Reliable classification of two-class cancer data using evolutionary algorithms

Kalyanmoy Deb*, A. Raji Reddy

Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, Kanpur 208 016, India

Abstract

In the area of bioinformatics, the identification of gene subsets responsible for classifying available disease samples to two or more of its variants is an important task. Such problems have been solved in the past by means of unsupervised learning methods (hierarchical clustering, self-organizing maps, k-mean clustering, etc.) and supervised learning methods (weighted voting approach, k-nearest neighbor method, support vector machine method, etc.). Such problems can also be posed as optimization problems of minimizing gene subset size to achieve reliable and accurate classification. The main difficulties in solving the resulting optimization problem are the availability of only a few samples compared to the number of genes in the samples and the exorbitantly large search space of solutions. Although there exist a few applications of evolutionary algorithms (EAs) for this task, here we treat the problem as a multiobjective optimization problem of minimizing the gene subset size and minimizing the number of misclassified samples. Moreover, for a more reliable classification, we consider multiple training sets in evaluating a classifier. Contrary to the past studies, the use of a multiobjective EA (NSGA-II) has enabled us to discover a smaller gene subset size (such as four or five) to correctly classify 100% or near 100% samples for three cancer samples (Leukemia, Lymphoma, and Colon). We have also extended the NSGA-II to obtain multiple non-dominated solutions discovering as much as 352 different three-gene combinations providing a 100% correct classification to the Leukemia data. In order to have further confidence in the identification task, we have also introduced a prediction strength threshold for determining a sample's belonging to one class or the other. All simulation results show consistent gene subset identifications on three disease samples and exhibit the flexibilities and efficacies in using a multiobjective EA for the gene subset identification task.
© 2003 Elsevier Ireland Ltd. All rights reserved.

Keywords: Gene subset identification; Classification of cancer data; DNA microarray; Prediction strength; Multiobjective optimization; Evolutionary algorithms

1. Introduction

The process of making proteins based on the information encoded in the DNA sequence is known as protein synthesis. Protein synthesis consists of three stages: transcription, splicing and translation. A strand of DNA molecule is transcribed to an mRNA sequence and then proteins are formed by creating amino acid sequences. Since mRNA is an essential by-product of protein synthesis, mRNA levels can provide a quantification of *gene expression levels*. Thus, a gene expression level is thought to correlate with the approximate number copies of mRNA produced in a cell. Microarray technology enables the simultaneous measurement of mRNA levels of thousands of genes, thus providing

* Corresponding author.
 E-mail addresses: deb@iitk.ac.in (K. Deb), arreddy@iitk.ac.in (A. Raji Reddy).
URL: http://www.iitk.ac.in/kangal.

an insight into which genes are expressed in a particular cell type, at a particular time, under particular conditions.

The DNA microarray is an orchestrated arrangement of thousands of different single-stranded DNA probes in the form of cDNAs or oligonucleotides immobilized onto a glass or silicon substrate. The underlying principle of microarray technology is the hybridization or the base pairing of nucleotides. An array chip, hybridized to a labeled unknown cDNA extracted from a particular tissue of interest, makes it possible to measure simultaneously the expression level in a cell or tissue sample for each gene represented on the chip. DNA microarrays (Gershon, 2002) can be used to determine which genes are being expressed in a given cell type at a particular time and under particular conditions, to compare the gene expression in two different cell types or tissue samples, to examine changes in gene expression at different stages in the cell cycle and to assign probable functions to newly discovered genes with the expression patterns of known genes. Moreover, the DNA microarray provides a global perspective of gene expression levels, which can be used in gene clustering tools (Alon et al., 1999), tissue classification methods (Golub et al., 1999), identification of new targets for therapeutic drugs (Clarke et al., 1999), etc. However, the important issue in such problems is that the availability of only a few samples compared to the large number of genes, a matter which makes the classification task more difficult. Furthermore, many of the genes are not relevant to the distinction between different tissue types and introduce noise in the classification process. Therefore, identification of small subset of informative genes, sufficient to distinguish between different tissue types is one of the crucial tasks in bioinformatics area and should be paid more attention. This work goes further in this direction and focuses on the topic of identification of small set of informative genes for reliable classification of cancer samples to two classes based on their expression levels. For some well-known cancer diseases, microarray data for two different variants of the disease are available on the Internet. Each data contain the gene accession numbers and corresponding expression values. The data considered here contain 62–96 different samples having 2000–7129 genes. The purpose of the gene subset identification problem is to find the smallest set of

genes (a classifier) which will enable a correct classification of as many available samples as possible.

