MOGE: GP Classification Problem Decomposition using Multi-objective Optimization

Andrew McIntyre
Dalhousie University
Halifax, Nova Scotia
Canada B3H 1W5
+ 1 902 494 2951
armcnty@cs.dal.ca

Malcolm Heywood
Dalhousie University
Halifax, Nova Scotia
Canada B3H 1W5
+ 1 902 494 2951
mheywood@cs.dal.ca

ABSTRACT
A novel approach to classification is proposed in which a Pareto-based ranking of individuals is used to encourage multiple individuals to participate in the solution. To do so, the classification problem is re-expressed as a cluster consistency problem, thus allowing utilization of techniques from multi-objective optimization. Such a formulation enables classification problems to be automatically decomposed and solved by several specialist classifiers rather than by a single 'super' individual. In this paper, we demonstrate the proposed approach to two benchmark binary problems and recommend a natural extension to multi-class problems. Results indicate the general appropriateness of the approach.

1. INTRODUCTION
Since the conception of Genetic Programming (GP) several methodologies have been proposed for encouraging solutions to take the form of a set of programs (individuals) solving different parts of the problem, as opposed to the population converging on a single 'super' individual. Recent examples might include the cooperative [1], and competitive [2] co-evolutionary paradigms, where both have been demonstrated within a Genetic Algorithm (GA) context. Both co-evolutionary approaches are multi-population models. In this work we are specifically interested in developing a multi-member solution from a single population, as opposed to 'super' individuals, where the latter has recently been investigated in a Multi-objective GP context [3].

Recent advances in the Multi-objective GA (MOGA) optimization literature have demonstrated that a Pareto front may be used to maintain a set of candidate solutions to multi-modal problems [4]. Within the MOGA context, candidate solutions describe a point in multi-dimensional space, where the basic objective is to locate the set of solutions (points) minimizing a predefined objective over an unknown (multi-modal) function. The specific interest of this work is to provide a methodology for utilizing the MO framework, currently demonstrated under a GA context, for solving the problem decomposition issue under Genetic Programming. To do so we make use of a recent result from GP in which the switching (or global) wrapper classically employed to map the 'raw' GP output to a discrete number of class labels is replaced with a local or Gaussian-type wrapper function [5]. Under this context, it is now possible to phrase the classification problem as finding the minimum set of mappings from a multi-dimensional input space to class consistent clusters on the one-dimensional GP output space.

In the following section, we provide the rationale for the proposed approach before detailing the multi-objective GP (MOGE) classifier itself in Section 3. Results on two real world binary classification benchmarks are presented in Section 4, with conclusions and future work presented in Section 5.

2. BACKGROUND
Binary GP classifiers have typically assumed a wrapper operator based on a switching type function which partitions the entire range of GP output values into a binary space for comparison against the class label. Such a methodology was recently questioned, and demonstrated to hide the underlying information regarding the quality of the mapping performed by GP from the original multi-dimensional input space to the one dimensional output space [5]. Specifically, basing the wrapper on a global as opposed to a local activation function ignores the degree of class separation implicit in the mapping performed by GP. This issue was addressed by expressing the objective of the classification problem as one of maximizing a cluster separation distance; where a cluster is comprised of all values associated with the same class label as mapped by GP to the one dimensional GP output space. The implicit assumption in this model however, is that a single mapping is sufficient and / or appropriate to solve the classification problem.

In this work, we build on [5] by supporting multiple mappings. Thus, we are interested in identifying the minimum set of non-overlapping clusters capable of describing the in-/out- class data. Such a methodology provides the basis for decomposing the problem into a series of smaller problems. To do so, we express cluster membership by assuming a Gaussian membership function, thus we are nominally interested in minimizing several properties, including the overlap in Gaussian membership, the number of mappings necessary to describe the data, and classification error (that is, cluster membership must be consistent across class label(s)). As such, we have a multi-objective optimization problem that we can address using the recent advances from MOGA.

