Speciated GA for Optimal Ensemble Classifiers in DNA Microarray Classification

Sung-Bae Cho
Dept. of Computer Science
Yonsei University
Seoul, Korea
Email: sbcho@cs.yonsei.ac.kr

Chanho Park
Dept. of Computer Science
Yonsei University
Seoul, Korea
Email: cpark@sclab.yonsei.ac.kr

Abstract- **With a development of microarray technology, the classification of microarray data has arisen as an important topic over the past decade. From various feature selection methods and classifiers, it is very hard to find a perfect method to classify microarray data due to the incompleteness of algorithms, the defects of data, etc. This paper proposes a sophisticated ensemble of such features and classifiers to obtain high classification performance. Speciated genetic algorithm has been exploited to get the diverse ensembles of features and classifiers in a reasonable time. Experimental results with two well-known datasets indicate that the proposed method finds many good ensembles that are superior to other individual classifiers.**

I. Introduction

Cancer has been realized as a disaster for a long while, but radical advancement on modern medical sciences gives a chance to be healed if it can be detected in its early stage. Fortunately DNA microarray technology opens a new possibility to detect it early, and it has been studied seriously in recent days [1-3].

DNA microarrays provide the measurement of expression levels of thousands of genes simultaneously. These measured data consist of either monitoring each gene in multiple times or single time point in different states (disease or tumor type) [1, 2]. An important goal of analyzing them is to identify functionally related genes or to classify samples using informative genes [4, 5]. However, there is no perfect method to classify the dataset due to the limitation of algorithms and the defects of data.

The ensemble classifier, combination of several feature-classifier pairs, has been regarded as promising because we can search a much wider solution space than only using an individual feature-classifier [3, 6]. However, because not all the ensembles yield good classification performance, we need a clever way to find good ensembles to classify samples accurately [7]. A straightforward method to find optimal ensemble is to compare all the ensembles and select the best one, but there are too many possible numbers of the ensembles. In our previous study, we proposed a method based on GA to search the ensembles effectively [8].

The GA is quite effective to search and optimize a unimodal problem, but it has a shortcoming to get stuck in local optima [9]. This means that GA does not guarantee the optimality of the solutions, especially in multimodal problems. The bioinformatics problem at hand has an inherent problem of small number of samples with large dimensionality. In this paper, we propose a sophisticated method based on speciated GA to search diverse ensembles to classify DNA gene expression profiles more accurately than the conventional GA.

II. Backgrounds

A. DNA Microarray

DNA microarray consists of a large number of DNA molecules spotted in a systemic order on a solid substrate. Especially, depending on the size of each DNA spot on the array, DNA microarray means that the diameter of DNA spot is less than 250 microns. The arrays with the small solid substrate are also referred to as DNA chips[10].

DNA microarrays are composed of thousands of individual DNA sequences printed in a high density array on a glass microscope slide using a robotic arrayer. The relative abundance of these spotted DNA sequences in two DNA or RNA samples may be assessed by monitoring the differential hybridization of the two samples to the sequences on the array. For mRNA samples, the two samples are reverse-transcribed into cDNA, labeled using different fluorescent dyes mixed (red-fluorescent dye Cy5 and green-fluorescent dye Cy3). After the hybridization of these samples with the arrayed DNA probes, the slides are imaged using a scanner that makes fluorescence measurements for each dye. The log ratio between the two intensities of each dye is used as the gene expression data [11].

$$gene\_expression = \log_2 \frac{\text{Int(Cy5)}}{\text{Int(Cy3)}}$$

where Int(Cy5) and Int(Cy3) are the intensities of red and green colors. Since more than thousands of genes are put on the DNA microarray, it is so helpful that we can investigate the genome-wide information in short time. Fig. 1 shows the process of microarray data acquisition.

B. Genetic Algorithm

Genetic algorithms are stochastic search methods that have been successfully applied in many search, optimization, and machine learning problems [12]. GAs maintain a population of encoded candidate solutions that are competitively manipulated by applying some variation operators to find a global optimum. A population consists of many chromosomes that correspond to a candidate solution, which is composed of bit strings that represent a specific status or value.