The gene subset identification problem reduces to an optimization problem consisting of a number of objectives (Liu and Iba, 2002; Liu et al., 2001). Although the optimization problem is multiobjective, all previous studies have scalarized multiple objectives into one. In this paper, we have used a multiobjective evolutionary algorithm (MOEA) to find the optimum gene subset for three commonly used cancer data sets—Leukemia, Lymphoma and Colon. By using three objectives for minimization (gene subset size, number of misclassifications in training, and number of misclassifications in test samples) several variants of a particular MOEA (modified non-dominated sorting GA or NSGA-II) are applied to investigate if gene subsets exist with 100% correct classifications in both training and test samples. Since the gene subset identification problem may involve multiple gene subsets of the same size causing identical number of misclassifications (Kohavi and John, 1997), in this paper, we have proposed and developed a novel multimodal NSGA-II for finding multiple gene subsets simultaneously in one single simulation run.

One other important matter in the gene subset identification problem is the confidence level with which the samples are classified. We introduce the classification procedure based on the prediction strength consideration, suggested in Golub et al. (1999), in the proposed multimodal NSGA-II to find gene subsets with 20 and 30% prediction strength thresholds.

In the reminder of the paper, we briefly discuss the procedure of identifying gene subsets in a set of cancer samples and then discuss the underlying optimization problem. Thereafter, we discuss the procedure of using NSGA-II to this problem. Finally, we present simulation results on three disease samples using a number of variants of NSGA-II, the proposed multimodal NSGA-II, and prediction strength considerations. We conclude the paper by discussing the merits of using an MOEA to the gene subset identification problem.

2. Identification of gene subsets

In this study, we concentrate on classifying samples for two classes only, although modifications can

be made to generalize the procedure for any number of classes. In most problems involving identification of gene subsets in bioinformatics, the number of samples available is small when compared to the gene pool size. This aspect makes it difficult to identify which and how many genes are correlated with different classifications. It is a common practice in machine learning algorithms to divide the available data sets into two groups: one used for training purposes for generating a classifier and the other used for testing the developed classifier. Although most classification methods will perform well on samples used during training, it is necessary and important to test the developed classifier on unseen samples which were not used during training to get a realistic estimate of performance of the classifier and to avoid any training error. Most commonly employed method to estimate the accuracy in such situations is the cross-validation approach (Ben-Dor et al., 2000). In cross-validation, the training data set (say t of them) is partitioned into k subsets, $C_1, C_2, \ldots, C_k$ (k is also known as the number of cross-validation trials). Each subset is kept roughly of the same size. Then a classifier is constructed using $t_i = t - |C_i|$ samples to test the accuracy on the samples of C_i. The construction procedure is described a little later. Once the classifier is constructed using t_i samples, each sample in C_i is tested using the classifier for its class A or B. Since these t_i samples are used as training samples, we can compare the classification given by the above procedure with the actual class in which the sample belongs. If there is a mismatch, we increment the training sample mismatch counter τ_{train} by one. This procedure is repeated for all samples in C_i in the i-th subset. Thereafter, this procedure is repeated for all k subsets and the overall training sample mismatch counter τ_{train} is noted. Cross-validation has several important properties including that the classifier is tested on each sample exactly once. One of the most-commonly used method in cross-validation is leave-one-out-cross-validation (LOOCV), in which only one sample in the training set is withheld (or $|C_i| = 1$ for all i) and the classifier is constructed using the rest of the samples to predict the class of withheld sample. Thus, in the LOOCV there are $k = t$ subsets. In this study, we have used LOOCV to estimate the number of mismatches in the training set.