3. ALGORITHM DESCRIPTION

The following study is performed using Grammatical Evolution (although the algorithm is not specific to the type of Genetic Programming employed). Grammatical Evolution (GE) permits automatic and language independent evolution of programs of arbitrary complexity [6]. There are some obvious similarities between GE and GP; however, GE does not operate directly on the programs themselves as in traditional GP; rather the programs are stored as a series of Backus-Naur form (BNF) grammar rule selectors, which are in turn indirectly represented by a fixed length binary string individual. In this sense, GE is similar to GA and consequently permits the use of simple GA search operators. Since the algorithm presented here is not specific to GE and because of the numerous similarities, the terms GE and GP are used interchangeably in this paper.

3.1 GE Search Operators

GE uses a context free grammar (CFG) to perform the mapping between genotype and phenotype. This implies that changing the value of a single gene will frequently change the phenotypic representation for all the following the genes. That is to say, the genotype is incrementally converted into a corresponding phenotype using the CFG to convert non-terminals into terminals, where there non-terminals frequently expand into other non-terminal symbols before the grammar identifies a terminal. This has resulted in the use of context based crossover operators that note the genes corresponding to a terminal in the phenotype [7]. Crossover is limited to exchanging gene sequences between terminal symbols in the phenotype, thus the remainder of the phenotype is largely preserved. Such a crossover operator was demonstrated to perform significantly better than the single point crossover typically employed by GE. In this work a similar crossover operator is employed along with a mutation operator that explicitly mutates genes corresponding to terminals in the phenotype, resulting in alternative terminals of the same arity.

3.2 MOGE Classifier Output Representation

In this work, the term 'classifier' refers to a set of GE individuals (mapped as arithmetic expressions in this case) that collectively solve a classification problem. Individual expressions here are defined by the following context-free grammar, where variables are replaced by their associated pattern features:

```
code: exp
exp : var | const | exp op var | exp op exp
op  : + | - | * | %
var : x1 | x2 | … | xi
```

Classification rules are built by first mapping the multi-dimensional input exemplars to the one dimensional output space (GP_{out}) and then re-expressing a subset in terms of a Gaussian membership function i.e., a local membership function. A decision value of more than a predefined decision threshold (0.1 in this work) indicates that the pattern is considered 'in-class' or 1; it is 'out of class' or 0 otherwise (See fig. 1).

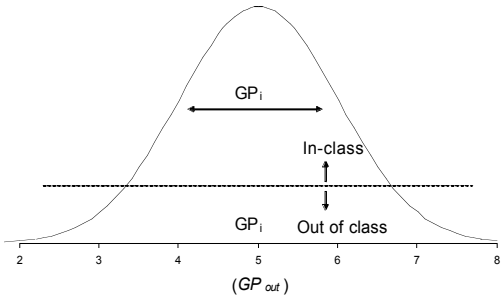

Figure 1. Sample GE decision based on raw output.

Thus in this work, an individual participating in a classifier is a data structure containing three members: GE arithmetic expression, Gaussian center (μ) and width (σ). This is a key difference between a Multi-objective GE (MOGE) classifier and the conventional approach to classification with GP, where a hard switching function at the output value of 0 is frequently employed.

3.3 MOGE Algorithm Overview

The proposed algorithm has the following basic form:

1. Initialize the GE population (POP_SIZE, CODONS, BITS_PER_CODON, MAX_EXP_LEN) with uniform probability;
2. Evaluate initial population:
    a. For each individual, transform the (multi-dimensional) input data to the associated (one-dimensional) output space, or GP_{out};
    b. Cluster each individual's points on GP_{out} to establish initial Gaussian centers (μ) and widths (σ);
    c. Calculate Pareto variables corresponding to the classification objectives;
    d. Assign Pareto ranking and initial fitness;

3. While stopping criteria is not satisfied, execute the following inner loop for $\frac{POP\_SIZE}{3}$ (≡1 EPOCH) times:
    a. Perform fitness proportionate, stochastic selection of two parents;
    b. Apply search operators (*P*(XOVER), *P*(MUTATE)) to generate two children.
    c. Evaluate children (step 2) with respect to the current population, updating population members as necessary;
    d. Children replace the two lowest ranked members of population (ties are resolved using uniform probability);
4. Select classifier participants from the members of the Pareto front based on maximum classification performance on training data.