0-7803-8515-2/04/$20.00 ©2004 IEEE

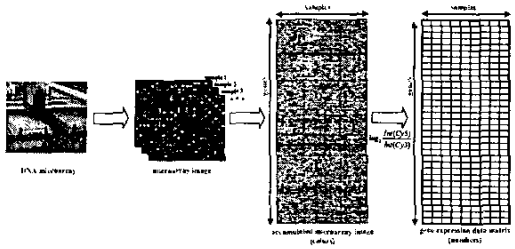


Figure 1. Process of microarray data acquisition. The images are obtained from microarray and then transformed to gene expression data matrix.

A GA proceeds in an iterative manner by generating new populations of strings from old ones. Every string is the encoded version of a candidate solution. An evaluation function assigns a fitness measure to every string indicating its fitness to the problem. The standard GA applies genetic operators such as selection, crossover, and mutation to an initially random population in order to compute the next population of new strings.

III. Optimal Ensemble Classifiers by Speciated GA

The proposed method of searching optimal ensemble based on GA is described in Fig. 2. At first, gene expression is measured as ratio of Cy5/Cy3 intensity. After log-transforming them they are normalized as value between 0 and 1. The preprocessed data are divided into training and test data sets. Some training dataset is used for training classifiers and the other validation dataset for finding the optimal ensemble. In the part of gene selection, informative genes are selected by m (in our case m is 8) different feature selection methods, and they are entered to the input of n (in our case n is 6) different classifiers. The individual feature-classifier pairs are exploited to compose an ensemble, and the good ensembles are search by speciated GA with validation dataset. The performance of the best ensemble found is verified with another test dataset. A leave-one-out cross validation (LOOCV) is performed to measure the general performance of the proposed method.

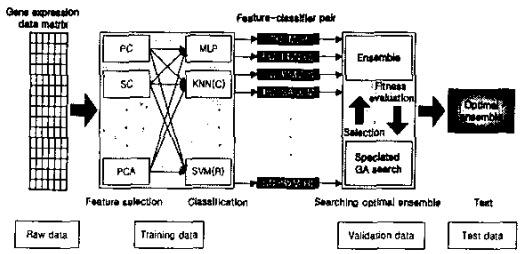


Figure 2. Flow of the proposed method

A. Feature Selection Methods

Since microarray data consist of large number of genes in small samples and relatively small number of genes are relevant to the target function, we need to select useful information from the data for proper classification. In the case of selecting only informative genes, this process is referred to as gene selection [4] and generally called as feature selection or variable selection in the areas of pattern recognition and data mining. In gene selection, informative genes highly correlated with classification can be selected using statistical correlation analysis, and clustering. Several method based on principal component analysis (PCA) or GA have been also proposed [5, 13]. Because all the previous methods have their own characteristics, we have implemented eight representative methods according to the four approaches: correlation coefficients, distance measure, information theoretic and PCA. We discussed most of them in our previous work, so only PCA, the additional feature selection method is explained.

1) Principal component analysis (PCA): PCA is a widely used statistical data analysis technique that allows reducing the dimensionality of the variable while preserving information on the interaction among original variables without much loss of information [14]. PCA is a powerful method for analyzing high dimensional data that transforms the original variables into a set of linear combinations, the principal components, which hold the data variability. The principal components are linearly independent and weighted in decreasing order of variance coverage.

To calculate the principal components, we have to get the microarray data in matrix form and subtract the mean matrix from it. With the mean adjusted matrix, the covariance matrix is calculated. After that we have to calculate the eigenvectors and eigenvalues of the covariance matrix. The components are selected by decreasing order of their own eigenvalues, forming feature vector. The final scaled data t_{ij} are derived from the feature vector and mean adjusted matrix.

$$t_{ij} = \sum_{k=1}^{n} p_{ik} m_{kj}$$

where n is the number of significant principal components, p_{ik} the score of sample i on component k and m_{kj} the loading on component k of variable j. Detailed aspects of PC calculation will be omitted because several texts and papers address the topic in detail [14].

B. Classifiers

With the selected features from previous step, the classifiers operate to decide proper determination. Classification can be defined as the process to approximate I/O mapping from the given observation to

the optimal solution. From pattern recognition or machine learning field various classification methods are developed and applied to have high prediction power. Classifiers can yield different classification results because they have their own characteristics, so we have utilized six methods among them [8]. These methods are well described in our previous study. Table I shows the list of feature selection methods and classifiers we used for this paper.

