A New Multi-Objective Evolutionary Approach for Creating Ensemble of Classifiers

Kushan Ahmadian, Abbas Golestani, Nasser Mozayani, and Peyman Kabiri

Abstract - In recent years, an increasing amount of research has been focused on feature selection techniques. These techniques rely on an idea that by selecting the most discriminant features, it may reduce the number of features and increase the recognition. Instead of using a feature selection technique which has been widely used in multi objective evolutionary approaches for ensemble generating, this paper presents a new multi objective evolutionary algorithm based on the NSGA II which automatically preserves diversity and also covers problems with lower dimensional feature spaces in which using feature selection technique may lead to ambiguous subspaces. After creating classifiers based on the amount of error created for each class, another multi-objective genetic algorithm was used to combine them and to produce a set of powerful ensembles. Comprehensive experiments demonstrate the effectiveness of the proposed strategy.

I. Introduction

Multi objective optimization (MOO) methods that have been used to create Ensemble Of Classifiers (EoCs) are mostly based on feature selection. These methods have shown that choice of features to represent the pattern, affects several aspects of the pattern recognition problem such as accuracy, required learning time and necessary number of samples. There are also other different techniques of ensemble creation, such as Bagging [1], Boosting [2] in which different datasets are used to create different classifiers in the ensemble. Bagging and Boosting are the most popular and effective methods for creating EoCs. Re-sampling the individual training sets and using the complete set of samples for training the EoCs, is an important point that causes the diversity, which in its turn causes effectiveness of these two methods. Some other techniques are Input Decimation [3], Random Subspace [4], and Feature Selection [5].

Feature selection strategy for creating EoCs using MOO presented by Oliveira [5], which generates EoCs in the context of supervised learning where their base classifier is a neural network. Their approach operates in two different levels. In the first level classifiers have been generated by using feature selection strategy and in the second level, it searches the best possible ensemble among such classifiers. A similar attempt has been used in [6], where ensemble of k Nearest Neighborhood classifiers has been generated by applying a feature subset selection approach. In the reported work ambiguity and error rate are measures which have been used as the objectives of a multi-objective optimization method used for generating the most accurate ensemble. It is possible that samples that are distinguishable in the original feature space become ambiguous in the new feature space, specially if there is a large reduction in dimensionality or when the number of features in the original set is low. Therefore caution has to be taken in applying this method to low-dimensional data. The choice of number selected features could have a strong impact on accuracy [7]. There is also an agreement on the role of diversity in the ensembles. Deferent measures of diversity and their relationship with the ensemble accuracy has been demonstrated in [8]. Despite of this, it has been shown that there is not a clear relationship between diversity and accuracy and diversity is not a better measure than the combined error rate [9]. In this research diversity was not used as an objective of optimization for the proposed evolutionary algorithm to create EoCs, but as it has been demonstrated in section III, entropy (as a measure of diversity) has been increased during the evolution of the algorithm.

The outline of this paper is as follows: Section II explains the main concepts of multi objective optimization using genetic algorithms and NSGA II. Section III presents the proposed method and diversity preserving effects of this algorithm. Section IV covers experimental results of this algorithm using different datasets and finally section V, explains the conclusions.

II. Multi objective optimization using genetic algorithms

Due to the lack of suitable solutions, a multi objective optimization problem has been mostly cast and solved as a single objective optimization problem. Optimizing multiple objectives at the same time involves finding a set of solutions which would provide the values of all objective functions [7].

A general optimization problem of objectives can be mathematically stated as:

$$\begin{array}{lll} \text{Minimize/ Maximize} & f(x) = [\, f_i(x), i = 1,\dots, M] & \\ \text{subject to :} & g_n(x) \le 0 & n = 1,2,\dots, N \\ & h_p(x) = 0 & p = 1,2,\dots, P \end{array} \qquad (1)$$

Where $g_n(x)$ and $h_p(x)$ are the N^{th} and P^{th} equality constraints respectively and $f_i(x)$ is the i^{th} objective function. The goal in multi objective optimization is to discover the Pareto Front. In contrast to traditional single-objective optimization techniques, which are limited in their ability to solve this sort of problem, multi objective optimization algorithms such as NSGA-II [10] and SPEA2 [11] are designed to find a set of non-dominated solutions as close as possible to the true Pareto Front. They use a variety of different techniques to do so but all algorithms emphasize the gathering of a diverse collection of non-dominated individuals rather than a single outstanding solution.