The stopping criteria of Step 3 is defined in terms of either: successive epoch *Pareto rank histograms* match (Section 3.4.3), or maximum number of iterations are completed (MAX_EPOCHS).

3.4 Algorithm Details

3.4.1 Initialization

Initialization of an individual is performed by randomly setting each of the genes and attempting to map to a legal expression as defined by the grammar. The process is repeated until a legal mapping is successful such that no degenerate GE individuals are explicitly defined in the initial population [6]. There are no requirements on expression length; however, for memory considerations, expressions are constrained to a maximum length of MAX_EXP_LEN.

3.4.2 Individual Evaluation

Each individual is decoded into the corresponding phenotype, and evaluated over the training set, effectively mapping the original multi-dimensional input data onto points in a single dimensional (GP_{out}) space. This set of GP_{out} points are then clustered, in this case using the *Potential Function* method (section 3.4.2.1), thus associating a cluster center with the region of highest density. The cluster center now defines the individual's Gaussian center (μ) and the GP_{out} points that are clustered about this center (also defined by the Potential Function) that are used to estimate the individual's Gaussian width, σ,

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(GP_{out}(i)-\mu)^2}$$

where *i* indicates the GP_{out} points having membership to this cluster. The output from the individual is now defined by a Gaussian probability density function:

$$GPG(GP_{out}(i),\mu,\sigma) = \exp\left(-\frac{(GP_{out}(i)-\mu)^2}{2\cdot\sigma^2}\right)$$

where *GPG* is the GP's *(Gaussian) class membership function*. That is to say, at this point we have not explicitly enforced class membership, but merely constructed the membership operator on the basis of a single distribution describing the region of most density on GP_{out}. When the membership function returns a value of more than the chosen decision threshold (0.1 in this work), the point's associated pattern is considered 'in-class' (with the associated degree of membership); an exemplar is 'out of class' or has 0 membership otherwise (see fig. 1). This value is denoted the individual's *decision value* for a given point and the set of decision values for an individual (over all exemplars) is hereafter referred to as the individual's *decision vector*.

It is also through the *decision vector* that cluster class consistency is assured. In other words, the membership function is used during evaluation to establish the individual's adherence to the objectives that are defined for identifying the notion of classifier success, including: minimization of the sum squared error (SSE); minimization of overlap between patterns classified among individuals in the population; minimization of expression length; and maximization of pattern coverage. Thus the membership function provides the basis for measuring the degree to which the objectives are satisfied. Ultimately, it is the degree of success in optimizing this set of objectives that leads to the individual's Pareto ranking and therefore its overall fitness assignment.

3.4.2.1 The Potential Function

The Potential Function is a four step iterative process, which proceeds as follows [8]:

1. Identify each point's candidate potential with respect to all other points in the set using a suitable distance metric.
2. Select the point with the highest total potential.
3. Subtract the highest potential (as determined in step two) from all other points.
4. Repeat steps two and three until a terminating condition is realized.

The distance metric identified in step one is referred to as the 'Potential Function':

$$P_t(GP_{out}(j)) = \sum_{i=1}^{K}\exp\left(-\alpha\left\|GP_{out}(i)-GP_{out}(j)\right\|^2\right)$$

K is the total number of points in the set and α is the cluster radius constant (providing a means to influence granularity of clusters).

Points with the most similarity to the current point, $GP_{out}(j)$, contribute most to the corresponding potential $P_t(GP_{out}(j))$. Points with the most (or very near) neighbors will be assigned the greatest potential (as required in step two).

Step three removes the influence of the 'winning' point (that with the highest potential) from all others within the same cluster. Thus, having found a winner, each cluster member's potential is reduced by an amount proportional to its distance from the current winning potential $P_t(GP_{out}^*(j))$, or $\forall\, i \in \{1...K\}$, $P_{t+1}(GP_{out}(i))$ is defined as:

$$P_t(GP_{out}(i)) - P_t(GP_{out}^*(j)) \exp\left(-\beta \left\| GP_{out}(i) - GP_{out}^*(j) \right\|^2\right)$$

where $P_{t+1}(GP_{out}(i))$ is the updated potential at iteration $t + 1$ and β ($< \alpha$) is the radius associated with the Potential decay process. This process continues until an end condition is reached, as defined by the ratio of the potential at the current time step $P_t(GP_{out}^*(j))$, to the initial potential, $P_0(GP_{out}^*(j))$ as follows:

IF $P_t(GP_{out}^*(i)) > \gamma_{upper}\left(P_0(GP_{out}^*(j))\right)$

THEN create a new cluster;

ELSE IF $P_t(GP_{out}^*(i)) < \gamma_{lower}\left(P_0(GP_{out}^*(j))\right)$

THEN end;

ELSE ignore the point (it does not represent a significant cluster).