TABLE I. FEATURE SELECTION METHODS AND CLASSIFIERS

Feature selection methods	Classifiers
PC: Pearson correlation coefficients	**MLP**: Multilayer perceptron
SC: Spearman correlation coefficients	**KNN(C)**: K nearset neighbor with CC measure
CC: Cosine coefficients	**KNN(P)**: K nearset neighbor with PC measure
ED: Euclidean distance	**SVM(L)**: SVM with linear kernel function
IG: Information gain	**SVM(R)**: SVM with RBF kernel function
MI: Mutual Information	**SASOM**: Structure adaptive self organizing map
SN: Signal to noise ratio	
PCA: Principal component analysis	

C. Speciated GA

Speciation is a technique to generate multiple species within a population in evolutionary computation [9, 15]. Some restrict an individual to mate only with similar ones, while others manipulate the fitness values of species as pressure to control the probability of selection. Especially the latter ones are naturally based on niching method. In this paper, we use explicit fitness sharing and deterministic crowding methods that were introduced as a niching method by Goldberg and Richardson in 1987 [12].

Explicit fitness sharing has a fitness scaling mechanism which modifies fitness evaluation process of GA. The main idea of explicit fitness sharing is that similar individuals (species) share fitness (resources) and the number of those that can reside in any one region of the fitness landscape gets limited as shown in Fig. 3. As the result, diverse species can survive fairly as the evolution advances.

Explicit fitness sharing modifies the search landscape by reducing the fitness in densely populated regions. It lowers the individuals' fitness to the amount which is nearly equal to the number of similar individuals in the population. Typically the shared fitness sf_i of an individual i with fitness f_i is

$$sf_i = \frac{f_i}{m_i}$$

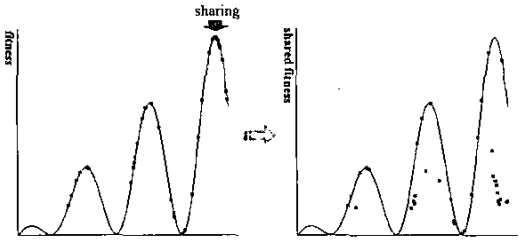


Figure 3. An example of sharing in fitness landscape.

where m_i is the niche count which measures the approximated number of individuals, with which the fitness f_i is shared. The niche count is calculated by summing sharing function values over all individuals of the population

$$m_i = \sum_{j=1}^{N} sh(d_{ij})$$

where N denotes the population size and d_{ij} represents the distance between the individuals i and j. Hence, the sharing function (sh) measures the similarity among the individuals of population. It returns '1' when the members are regarded as identical while '0' when the similarity between them is over a threshold of dissimilarity. A value between 0 and 1 is returned according to the degree of dissimilarity between them. A common sharing function is given as follows:

$$sh(d_{ij}) = \begin{cases} 1 - (\frac{d_{ij}}{\sigma_s})^{\alpha}, & \text{for } 0 \le d_{ij} < \sigma_s \\ 0 \quad , & \text{for } d_{ij} \ge \sigma_s \end{cases}$$

where σ_s denotes the threshold of dissimilarity (sharing radius) while α is a constant to regulate the shape of sharing function. α is usually set to one with the resulting sharing function referred to as triangular sharing function [12].

The distance d_{ij} between two individuals i and j is calculated by a similarity metric based on either genotypic or phenotypic similarity. Genotypic similarity is obtained with Hamming distance based on bit-string representation which means the number of non-matched bits between two strings. Phenotypic similarity is directly calculated from real parameters of the search space, such as Euclidian distance between instances [9].