However, to predict the class of each test sample, we use the classifier obtained from all t training samples. The number of mismatches τ_{test} obtained by comparing the predicted class with the actual class of each sample is noted. Note that the LOOCV procedure is not used with the test samples, instead the classifier obtained using the training samples is directly used to find the number of mismatches in the test samples (Golub et al., 1999).

2.1. Class prediction procedure

For the identification task, we begin with the gene expression values available for cancer disease samples obtained from the DNA microarray experiments. For many cancer cell types, such data are available on the Internet. In addition to the gene expression values, each sample in the data set is also labelled to belong to one class or the other. For identifying genes responsible for proper classification of samples into two classes, the available data set is divided into two groups: one used for the training purpose and the other used for the testing purpose. The top-left box in Fig. 1 shows a sketch of gene expression values corresponding to a gene g for all training samples.

Although in some disease samples, such values are already available in normalized form, for some other samples they have to be processed. In this figure, we consider the latter case and describe the procedure of filtering and normalizing the data for their use in the identification process. Since negative and very small gene expression values arise mainly due to various experimental difficulties, we first filter all expression values as follows. Any value less than θ (we have used $\theta = 20$ for the Leukemia samples here) is replaced by θ. For classifier genes, it is expected that there will be a wide variation in the gene expression values differentiating one disease sample from the other. For this reason, we eliminate genes with not much variation in its expression values. To achieve this, we first calculate the difference of the maximum and minimum gene expression values for every gene in the available samples. If the difference is less than a quantity β_1 (we have used $\beta_1 = 500$ in the Leukemia sample here), we discard that gene for further processing. Thereafter, we calculate the ratio of maximum and minimum expression values of the gene. If the ratio is less than β_2 (we have used $\beta_2 = 5$ here), we discard
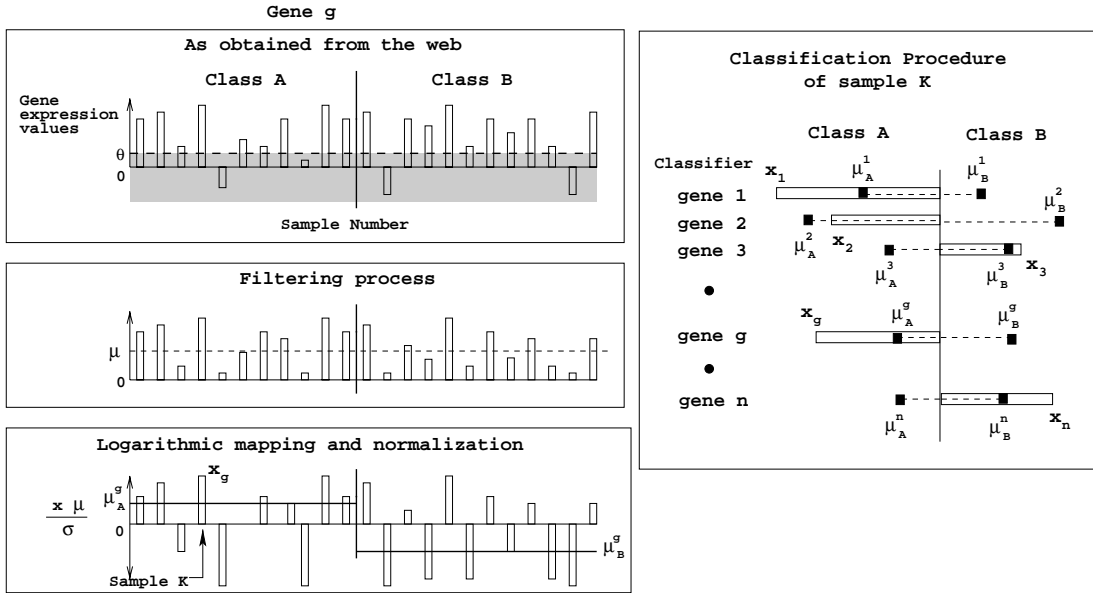
Fig. 1. Processing of gene expression values and the classification procedure are shown. The left figure shows different pre-processing steps used to the raw gene expression data. The right figure illustrates the classification procedure of a sample. The classifier has n genes. The parameters μ_A^g and μ_B^g are mean expression values of gene g in training samples of class A and B, respectively. x_g is the actual gene expression value of gene g in the sample K. Refer Eq. (1) for the exact procedure.

