer, 2008.

[9] J.-B. Mouret and S. Doncieux. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *IEEE Congress on Evolutionary Computation*, pages 1161–1168, 2009.

[10] S. Risi, S. D. Vanderbleek, C. E. Hughes, and K. O. Stanley. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 153–160. ACM, 2009.

[11] D. Zongker and B. Punch. *lil-gp 1.0 User's Manual*. Michigan State University, 1995.