Deterministic crowding is the strategy which maintains both high fitness and diversity of chromosomes. This method, unlike fitness sharing method, does not allocate elements proportional to peak fitness. Instead, the number of individuals congregating about a peak is largely determined by the size of that peak's basin of attraction under crossover. Deterministic crowding works according to the pseudocode in Fig 4.

```
Input: g - number of generations to run,
       s - population size
Output: P(g) - the final population
P(0) ← initialize()
for t ← 1 to g do
  P(t) ← shuffle(P(t-1))
  for i ← 0 to s/2 -1 do
    p1 ← a2i+1(t)
    p2 ← a2i+2(t)
    {c1, c2} ← recombination(p1, p2)
    c1' ← mutate(c1)
    c2' ← mutate(c2)
    if[d(p1,c1')+d(p2,c2')] < [d(p1,c2')+d(p2,c1')] then
      if F(c1') > F(p1) then a2i+1(t) ← c1' fi
      if F(c2') > F(p2) then a2i+2(t) ← c2' fi
    else
      if F(c2') > F(p1) then a2i+1(t) ← c2' fi
      if F(c1') > F(p2) then a2i+1(t) ← c1' fi
    fi
  od
od
```

Figure 4. Pseudocode for deterministic crowding

Finally, speciation helps the GA to obtain diverse optimal solutions. These diverse optimal solutions are useful to increase the stability of the method, especially in the case of huge-scale feature selection which can be easily unstable and converge to a local optimum.

D. Speciated GA for Optimal Ensemble

We can compose many feature-classifier pairs through the combination of features and classifiers. However, none of them is perfect because of the incompleteness of each method or the faultiness of the data. Moreover, they yield different classification results, sometimes work well and sometimes not, according to the environments they are embedded. To solve these problems, the ensemble method is attempted and studied [7, 8]. An ensemble consists of a set of individually trained classifiers whose predictions are combined when classifying novel samples [16].

$$O_e = F(fs_1, fs_2, ..., fs_n)$$

where O_e is the output of an ensemble fs_i is the output of i-th feature-classifier and F is the ensemble function.

It is reported that the ensemble is generally more accurate than any of the individual classifiers making up the ensemble. Both theoretical and empirical research have demonstrated that for the current integrating method of the ensemble approach, a good ensemble includes base classifiers that are accuracy and make their errors in different parts of the problem domain [16-18]. Therefore it is important to find such feature-classifier sets to compose a good ensemble. We have 48 feature-classifiers from 8 feature selection methods and 6 classifiers, yielding about 2.5×10^{11} different combinations of the ensemble classifier. The optimal ensemble can be searched by computing all the combinations and compare them, but it takes too long to try all the possible ensembles. Hence, efficient method to find optimal ensemble is needed and this paper exploits the speciated GA to work out this problem.

The GA with speciation can be applied effectively to solve combinatorial optimization problems. The structure of chromosome to find optimal ensemble is as shown in Fig. 5. One chromosome can be regarded as one ensemble and each chromosome is composed of a bit string of length 48, each of which indicates whether the corresponding feature-classifier pair is included to the ensemble or not. Each bit of corresponds to a specific feature-classifier pair, say the first bit to CC-MLP, the second bit to ED-MLP, and so on. Fig. 5 is an example of an ensemble of the second (ED-MLP), third (IG-MLP), and sixth (PCA-MLP) feature-classifier pairs. We can see that this is an example of good ensemble. Each participant classifier for the ensemble ranges from 62.5% to 75% of accuracy, leading to 87.5% of ensemble accuracy. The fitness is evaluated as follow:

$$fitness = \frac{\#\text{ of corelectly classified samples}}{\#\text{ of total validation samples}}$$

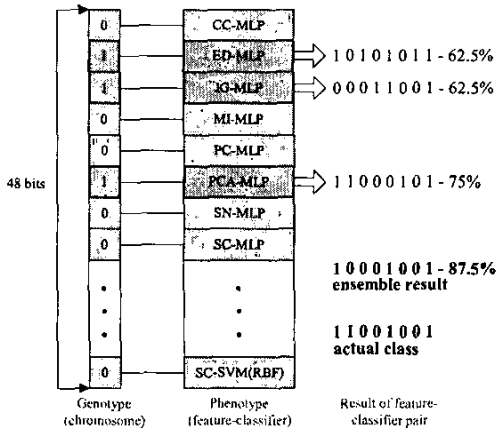


Figure 5. An example of chromosome structure

The optimal ensemble can be found using standard GA but speciation is applied to find solutions as wide as possible and analyze them.