the gene from further consideration. Otherwise, the gene is included in the study. Based on the suggestion in (Golub et al., 1999), the logarithm of the gene expression values (denoted as x_g) are calculated and then normalized as follows: $\bar{x}_g = (x_g - \mu)/\sigma$. Here, μ and σ are the mean and standard deviation of the x_g values. We call $\bar{x}_g$ as the normalized gene expression value. In some cases (such as in Lymphoma and Colon cancer samples), the logarithm of gene expression values are already available. In those cases, there is no need to follow the above procedure. although the values are normalized for further processing.

For a given gene subset G, we can predict the class of any sample K (whether belonging to A or B) with respect to a known set of S samples in the following manner. Let us say that S samples are composed of two subsets S_A and S_B, belonging to classes A and B, respectively. First, for each gene $g \in G$, we calculate the mean μ_A^g and standard deviation σ_A^g of the normalized gene expression levels $\bar{x}_g$ of all S_A samples. This procedure is shown in Fig. 1. The same procedure is repeated for the class B samples and μ_B^g and σ_B^g are computed. Thereafter, we determine the class of the

sample K using a standard weighted voting approach (Golub et al., 1999):

$$
\begin{aligned}
&\text{class}(x) \\
&= \text{sign}\left\{ \sum_{g \in G} \left(\frac{\mu_A^g - \mu_B^g}{\sigma_A^g + \sigma_B^g} \right) \left(\bar{x}_g - \frac{\mu_A^g + \mu_B^g}{2} \right) \right\},
\end{aligned}
\tag{1}
$$

If the right term of the above equation is positive, the sample belongs to class A and if it is negative, it belongs to class B.

2.2. Resulting optimization problem

One of the objectives of the above task is to identify the smallest size of a gene subset for predicting the class of all samples correctly. Although not obvious, when a too small gene subset is used, the classification procedure becomes erroneous. Thus, minimization of class prediction mismatches in the training and test samples are also important objectives. Here, we use these three objectives in a multiobjective optimization

problem: The first objective f_1 is to minimize the size of gene subset in the classifier. The second objective f_2 is to minimize the number of mismatches in the training samples calculated using the LOOCV procedure and is equal to τ_{train} described above. The third objective f_3 is to minimize the number of mismatches τ_{test} in the test samples.

2.3. Solution procedure using evolutionary algorithms

Similar to a previous study (Liu and Iba, 2002), we use a ℓ-bit binary string (where ℓ is the number of filtered genes in a disease data set) to represent a solution. For a particular string, the positions marked with a 1 are included in the gene subset for that solution. For example, in the following example of a 10-bit string (representing a total of 10 genes in a data set), first, third, and sixth genes are considered in the gene subset (also called the *classifier*):

(1 0 1 0 0 1 0 0 0 0)

The procedure of evaluating a string is as follows. We first collect all genes for which there is a 1 in the string in a gene subset G. Thereafter, we calculate f_1, f_2, and f_3 as described above as three objective values associated with the string. We initialize each population member by randomly choosing at most 10% of string positions to have a 1. Since the gene subset size is to be minimized, this biasing against 1 in a string allows

an EA to start with good members in the population. The population is assumed to have a fixed size of N strings.

To handle three objectives, we have used a multiobjective GA (NSGA-II) (Deb et al., 2002), which we briefly described here. NSGA-II has the following features:

(1) It uses an elitist selection.
(2) It uses an explicit diversity preserving mechanism.
(3) It emphasizes the *non-dominated* solutions (Miettinen, 1999).