Where the cutoff values (γ_{lower} and γ_{upper}) were chosen experimentally.

In this work we are only interested in identifying the first cluster and associated neighborhood of points. That is to say, the performance of each individual is described in terms of exemplars mapped to a single local membership function. Since the assignment of points, $GP_{out}(i)$, to clusters is determined by the 'j' which resulted in the greatest decrease of potential during the decay steps, a complete run of the Potential Function is done per evaluation.

3.4.2.2 Classification Objectives

We have identified several objectives that need to be optimized in order to qualify the value of an individual that might be chosen to participate in the final classifier. Each metric used in the list below is referred to as a *Pareto variable*. The associated objectives are referred to hereafter as the *Pareto objectives*. The Pareto variables and objectives chosen for this work are as follows:

1. **Minimize the sum squared error (SSE)**: this objective describes classification performance by rewarding true positive classifications while discouraging the occurrence of false positives (an individual incorrectly labeling a pattern of class 0 as class 1). The opposite situation (occurrences of false negatives) is not of major concern here, since our hypothesis is that many, highly specialized individuals will naturally decompose the problem and participate in the final classifier. Here, the SSE for an individual is calculated by:

$$SSE = \sum_{i=1}^{N} \left(actual_i - GPG(GP_{out}(i), \mu, \sigma)\right)^2$$

   where i indicates raw GP output points having membership to this individual's cluster center as identified by the Potential Function; *actual* is the actual pattern label.

2. **Maximize the count of 'in-class' patterns correctly classified**: aims to include as many patterns as possible within the Gaussian mapping. This objective is expected to encourage survival of individuals that map patterns densely in GP_{out}. That is, mapping many patterns (in a class-consistent fashion) to the same region of the GP's output space is a desirable quality.

3. **Minimize expression length**: this objective, based on techniques investigated in [3], aims to impose parsimony and discourage unnecessarily lengthy solutions, which are clearly undesirable since longer expressions lead to more computational overhead and less solution transparency.

4. **Minimize pattern overlap**: aims to encourage diversity in the patterns that are being correctly classified; i.e. this objective is intended to discourage the population from repeatedly overlapping in their decision vectors (as defined in 3.4.2). In other words, it is largely unproductive to have many individuals all classifying the same training patterns. Therefore, the overlap value for an individual is simply based on a count of the number of times that each exemplar has already been correctly classified by other members of the population.

3.4.2.3 Pareto Ranking and Fitness Assignment

The Pareto variables introduced in the previous section collectively define a four-dimensional Pareto vector for each individual which is then the basis for Pareto ranking. Multi-objective optimization with Pareto ranking involves the notion of *Pareto dominance*, where Pareto vector A dominates vector B if A performs at least as least well as B over all dimensions, and better than B in one or more. Conventionally, an individual is said to be *dominated* if at least one other individual dominates it and *non-dominated* if it dominates all others. The set of non-dominated solutions is known as the *Pareto front*.

In this algorithm, ranking with ties is employed [9]. When evaluated, the rank of an individual is the number of individuals by which it is dominated plus one. All non-dominated solutions are given the same rank (=1, defining the Pareto front). In the event of identical Pareto vectors in two individuals (that is, two vectors having the same entries in each dimension within a small degree of precision), one of the ranks is randomly increased by one thus naturally removing duplicates in the set of non-dominated solutions.

For simplicity, this work assigns the fitness of an individual in proportion to its Pareto rank. The final step of identifying the classifier participants considers only members of the Pareto front.

3.4.3 Convergence Criteria

To identify a converged state among population members, we employ the convenient stopping criteria identification method of Pareto-rank histograms, introduced by Kumar and Rockett [9] as a frequency distribution of tied ranks in a population.