The procedure of fitness sharing for the optimal ensemble is like this: At first, chromosomes are selected randomly as many as the population size. Each of them is evaluated with fitness function by the fitness function (step 1). The fitness is re-calculated by fitness sharing and then chromosomes go to mating pool for genetic operations (step 2). Good chromosomes have higher

chance to be selected than poor ones. In the mating pool, pairs of chromosomes are selected and then the information of chromosome for each part is exchanged by crossover and some bits are mutated according to the mutation probability (step 3). Through the genetic operations, the chromosomes evolve to the optimal ensembles. At this time, chromosomes have new body and they are evaluated with their new body (step 4). Steps 2 to 4 are repeated until the solution is satisfied with the given condition.

On the other hand, the procedure of deterministic crowding is like this: At first, chromosomes are selected randomly as many as the population size (step 1). They are shuffled and adjacent two chromosomes compose a pair (step 2). All pairs produce their own two children by genetic operations and two parent-child pairs are composed as shown in Fig. 4 (step 3). After that, a chromosome whose fitness is higher than the other is selected per pair (step 4). Steps 2 to 4 are also repeated until it satisfies some condition.

On the other hand, among various methods to combine feature-classifier pairs, we have chosen majority voting and weighted voting methods as described in Table II. In the weighted voting, the accuracy of corresponding single feature-classifier pair for validation dataset is used for the weight.

TABLE II. ENSEMBLE METHOD ($C_{1i}(x) = 1$ IF $e_i(x) = 1$, OTHERWISE $C_{1i}(x) = 0$; $C_{0i}(x) = 1$ IF $e_i(x) = 0$, OTHERWISE $C_{0i}(x) = 0$; w_i IS THE ACCURACY OF $e_i(x)$ THAT IS THE OUTPUT OF INDIVIDUAL FEATURE-CLASSIFIER.)

Combining method	Output	Condition
Majority voting	1	$\sum_i c_{1i}(x) > \sum_i c_{0i}(x)$
	0	otherwise
Weighted voting	1	$\sum_i w_i c_{1i}(x) > \sum_i w_i c_{0i}(x)$
	0	otherwise

IV. EXPERIMENTAL RESULTS

A. Experimental Datasets and Environments

We have used two well-known gene expression profiles which are available from the internet: lymphoma (http://genome-www.stanford.edu.lymphoma) [19] and colon (http://www.sph.uth.tmc.edu:8052/hgc/default.asp) [2] datasets. These are well explained in our previous work [8].

The experiments consist of gene selection, classification, searching for the optimal ensemble using speciated GA and validation. For gene selection, the genes are ranked according to their feature scores, and 25 high score genes are selected for classification, since a preliminary study suggested the optimal number of genes as 25~30 [20]. For classification, we have used a two-layered MLP with 8 hidden nodes, 2 output nodes, 0.01~0.50 of learning rate, 0.3~0.9 of momentum, 500 of maximum iterations. The back-propagation algorithm stops the training when it reaches to 98% of training accuracy. In the case of KNN, we have set k from 3 to 9, and used the Pearson correlation coefficients (KNN(P)) and the cosine coefficients (KNN(C)) for the similarity measures. We have used SVM's with linear (SVM(L)) or RBF (SVM(R)) kernel function. In SASOM, we have used initial 4×4 map which has rectangular shape.

For speciated GA, we have used roulette wheel and rank-based methods for selection operator. In the rank-based selection, we give higher rank to the chromosomes whose number of 1s is smaller than the others, for tie-break. Because preliminary results showed that it sometimes converged to a local minimum when the GA was run with less than 100 chromosomes, we set the chromosome size as greater than 100. We have conducted experiments with different crossover rates of 0.3, 0.5, 0.7 and 0.9, and mutation rates of 0.01 and 0.05. In the mutation, to find the optimal ensemble fast, we set the ratio 0 to 1 is the half of that of 1 to 0. GA stops when it finds the perfect ensemble on the validation dataset, or when the generation exceeds 10,000. In the fitness sharing, we have applied sharing to genotype, calculated Hamming distance, and set the sharing radius as 5 and alpha as 2.

We have separated lymphoma cancer dataset to 22 training samples, 24 validation samples and 1 test sample for leave-one-out cross validation. Similarly, we divided colon cancer dataset to 31 training samples, 30 validation samples and 1 test sample.