In NSGA-II, the offspring population Q_t (of size N) is first created by using the parent population P_t (of size N) and the usual genetic operators (such as single-point crossover and bit-wise mutation operators) (Goldberg, 1989). Thereafter, the two populations are combined together to form R_t of size $2N$. Then, a non-dominated sorting procedure (Deb, 2001) is used to classify the entire population R_t. Once the non-dominated sorting is over, the new parent population P_{t+1} is created by choosing solutions of different non-dominated fronts, one at a time. The ordering starts with strings from the best non-dominated front and continues with strings of the second non-dominated front, followed by the third non-dominated front, and so on. Since the overall population size of R_t is $2N$, not all fronts can be accommodated in the new parent population. The fronts which could not be accommodated at all are simply
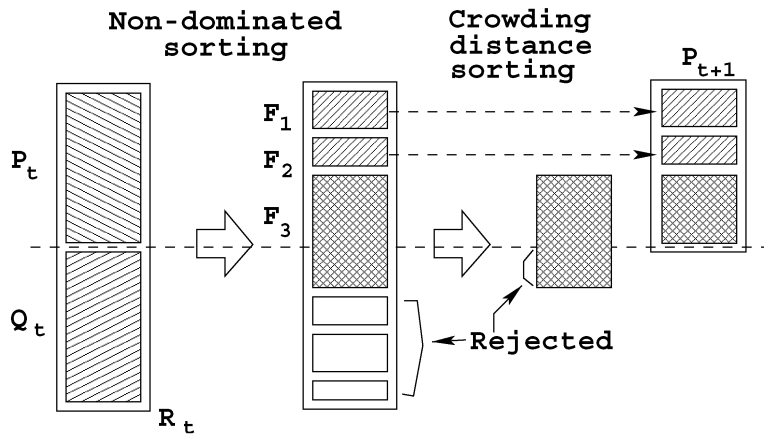

Fig. 2. Schematic of the NSGA-II procedure. The combination of parent and offspring population ensures elitist selection, the non-dominated sorting ensures progress towards the Pareto-optimal front, and the crowding distance sorting ensures diversity among obtained solutions.

deleted. However, while the last allowed front is being considered, there may exist more strings in it than the remaining population slots in the new population. This scenario is illustrated in Fig. 2.

Instead of arbitrarily choosing some strings from this last front, the strings which will make the *diversity* of the selected strings the maximum are chosen. For each string, we calculate a computationally simple *crowding distance* measuring the Euclidean distance among the neighboring strings in the objective space. Thereafter, those strings having largest crowding distance values are chosen to become the new parent population. This procedure is continued for a maximum of user-defined T iterations. For detail information about NSGA-II, readers are referred to Deb et al. (2002). Due to the emphasis of the non-dominated solutions, maintenance of diversity among population members, and an elitist approach, NSGA-II has been successful in converging quickly close to the true Pareto-optimal front with a well-diversed set of solutions in the objective space.

3. Simulation results

In this section, we show the application of NSGA-II and its variants on three different cancer data sets: Leukemia, Lymphoma, and Colon. The flexibility in using different modifications to NSGA-II in the identification task is mostly demonstrated for the well-studied Leukemia samples. The Leukemia data set is a collection of gene expression measurements from 72 Leukemia (composed of 62 bone marrow and 10 peripheral blood) samples reported elsewhere (Golub et al., 1999). It contains an initial training set composed of 27 samples of acute lymphoblastic Leukemia (ALL) and 11 samples of acute myeloblastic Leukemia (AML), and an independent test set composed of 20 ALL and 14 AML samples. The gene expression measurements were taken from high density oligonucleotide microarrays containing 7129 probes for 6817 human genes. These data are available at http://www.genome.wi.mit.edu/MPR.

The Lymphoma data set is a collection of expression measurements from 96 normal and malignant lymphocyte samples reported elsewhere (Alizadeh et al., 2000). It contains 42 samples of diffused

large B-cell Lymphoma (DLBCL) and 54 samples of other types. The Lymphoma data containing 4026 genes is available at http://llmpp.nih.gov/lymphoma/data/figure1.cdt.