For this work, rank histograms are generated from the ratio of the number of individuals at each rank between the current and previous epochs, beginning after epoch $\frac{\text{MAX\_EPOCHS}}{3}$. A match between rank histograms of two successive epochs is used as an indicator that a sufficient degree of convergence has been reached. Obviously if this condition is never met, the evolution process reaches a stopping point after MAX_EPOCHS, as defined in Table 2.

3.4.4 Selecting Classifier Participants

Once the stopping criteria have been met, the final step is to choose the best set of individuals from the Pareto front in terms of overall classification on the training set. We refer to this set of individuals as the *classifier participants*, identified using the following algorithm,

1. **Select 'free' individuals**: begin with the participants defined as: all Pareto front individuals having false positives $== 0$ and true positive > 0 on the training set;
2. **Select individuals with low count of false positives**: successively add individuals from the Pareto front (ordered by false positives) to the classifier set if:
    a. The number of new training patterns correctly labeled by the individual as 'in-class' > the number new false positives generated;
3. **Repeat step 2 until**:
    a. All training patterns are correctly classified by the classifier participants or;
    b. False positives generated by a candidate participant surpass a maximum threshold (in this work, 5% of the total number of training patterns is used as a cutoff).

3.5 A Standard GE Classifier Algorithm

The standard GE classifier used in this work follows a similar algorithm (and employs identical parameters, given in Table 2) to that presented in section 3.3, with the following changes:

- A (global) switching wrapper function centered at the GP output value of 0 determines a classifier's decision value on each pattern.
- Steps 2 and 3c now evaluate individuals solely on the grounds of a hits-based metric over the entire training set; fitness is therefore assigned in proportion to the overall accuracy of the classifier under evaluation.
- Step 3d now performs each replacement by choosing 5 individuals at random and replacing the one with the lowest fitness score.
- Step 4 chooses the individual with the best training accuracy as the solution.
- An early stopping criterion is defined by a mean training accuracy of 95%.

4. RESULTS AND ANALYSIS

Classification results of the Multi-objective GE classifier (MOGE) algorithm, as described in sections 3.3-3.4, are compared against those of the standard GE classifier, as outlined in section 3.5.

50 random initializations were run for each GP experiment. Results are reported for maximum (MAX), minimum (MIN) and median (MED) values along with first and third quartiles (Q1 and Q3) to indicate the variation in results. Statistics presented in the results are defined on true positives (*TP*), true negatives (*TN*), false positives (*FP*) and false negatives (*FN*) and are summarized below in Table 1.

Table 1. Definitions and abbreviations for result statistics

Statistic	Abbreviation	Definition
Accuracy	ACC	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	SENS	$\frac{TP}{TP+FN}$
Specificity	SPEC	$\frac{TN}{TN+FP} = 1 - \text{FPR}$
False Positive Rate	FPR	$\frac{FP}{TN+FP}$

Score	SCO	$\frac{SENS+SPEC}{2}$

4.1 Parameters

Table 2 provides the parameters used all runs of MOGE and the standard GE classifier.

Table 2. GE parameters using in this work

Parameter Name	Value
POP_SIZE	300
NUMBITS	560
CODON	8
MAX_EXP_LEN	1025
MAX_EPOCHS	400
P(XOVER)	50%
P(MUTATE)	3%

4.2 Datasets and Partitioning

In the experiments that follow, two binary data sets were used to benchmark classification performance of the proposed algorithm. Partitioning of the data into training and test sets was performed by randomly assigning patterns (without replacement) to training or test such that 75% of patterns appear in training and 25% appear in test. In and out of class data are stratified in order to achieve proportional representation from each class within the two partitions. Note that the test partition represents a disjoint set ('unseen' exemplars) from the classifier's perspective.

Two widely known binary classification problems, taken from the UCI repository for Machine Learning[1] are used in the experiments reported on here:

1. **Breast (Original Database)**: This binary classification set contains 699 exemplars with 9 numeric-valued features per pattern. This is an unbalanced data set, with 244 patterns of class 1 and 455 of class 0.
2. **Liver Disorders Database**: A binary classification problem containing 345 exemplars with 6 numeric-valued features per pattern. This is an unbalanced data set, with 145 patterns of class 1 and the remaining 200 of class 0.