Manuscript received March 8, 2007.

The authors are with the Department of Computer Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran (e-mails: ku_ahmadian@comp.iust.ac.ir, abbas_golestani@comp.iust.ac.ir, mozayani@iust.ac.ir, peyman.kabiri@iust.ac.ir).

1-4244-0991-8/07/$25.00/©2007 IEEE

NSGA II

This method uses crowding distance as an explicit diversity preserving mechanism. This method allows a global non-domination check among the offspring and parent solutions by combining them into a single intermediate population and finally the new population is filled by solutions of different non-dominated fronts of the intermediate population based on crowded operator. the crowded operator ($<_c$) compares two solutions and returns the winner of the tournament. It assumes that every solution i has two attributes:

1. A non-domination rank r_i in the population.
2. A local crowding distance (d_i) which is a measure of the search space around i which is not occupied by any other solution in the population.

Based on these attributes the crowded tournament selection operator has defined as follows:

1. If solution i has a better rank, that is $r_i < r_j$.
2. If they have the same rank but solution *i* has a better crowding distance than solution j, that is, $r_i = r_j$ and $d_i > d_j$.

Finally the crowding distance assignment procedure is defined as below:

For each objective function $m = 1,2,...,M$, the set should be sorted in worse order f_m and boundary solutions should be assigned large distances and for all other solutions:

$$d_{I_m^j} = d_{I_m^j} + \frac{f_m^{\left(I_{j+1}^m\right)} - f_m^{\left(I_{j-1}^m\right)}}{f_m^{\max} - f_m^{\min}} \tag{2}$$

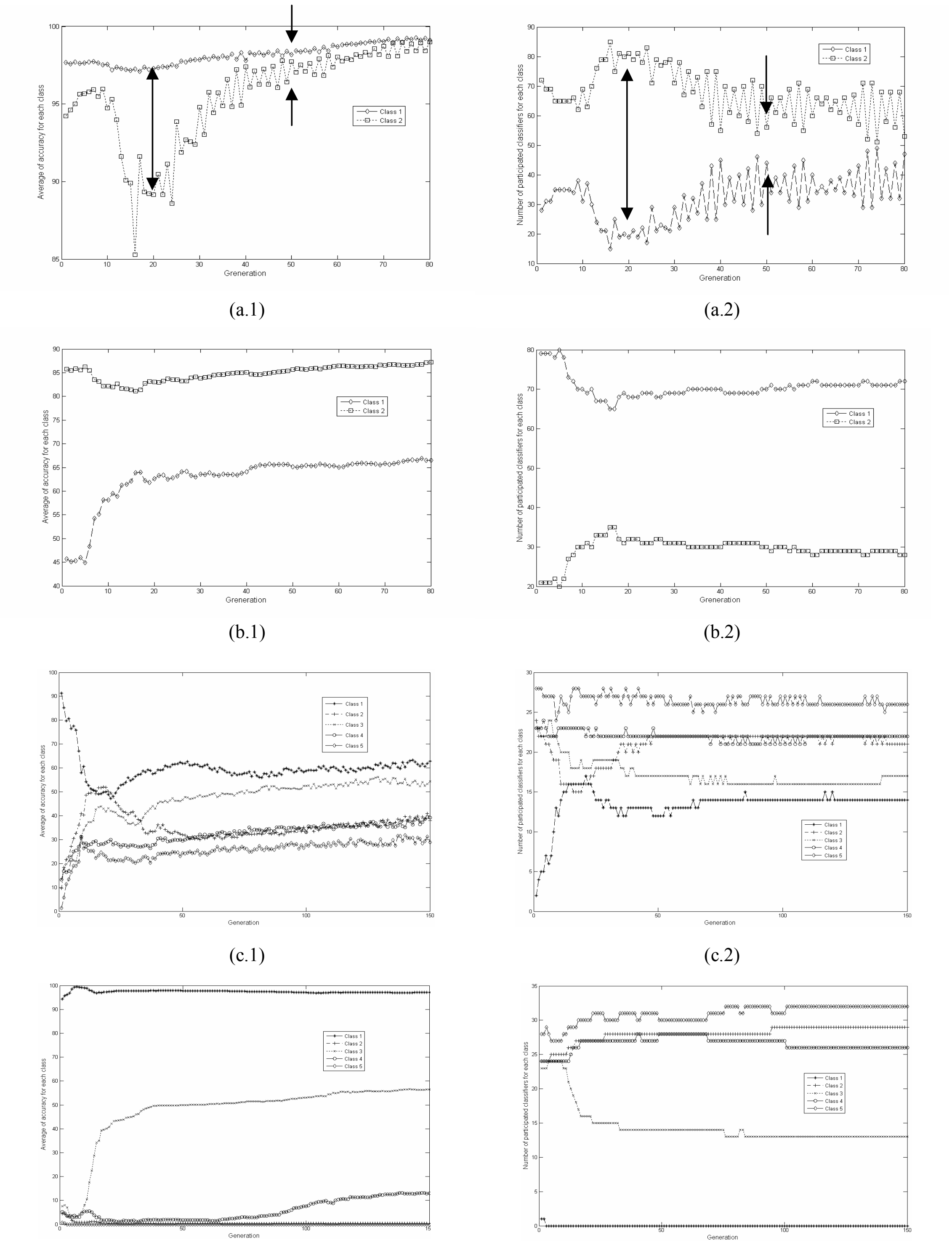
Where $j = 2$ to (number of solutions – 1).