B. Individual Feature-Classifiers

Table III and IV show the average accuracies (the percentages of correct classification) of all feature-classifiers for test samples of lymphoma and colon cancer dataset, respectively. The experimental results show that PCA and IG yield good performance in gene selection and MLP, KNN(C) and KNN(P) produce superior performance in lymphoma classification. The average accuracy of all feature-classifiers is 73.5%. In colon cancer dataset, the results of individual feature-classifier are not largely different from lymphoma cancer dataset. In this dataset, PCA is superior to other feature selection methods, and MLP and KNN are competitive classifiers. The average accuracy is 72.4%.

C. Optimal Ensembles

On the validation dataset, the GA practically found the perfect ensembles. On average, the ensembles searched by GA are always superior to individual feature-classifier pairs. Table V is two examples of the optimal ensembles on lymphoma dataset. Although the participant feature-classifiers are not all good ones, through the

complementary combination they could compose the optimal ensemble.

TABLE III. ACCURACY OF INDIVIDUAL FEATURE-CLASSIFIER PAIRS ON LYMPHOMA DATASET

	MLP	SASOM	SVM(L)	SVM(R)	KNN(C)	KNN(P)	Avg
PC	77.6	67.6	66.4	55.6	78.4	78.0	70.6
SC	78.8	67.2	68.0	57.6	78.4	76.8	71.1
ED	75.2	62.8	66.4	64.0	76.0	77.6	70.3
CC	80.0	64.4	72.4	56.4	78.0	78.4	71.6
IG	85.2	75.2	77.6	66.8	81.6	83.2	78.3
MI	80.0	67.6	67.2	58.4	76.4	77.2	71.2
SN	81.2	70.8	68.0	58.4	78.8	79.2	72.7
PCA	87.2	84.0	**88.4**	58.4	86.0	86.4	81.7
Avg	80.7	70.0	71.8	59.5	79.2	79.7	73.5

TABLE IV. ACCURACY OF INDIVIDUAL FEATURE-CLASSIFIER PAIRS ON COLON DATASET

	MLP	SASOM	SVM(L)	SVM(R)	KNN(C)	KNN(P)	Avg
PC	79.4	71.0	67.1	67.1	72.9	78.7	72.7
SC	77.4	61.3	67.1	67.1	69.7	69.7	68.7
ED	77.4	67.7	67.7	67.7	76.8	77.4	72.5
CC	80.6	72.9	67.1	67.1	77.4	79.3	74.1
IG	78.7	65.2	67.7	69.7	73.5	74.8	71.6
MI	78.1	65.8	69.0	65.8	71.0	74.8	70.7
SN	74.8	60.0	67.1	67.1	75.5	73.6	69.7
PCA	87.1	74.2	**87.7**	67.1	80.0	78.1	79.0
Avg	79.2	67.2	70.1	67.3	74.6	75.8	72.4

TABLE V. TWO CASES OF THE OPTIMAL ENSEMBLES ON LYMPHOMA DATASET

Methods	Feature-classifier	Accuracy (%)
Majority voting	CC-KNN(P)	75.0
	MI-KNN(C)	83.3
	SN-KNN(C)	79.2
	SC-SASOM	62.5
	IG-SVM(L)	91.7
	ensemble	100.0
Weighted voting	IG-KNN(C)	91.7
	MI-KNN(C)	83.3
	SN-KNN(C)	79.2
	SN-KNN(P)	79.2
	CC-SASOM	54.2
	IG-SASOM	83.3
	PC-SVM(R)	62.5
	ensemble	100.0

If optimal ensemble can be found through the ensemble of good feature-classifier pairs or just many random ensembles, our proposed method become useless. Fig. 6 shows the comparison of accuracy for some method. We can see that the proposed method is always equals to or superior to other method though there are some case that could not find the perfect ensemble.

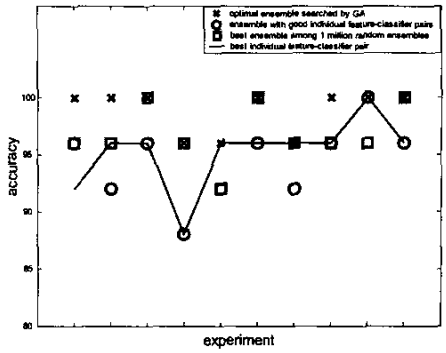


Figure 6. Comparison of accuracy. Optimal ensemble searched by GA always yields the performance superior to other methods.