4.3 Experiments

All experiments were run on a commodity AMD Sempron 2800+ processor with 1GB of RAM, running the Fedora Core 2 operating system. The GE mapping code employed in these experiments is widely available on the web[2].

This paper reports on four experiments in total. Two are run with the Multi-objective GE classifier (MOGE) algorithm (one on each data set) and these experiments are directly compared against those run with the standard GE classifier.

- **Experiment 1**: Run 50 initializations of the standard GE classifier algorithm on the UCI Breast Database.
- **Experiment 2**: Run 50 initializations of the MOGE algorithm on the UCI Breast Database.
- **Experiment 3**: Run 50 initializations of the standard GE classifier algorithm on the UCI Liver Disorders Database.
- **Experiment 4**: Run 50 initializations of the MOGE algorithm on the UCI Liver Disorders Database.

In each experiment, results are collected on training and test sets for the classifier's basic performance characteristics (see Table 1). In experiments 2 and 4, statistics for the number of classifier participants, participant expression size, and percentage of patterns classified by each participant are also collected.

In experiments 1 and 3 we are evolving only a single 'super' individual, and therefore only the winning expression size is collected in addition to the classifier's basic performance characteristics.

4.4 Classifier Performance

Tables 5, 6, 7 and 8 summarize the results of training and test runs in terms of the basic performance metrics for each of experiments 1-4.

4.5 Expression Sizes

Table 3 indicates expression lengths for MOGE participants and solutions of standard GE runs. Table 4 summarizes the number of MOGE classifier participants for each problem considered. No introns are removed prior to calculating expression lengths for the MOGE or standard GE results.

---

[1] See http://www.ics.uci.edu/~mlearn/MLSummary.html

[2] See http:// www.grammatical-evolution.org

Based on the results in Table 3, it is clear that MOGE participant sizes are considerably smaller than solutions in Standard GE, and participant sizes vary slightly more than the solutions of the standard GE. This is particularly noticeable in the case of the MOGE Liver results, where there is a difference of 10 between first and third quartile expression sizes. This is thought to be due to the evolution of specific (and of varying length) participants to accurately deal with certain subsets of exemplars in the difficult Liver training set. Implicit in the notion of smaller solution sizes is lower computational overhead during expression evaluation and improved solution transparency.

4.6 MOGE Problem Decomposition

Samples of percentage of in-class patterns classified by each participant in Experiments 2 and 4 are presented in the Figures 2 and 3. It is readily apparent that there is a strong correlation between participation and associated performance of classifiers under training and test conditions. Moreover, in the case of the more difficult Liver dataset, the test data results in individuals representing 1 or 2 percent of the training data being dropped in favor of individuals expressing more of the dataset (e.g. participants 13 and 14). This resulted in the test data being classified by 9 out of the 18 classifiers originally identified during training.

From the performance tables it is apparent that MOGE does indeed successfully decompose the problem to build solutions using a set of individuals. Classification performance is similar to that returned using our baseline GE, with better results returned under the more difficult Liver dataset when using MOGE. On the easier Breast problem, it appears that the MOGE system emphasized the false positive rate as opposed to detection rate. The MOGE results on the test partition were marginally worse as compared to those of the standard GE classifier, despite MOGE's better results on the training data. Additionally, MOGE provides a measure of certainty for each classification (as opposed to a binary response alone) and decomposes the problem, yielding smaller solutions than the standard GE thus potentially reducing expression evaluation cost while permitting more insight into the solutions provided.

5. CONCLUSIONS

A methodology is proposed for integrating problem decomposition with model building under a classification context. The basis for the MOGE algorithm is a process for building local (wrapper) functions on the GP output space provided by the GP mapping from input to output spaces. Class consistency is then imposed using a multi-objective optimization setting in which a Pareto ranking of the population is enforced. Moreover, by utilizing the MO framework of Kumar and Rocket, we benefit from the availability of an early stopping criteria associated with the behavior of the Pareto front.