1. Create a random population of size N, P_0; and sort the population into different non-dominated levels
2. Assign each solution a fitness (or rank) equal to its non-domination level (minimization of fitness is assumed);
3. Use the usual binary tournament selection, recombination, and mutation operators to create an offspring population Q_0 of size *N*;
4. Combine the offspring and parent population to form an extended population of size *2N*; $R_t = P_t \cup Q_t$ and sort the extended population based on non-domination;
5. Fill new population P_{t+1}. until $|P_{t+1}|+|F_i| < N$ perform $P_{t+!} = P_{t+1} \cup F_i$ and $i=i+1$.
6. Perform the crowding sort to ensure diversity if a front can only partially fill the next generation (This strategy is called "niching") and include the most widely spread $(N-|P_t+1|)$ solutions by using the crowding distance values in the sorted F_i to P_{t+1}
7. Repeat until the stopping criterion is met. The stopping criteria may be a specified number of generations.

III. Proposed Methodology

This section will explain the proposed method of approach. This approach performs in two levels, where the first level generates a set of good classifiers based on the aggregated error in each separate class and the second level will search the best ensemble among these classifiers. The first level takes into account a MOGA algorithm which is based on NSGA II algorithm and uses some aspects of bagging and boosting methods. Both stable (kNN classifiers with mutable k) and unstable (MLP) classifiers have been used in this method, which showed that choosing different sets from these two types of classifiers has no special effect on the final results, except that unstable classifiers like neural networks have more duration overload and need bigger training sets. In order to achieve a higher diversity, two strategies were considered. The first one was introduced to the algorithm as an objective which controls the size of the samples used for training and sum of deviation from mean error of classifiers based on different classes. The second strategy which is the key point of this algorithm is to choose candidate classifiers for the new generation based on the error of each class. For this reason we have made M copies of the last generation where M is the number of classes. Algorithm uses different objectives for each copy based on the accuracy of classifiers for each class and performs parallel non-dominated sorting genetic algorithm on each set. The final population is filled with the best results of each of these copies (using rank and crowding distance). The number of classifiers from each copy that contributes the new generation is proportional to the mean error made by the corresponding class in the previous generation. Higher the error rate for a class, the larger portion of that copy contributes in the new generation. This policy causes an automatic diversity preserving mechanism. Fig. 1(a and c) shows the relation between error rate in each class and the number of classifiers that are sensitive to the corresponding class in the next generation. Considering Entropy as a diversity measure, the proposed method has been compared with another algorithm that has been used for comparison of results in this paper. The logic of Entropy is based on the fact that if all of the classifiers in an ensemble recognize the samples similarly, then there would be the least measure among these classifiers and the highest diversity is manifested by $\lfloor L/2 \rfloor$ of the classifiers in an ensemble have the same output for a sample and $L - \lfloor L/2 \rfloor$ of them have the inverse output value. Let $l(\mathbf{z_j})$ to be the number of classifiers that correctly recognize a particular $\mathbf{z_j} \in \mathbf{Z}$, i.e., $l(\mathbf{z_j}) = \sum_{i=1}^{L} y_{j,i}$, here $\mathbf{Z} = \{\mathbf{z_1},...,\mathbf{z_n}\}$ is a labeled dataset and the entropy which varies between 0 and 1 and it could be calculated as:

$$E = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{L - \lfloor L/2 \rfloor} \min\{l(\mathbf{z_j}), L - l(\mathbf{z_j})\} \tag{3}$$



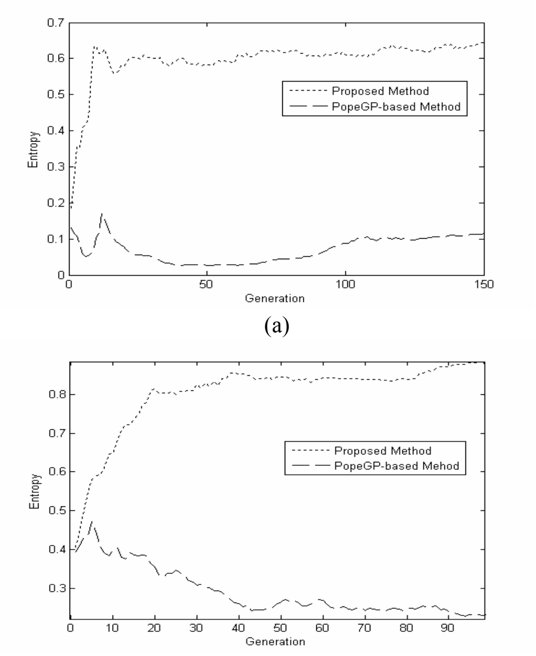
(a.1) (a.2)

(b.1) (b.2)

(c.1) (c.2)

(d.1) (d.2)

Fig. 1 Relation between accuracy of classifiers for each class and number of contributed class-sensitive-classifiers for each of these classes: (a) Wisconsin breast cancer dataset using proposed algorithm; (b) Pima Indian diabetes dataset using PopeGP-based algorithm; (c) Heart diseases dataset using proposed methodology; (d) Heart diseases dataset using PopeGP-based algorithm. Figures on the left side show the average accuracy for each of these classes in a particular generation and figures on the right side show the percent of classifiers which are more sensitive to each of these classes. Arrows in part a. show dependence between accuracy and percent of dedicated classifiers for each class.



(a)

(b)

Fig. 2 Entropy for (a) Heart disease and (b) Pima datasets

In Fig. 2 we have calculated entropy for two methods and their corresponding datasets. As it is displaying in Fig. 2 the proposed method increases the entropy without using it as an objective of the optimization algorithm.

Second level is an ordinary non-dominated sorting genetic algorithm with elitism [10] using variable length chromosomes and it is based on bit representation, one-point crossover and bit-flip mutation. For the second level the whole training data have been used to obtain an accurate ensemble and used the majority vote strategy for choosing the proper class.

Fig. 3 shows the main steps of this algorithm.

A. FIRST LEVEL'S ALGORITHM

1. Create a random population of size *N, P_0*;
2. Make $\{P_1,\ldots,P_M\}$ copies of the population where M is number of classes in the set and apply the following steps for each copy.
3. Repeat steps 2-6 of NSGA with elitism algorithm for each copy (Assign each solution in the population P_i, a fitness (or rank) with respect to classifier's accuracy for the Class C_i, ($i=\{1,2,\ldots,M\}$)).
4. after performing crowding sort on each population set, select top S_i number of members from population P_i, where S_i is defined as follow:

$$S_i^t = \frac{\sum_{j=1}^{N}\varepsilon_j^{t-1}(C_i)}{\sum_{m=1}^{M}\sum_{j=1}^{N}\varepsilon_j^{t-1}(C_i)} \times N \qquad (4)$$

where $\varepsilon_j^{t-1}(C_i)$ is the error of jth classifier from generation t-1 for class C_i.