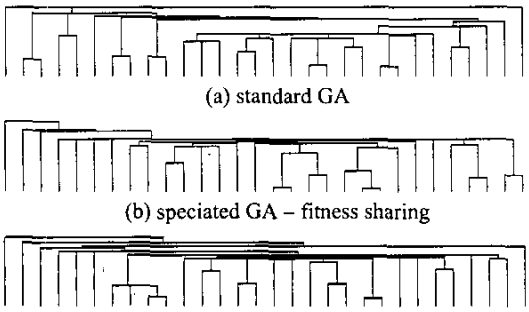
In the LOOCV, the proposed method yields a performance superior to individual feature-classifier pairs. It right classifies 44 of 47 samples on lymphoma dataset and 56 of 62 samples on colon dataset, leading to 93.6% and 90.3% accuracy respectively.

For the deep and wide analysis, many optimal ensembles are needed so we applied speciation to GA. Table VI shows the number of optimal ensembles found by standard GA and two speciation methods with same condition (500 iterations, 500 population size, roulette wheel selection method, majority voting) and in different data separation for lymphoma dataset. We can see that deterministic crowding find many optimal ensembles in comparison to a standard GA or fitness sharing. The number of optimal ensembles found by fitness sharing is similar to when standard GA is used. In this table, each column means different experiments. The speciated method found optimal ensemble even thought standard GA could not find it.

TABLE VI. THE NUMBER OF DIFFERENT OPTIMAL ENSEMBLES FOUND BY EACH METHOD (SGA: STANDARD GA, FS: FITNESS SHARING, AND DC: DETERMINISTIC CROWDING)

SGA	0	1	2	2	3	4
FS	1	4	3	0	1	8
DC	18	303	295	13	22	233
SGA	19	32	64	72	270	333
FS	33	26	66	65	310	291
DC	3905	3354	2135	2211	9073	8247

Figure 7 are the dendrograms for the first 30 optimal ensembles searched by the standard GA, fitness sharing and deterministic crowding. These are from single linkage, which is one of the hierarchical clustering algorithm. The average height of standard GA is lower than speciated GAs. This means the degree of diversity is smaller when we use standard GA. Between speciation methods, the degree of diversity of chromosomes searched by fitness sharing is larger than those searched by deterministic crowding. From Table VI and Fig. 7, it is clear that the speciation methods find the optimal ensembles with large amount (deterministic crowding) or much variously (fitness sharing).



(a) standard GA

(b) speciated GA – fitness sharing

(c) speciated GA – deterministic crowding

Figure 7. Comparison of dendrograms

To verify the difference between fitness sharing and deterministic crowding, we calculated the change of fitness with respect to the iterations and Figs. 8 and 9 indicate that respectively. In fitness sharing, the fitness is not so high because chromosomes share the fitness, but through the genetic operations it increases step by step. On the other hand, in deterministic crowding, the fitness increases steeply in an early stage because its strategy replaces bad one to good one.

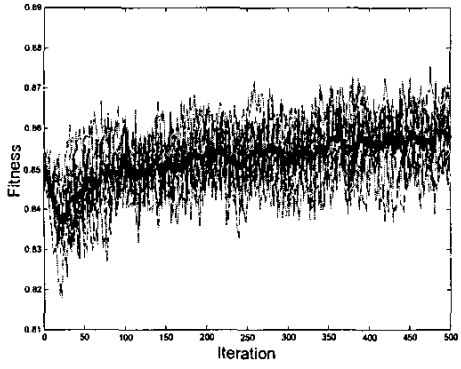


Figure 8. Change of fitness – fitness sharing (thick line is an average)

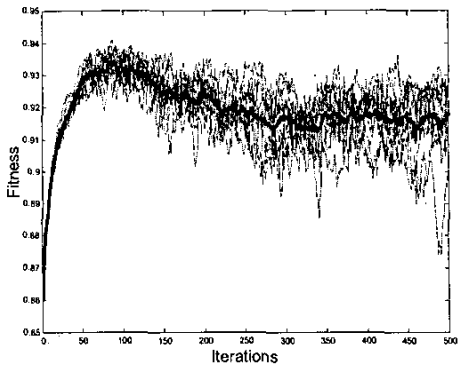


Figure 9. Change of fitness – deterministic crowding (thick line is an average)

From Figs. 8 and 9, we can know the fact that we can choose a speciation method according to the taste of researcher. If one wants early convergence then the deterministic crowding be useful, or one who wants the diversity of solutions can use the fitness sharing.