We demonstrate that the ensuing solutions are competitive with results from a standard GE classifier, in which solutions take the form of a single super individual. We note that the MOGE system additionally establishes improvements over the standard approach in terms of the simplicity of solutions provided while enabling automatic problem decomposition. Moreover, from the very low false positive rates returned by the MOGE method, it is apparent that the class membership of exemplars mapped to the local membership function is very consistent.

In the case of future work, the approach will be benchmarked on a wider range of datasets, in particular multi-class problems, where solutions for each class will all be returned from the same population. That is to say, generalizing to the multi-class classification problem only requires that cluster consistency be enforced over multiple labels. Thus the objectives are evaluated relative to the exemplar class most frequently appearing in the local membership function. Finally, we also anticipate reviewing the objectives used to direct the multi-objective component of the algorithm.

Table 3. MOGE vs. STD GE: Expression lengths

MOGE: Participant Sizes		
	BREAST	LIVER
Median	13	11
Q1	11	7
Q3	17	17
Min	5	3
Max	45	111

STD GE: Solution Sizes		
	BREAST	LIVER
Median	19	19
Q1	17	15
Q3	23	23
Min	9	5
Max	35	97

Table 4. Number of MOGE Classifier participants

MOGE: Classifier Participants	BREAST	LIVER
Median	26	21
Q1	24	19
Q3	29	24
Min	15	16
Max	35	28

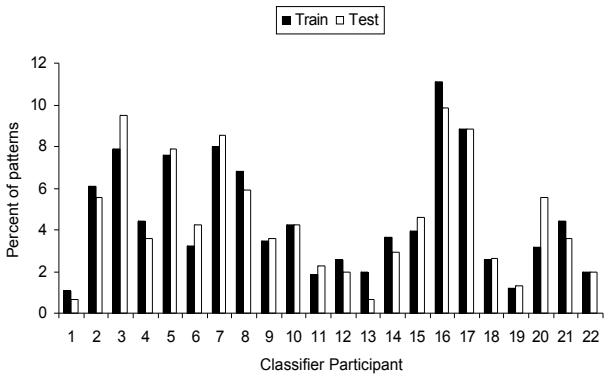

Figure 2. Percentage of Patterns Classified by each of MOGE Classifier Participants on Breast (Train and Test)

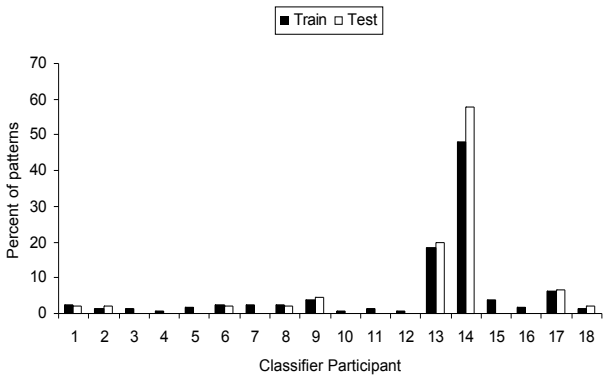

Figure 3. Percentage of Patterns Classified by each of MOGE Classifier Participants on Liver (Train and Test)

Table 5. Results of Experiment 1: Standard GE on Breast set

STD GE: TRAINING (Breast)					
	ACC	SENS	SPEC	FPR	'Score'
Median	0.956	0.967	0.953	0.047	0.960
Q1	0.947	0.961	0.938	0.039	0.952
Q3	0.964	0.972	0.961	0.063	0.965
Min	0.892	0.923	0.874	0.029	0.901
Max	0.972	0.989	0.971	0.126	0.972

STD GE: TEST (Breast)					
	ACC	SENS	SPEC	FPR	'Score'
Median	0.947	0.933	0.950	0.050	0.948
Q1	0.935	0.900	0.945	0.045	0.927
Q3	0.953	0.950	0.955	0.055	0.948
Min	0.841	0.817	0.845	0.027	0.836
Max	0.971	1.000	0.973	0.155	0.977