During all selection steps, if two members have the same fitness and size, algorithm chooses member with a lower total error rate and if this criteria too be the same, the one with the lowest sum of deviation *s* from the mean of its own error.

$$s = \sum_{i=1}^{M}\left|\varepsilon^{t-1}(C_i) - \frac{\sum_{i=1}^{M}\varepsilon^{t-1}(C_i)}{M}\right| \qquad (5)$$

B. DATA

- Wisconsin Breast Cancer This dataset, consisting of 699 cases of breast cancer to be classified as benign or malignant based on nine numerical attributes. We have removed 16 cases with missing attributes We have made two versions of this data set in the first one the data was split into a training set of 477 instances (70%) and a testing set of 206 (30%) instances with proportional representation of benign and malignant classes in the two sets. In the second version data was split into a training set of 327 (48%) instances and a set of 356(52%) instances which have used for validating and testing.
- Iris: The Iris database consists of three classes with 50 instances of each. There are four numeric attributes with no missing data. Data was split 50/50 into a training set of 75 instances and a testing and validating set of 75 instances with proportional representation of each class.
- Pima Indians Diabetes: this dataset consists of 768 patterns and 8 attributes. 450 samples has been for training and the remaining for validating and testing.
- Heart Disease: the number of samples in Cleveland subset of this dataset is 303 and it also includes incomplete patterns with missing values which was not used in this paper. There are also 13 attributes and 5 classes in this dataset. All of these datasets are available from UCI Machine Learning Repository.