V. Conclusion

This paper has presented a method to search optimal ensemble of diverse feature-classifier pairs using speciated GA. Compared to the previous study, the proposed method finds out much more diverse solutions, and even some interesting solutions that could not be obtained by the conventional GA. Moreover we have shown that we can use different speciation strategy according to the required types of solution set: diverse of amount. Hence, we can say that it is important and interesting to use speciation method to find multiple or different solutions though it needs some additional operations.

The difference of mutation rates between 0 to 1 and 1 to 0 makes it possible to find optimal ensemble rapidly. When we set them to be same, the optimal ensemble was scarcely found. This means that a number of feature-classifier pairs to be participated to an ensemble make the performance of ensemble poor. It is considered that there are dependencies among individual feature-classifier pairs. For this reason, the dependency among feature-classifier pairs will be studied as a future work. In Fig. 9 the fitness very quickly reached a peak and then drops steadily. The reason is also the difference of mutation rates. It makes the ensemble with small number of participant feature-classifier pairs as well as fast finding of the optimal ensemble.

For the future study, we will apply proposed to multiclass dataset and use more sophisticated ensemble method such as Bayesian ensemble. The exploration of meta-ensembles based on the diverse classifiers found by

the speciated GA is also one of the agenda of our future works.

ACKNOWLEDGMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center(BERC) at Yonsei University and a grant of Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea.

REFERENCES

[1] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield and E. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitorning," *Science*, 286, pp. 531-537, 1999.

[2] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Y. D. Mack and J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl. Acad. Sci. USA*, 96, pp. 6745-6750, 1999.

[3] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer and N. Yakhini, "Tissue classification with gene expression profiles," *Journal of Computational Biology*, 7, pp. 559-584, 2000.

[4] L. Li, C. R. Weinberg, T. A. Darden and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, 17, pp. 1131-1142, 2001.

[5] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, 7, pp. 673-679, 2001.

[6] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, 2nd Ed., *Wiley Interscience*. 2001.

[7] S.-B. Cho and J. Ryu, "Classifying gene expression data of cancer using classifier ensemble with mutually exclusive features," *Proc. of the IEEE*, 90, pp. 1744-1753. 2002.

[8] C. Park and S.-B. Cho, "Evolutionary ensemble classifier for lymphoma and colon cancer classification," *Proc. of Congress on Evolutionary Computation*, pp. 2378-2385, 2003.

[9] K. Deb, D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," *Proc. 3rd Int. Conf. Genetic Algorithms*, pp. 42-50, 1989.

[10] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, 95, pp. 14863-14868, 1998.

[11] J. L. DeRisi, V. R. Iyer and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, 278, pp. 680-686, 1997.

[12] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. *Addison Wesley*. 1989.

[13] J. Liu and H. Iba, "Selecting informative genes with parallel genetic algorithms in tissue classification," *Genome Informatics*, 12, pp. 14-23, 2001.

[14] I. T. Jolliffe, Principal Component Analysis. *Springer, New York*. 1986.

[15] K. S. Hwang, S. -B. Cho, "Evolving diverse hardwares using speciated genetic algorithm," *Proc. of Congress on Evolutionary Computation*, pp. 437-442, 2002.

[16] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, 11, pp. 169-198, 1999.

[17] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, 36, pp. 105-139, 1999.

[18] L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, pp. 993-1001, 1990.

[19] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Jr. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, E. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown and L. M. Staudt, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, 403, pp. 503-511, 2000.

[20] H.-H. Won and S.-B. Cho, "Data mining for gene expression profiles from DNA microarray," *International Journal of Software Engineering and Knowledge Engineering*, 13, pp. 593-608, 2003.