Table 6. Results of Experiment 2: MOGE on Breast set

MOGE: TRAINING (Breast)					
	ACC	SENS	SPEC	FPR	'Score'
Median	0.975	0.956	0.986	0.014	0.969
Q1	0.970	0.945	0.980	0.009	0.964
Q3	0.979	0.961	0.991	0.020	0.975
Min	0.951	0.934	0.951	0.000	0.951
Max	0.985	0.983	1.000	0.049	0.982

MOGE: TEST (Breast)					
	ACC	SENS	SPEC	FPR	'Score'
Median	0.929	0.883	0.955	0.045	0.917
Q1	0.919	0.850	0.945	0.036	0.906
Q3	0.935	0.900	0.964	0.055	0.928
Min	0.900	0.800	0.927	0.027	0.877
Max	0.953	0.983	0.973	0.073	0.960

Table 7. Results of Experiment 3: Standard GE on Liver set

STD GE: TRAINING (Liver)					
	ACC	SENS	SPEC	FPR	'Score'
Median	0.765	0.609	0.883	0.117	0.745
Q1	0.758	0.573	0.860	0.095	0.735
Q3	0.773	0.636	0.905	0.140	0.753
Min	0.700	0.500	0.760	0.067	0.689
Max	0.788	0.745	0.933	0.240	0.765

STD GE: TEST (Liver)					
	ACC	SENS	SPEC	FPR	'Score'
Median	0.659	0.471	0.770	0.230	0.630
Q1	0.635	0.429	0.720	0.185	0.605
Q3	0.682	0.543	0.815	0.280	0.648

Min	0.565	0.371	0.520	0.100	0.560
Max	0.718	0.686	0.900	0.480	0.693

Table 8. Results of Experiment 4: MOGE on Liver set

MOGE: TRAINING (Liver)					
	ACC	**SENS**	**SPEC**	**FPR**	**'Score'**
Median	0.798	0.700	0.867	0.133	0.783
Q1	0.778	0.655	0.843	0.113	0.764
Q3	0.821	0.755	0.887	0.157	0.806
Min	0.750	0.500	0.793	0.053	0.723
Max	0.842	0.827	0.947	0.207	0.840

MOGE: TEST (Liver)					
	ACC	**SENS**	**SPEC**	**FPR**	**'Score'**
Median	0.682	0.514	0.800	0.200	0.663
Q1	0.659	0.436	0.765	0.160	0.625
Q3	0.718	0.593	0.840	0.235	0.691
Min	0.576	0.200	0.680	0.100	0.550
Max	0.765	0.743	0.900	0.320	0.753

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of a Precarn Scholarship, NSERC and MITACS Research Grants, and a CFI New Opportunities Infrastructure Grant.

7. REFERENCES

[1] Potter M.A., and De Jong K.A. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation*, 8(1), 2000, 1-29.

[2] De Jong E. D., and Pollack J.B. Ideal Evaluation from Coevolution. *Evolutionary Computation*, 12(2), 2004 159-192.

[3] Parrott, D., Li, Xiaodong, and Ciesielski, V. Multi-objective Techniques in Genetic Programming for Evolving Classifiers. *IEEE Congress on Evolutionary Computation*, vol. 2, 2005, 1141-1148.

[4] Deb K. Multi-objective optimization using evolutionary algorithms. Chichester, UK: John Wiley and Sons, 2001.

[5] George A., Heywood M.I. GP Classification using a Cluster Separation Metric. Submitted to *Genetic and Evolutionary Computation Conference*, GECCO'06, 2006.

[6] O'Neill M., Ryan C. Grammatical Evolution. *IEEE Transactions on Evolutionary Computation.* 5(4), 2006, 349-358.

[7] Harper R., Blair A. A Structure Preserving Crossover in Grammatical Evolution. *IEEE Congress on Evolutionary Computation*, vol. 1, 2005, 2537-2544.

[8] Chiu, S. L. Fuzzy Model Identification based on Cluster Estimation. In *Journal of Intelligent and Fuzzy Systems*, vol. 2, 1994, 267-278.

[9] Kumar, R. and Rockett, P. Improved Sampling of the Pareto-front in Multiobjective Genetic Optimizations by Steady-State Evolution: A Pareto Converging Genetic Algorithm. In *Evolutionary Computation*, (10)(3), 2002, 283-314.