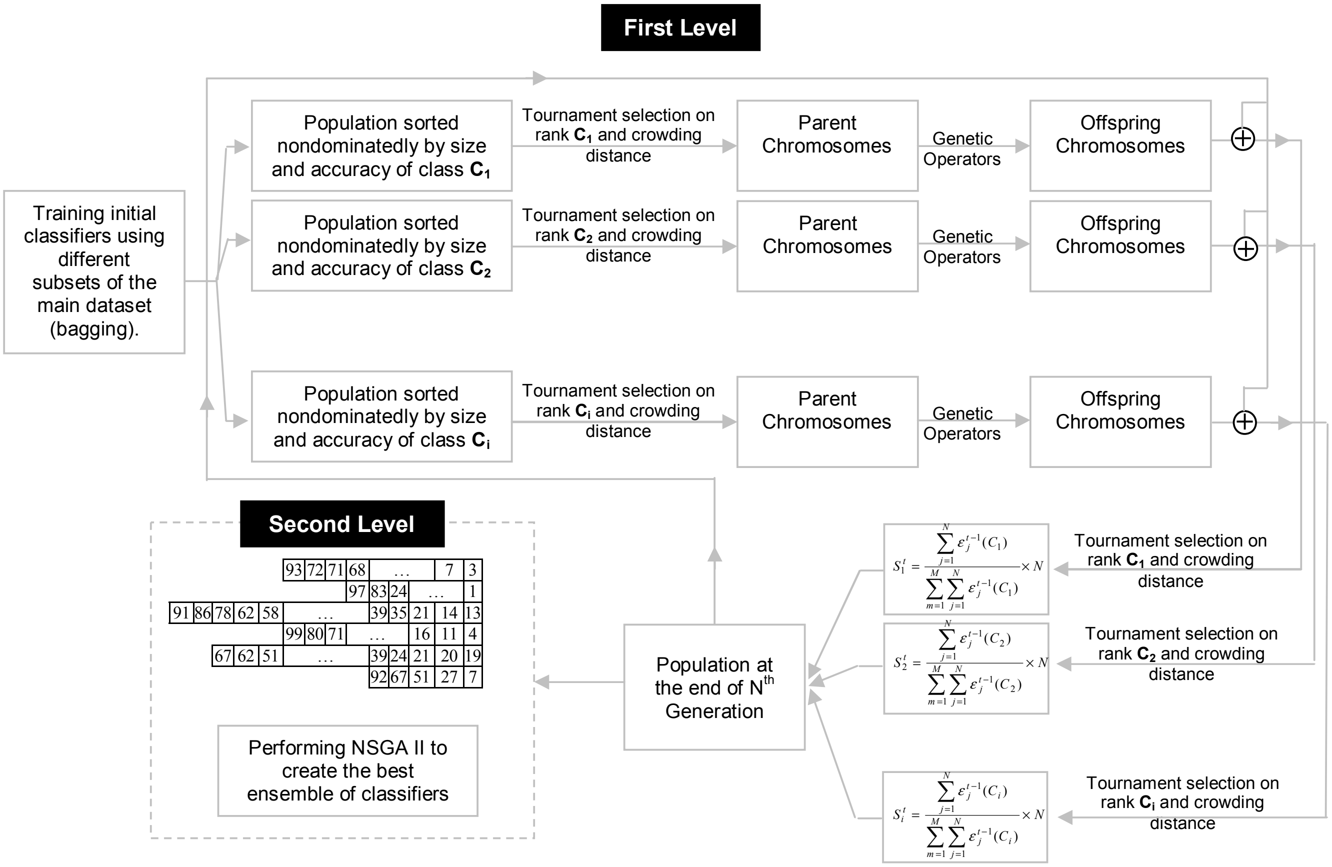

First Level
Training initial classifiers using different subsets of the main dataset (bagging).
Population sorted nondominatedly by size and accuracy of class C1
Population sorted nondominatedly by size and accuracy of class C2
Population sorted nondominatedly by size and accuracy of class Ci
Tournament selection on rank C1 and crowding distance
Tournament selection on rank C2 and crowding distance
Tournament selection on rank Ci and crowding distance
Parent Chromosomes
Genetic Operators
Offspring Chromosomes
Second Level
Performing NSGA II to create the best ensemble of classifiers
Population at the end of Nth Generation

Fig 3 Proposed Algorithm.

IV. Experiments and Discussions

In order to compare the results, another algorithm has been implemented based on POPE-GP [12] which uses kNN and MLP classifiers instead of tree-based classifiers. Experimental results are summarized in Tables 1-6. These tables show accuracy rates of the first and the second levels and demonstrate average accuracy with respect to training, validation and test data. In each table the best result for each dataset is highlighted by bold face. The following parameter settings were employed in both levels: population size=100, number of generations = 300, probability of crossover = 0.9 and probability of mutation = 0.1. The obtained number of generations is a trade off between over-trained and accurate classifiers which resulted in the most accurate ensembles in several experiments. The length of the chromosome in both levels is variable. In the first level the number of samples used to train a classifier is assumed to be the length of the chromosome and in the second level the length of each chromosome, is equal to the number of classifiers that contribute in the ensemble. It also might be considered as a fixed size chromosome where the gene *i* of the second level chromosome is represented by the classifier C_i from the first level.

In order to find the best EoCs, two objective functions has been used during this level of the algorithm, 1- maximization of the recognition rate of the ensemble and 2 - minimization of ensemble's size.

Table 1. Training results on Wisconsin breast cancer dataset (70%)

Wisconsin breast cancer (70%)				
	First level	Second level		
	Training & validation (Avg.)	Train	Test	No. of classifiers
MOGAEC	**100.00**	100	**99.51**	19
POPE-GP based	98.77	100	99.02	**17**

Table 2. Training results on Wisconsin breast cancer dataset (48%)

Wisconsin breast cancer (48%)				
	First level	Second level		
	Training & validation (Avg.)	Train	Test	No. of classifiers
MOGAEC	**98.92**	**98.12**	**99.33**	17
POPE-GP based	98.51	97.18	98.67	17

Table 3. Training results on IRIS dataset

IRIS				
	First level	Second level		
	Training & validation (Avg.)	Train	Test	No. of classifiers
MOGAEC	100	**99.04**	**98.67**	14
POPE-GP based	100	98.09	97.33	**12**

Table 4. Training results on Tic-tac-toe dataset

Tic-tac-toe				
	First level	Second level		
	Training & validation (Avg.)	Train	Test	No. of classifiers
MOGAEC	100	100	100	18
POPE-GP based	100	100	100	**12**

Table 5. Training results on Heart disease dataset

Heart disease				
	First level	Second level		
	Training & validation (Avg.)	Train	Test	No. of classifiers
MOGAEC	**74.33**	**83.15**	**78.26**	17
POPE-GP based	62.65	77.95	68.83	**12**

Table 6. Training results on Pima Indians Diabetes dataset

Pima Indians Diabetes				
	First level	Second level		
	Training & validation (Avg.)	Train	Test	No. of classifiers
MOGAEC	**83.79**	89.17	**83**	17
POPE-GP based	77.45	**92.33**	81	**12**

For the instant in the iris dataset, both algorithms has made classifiers with zero error rate during training phase, after applying the second level and creating ensemble the proposed algorithm (MOGAEC; MOGA for Ensemble of Classifiers) has achieved better results using both validation and testing sets while POPE-GP based algorithm has a smaller ensemble size.

As in the first level, the second level also generates a set of possible solutions which are trade-offs between size and accuracy of classifiers. In this article the ensemble which has a better recognition rate has been chosen, and between ensembles which have the same recognition rate, the one with a smaller ensemble size has selected. By comparing results of table 1 we see that in most cases MOGAEC has a better performance against POPE-GP based, but POPE-GP based algorithm makes less computational effort since in each generation, each instant is classifiered only once, but in the proposed approach there are copies of the base population in each generation.

V. Conclusion

In this paper, a new methodology for creating EoCs based on the accuracy of each class in the population, has been proposed. The error-proportionate-selection for each class yields a set of diverse classifiers while the second level aggregates the results and using the majority vote procedure makes the best possible choice between classes. This algorithm preserves diversity automatically without using diversity measures as the optimizing objectives. Experiments proved the efficiency and validity of proposed strategy that is capable of perform over datasets with small feature space without fear of creating ambiguous subspaces. For future work studying the effect of including different measures of diversity in the second step of the algorithm as the optimization objectives will be extended and also a new chaotic operator will be introduced to the new generation of ensemble-creation algorithms.

References

[1] L. Breiman, “Stacked regressions, ” *Machine Learning*, 24(1):49–64, 1996.

[2] Y. Freund and R. Schapire, “Experiments with a new boosting algorithm, ” In *Proc. of 13th International Conference on Machine Learning*, pages 148–156, 1996.

[3] N. C. Oza and K. Tumer, “Input decimation ensembles: Decorrelation through dimensionality reduction, ” In *Proc. Of the 2nd International Workshop on Multiple Classifier Systems*, pages 238–247, Cambridge, UK, 2001.

[4] T. K. Ho, “The random subspace method for constructing decision forests, ” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[5] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, “Feature selection for A hierarchical multi objective genetic algorithm approach, ” In *Proceedings of the 7th ICDAR*, pages 676–680, 2003.

[6] G. Zenobi and P. Cunningham, “Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error,” *Proc. European Conference on Machine Learning*, pp. 576–587, 2001.

[7] C. A. Coello and A. D. Christiansen, “Multi objective optimization of trusses using genetic algorithms, ” *Comput. and Struct.*, vol. 75, no. 6, pp. 647–660, 2000.

[8] L.I. Kuncheva and C.J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” Machine Learning, vol. 51, pp. 181–207, 2002.

[9] D. Ruta and B. Gabrys, “Classifier Selection for Majority Voting,” Information Fusion, vol. 6, pp. 163–168, 2005.

[10] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, “A Fast Elitist Nondominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA- II, ” *Proceedings of the Parallel Problem Solving from Nature VI* (PPSN-VI), pp. 849-858. 2000.

[11] E. Zitzler, M. Laumanns and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm, ” *Technical Report* 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[12] Y. Bernstein, X. Li, V. Ciesielski, and A. Song, “Multi-objective Parsimony Enforcement for Superior Generalization Performance, ” *In Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, IEEE Press, pp 83-89, 2004.