The Colon data set is a collection of 62 expression measurements from Colon biopsy samples reported elsewhere (Alon et al., 1999). It contains 22 normal and 40 Colon cancer samples. The Colon data having 2000 genes is available at http://microaaray.princeton.edu/oncology.

3.1. Minimization of gene subset size

First, we apply the standard NSGA-II on 50 genes which are used in another Leukemia study (Golub et al., 1999) to minimize two objectives: (i) the size of gene-subset (f_1), and (ii) the sum of mismatches ($f_2 + f_3$) in the training and test samples. In this case, filtering with a threshold $\theta = 20$ is performed and all 50 genes qualify the β-test (with $\beta_1 = 500$ and $\beta_2 = 5$) mentioned above. The gene expression value of the 50 genes is also normalized using the procedure described in Fig. 1. We choose a population of size 500 and run NSGA-II for 500 generations. With a single-point crossover with a probability of 0.7 and a bit mutation with a probability of $p_m = 1/50$, we obtain five non-dominated solutions, as shown in Fig. 3. We have found a solution with zero mismatches in all training and test samples. This solution requires only four (out of 50) genes to correctly identify all 72 samples as either belonging to class ALL or AML. The obtained solution has the following gene accession numbers: M31211, M31523, M23197 and X85116. The non-dominated set also has other solutions with reduced number of genes but with non-zero mismatches.

One difficulty with the above two-objective problem is that it is not clear from solutions with non-zero mismatches whether the mismatches occur in the training or in the test samples. To differentiate this matter, we now consider all three objectives—the gene subset size, the mismatches in the training samples, and the mismatches in the test samples. Fig. 4 shows the corresponding non-dominated solutions with identical parameter settings. For clarity, we have not shown the solution having no genes (causing 38 and 34 mismatches in training and test samples, respectively) in this figure. It is clear from the figure that smaller gene
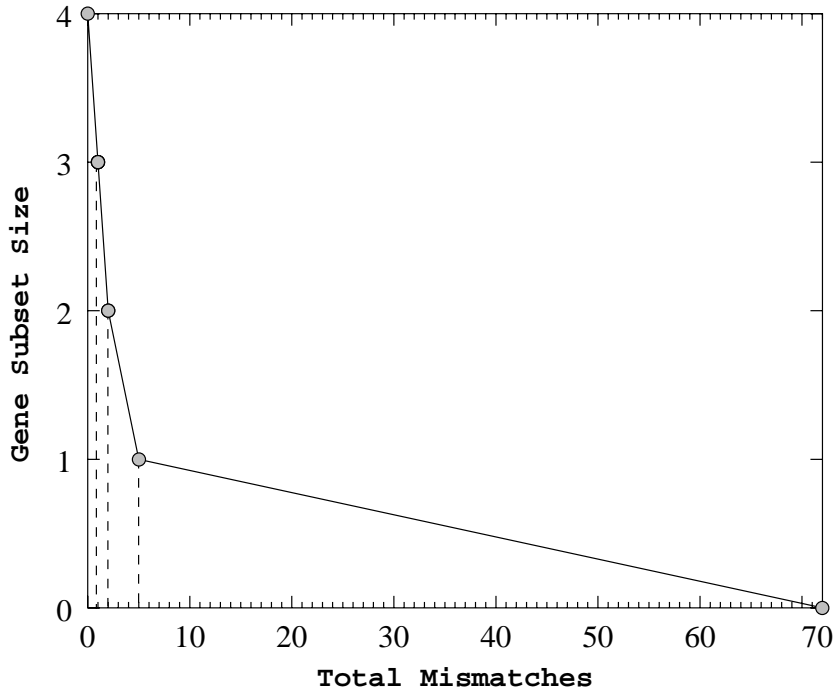
Fig. 3. Two-objective solutions obtained using NSGA-II for the Leukemia samples. The total mismatches indicate the sum of mismatches in training and test samples. With a null classifier, all 72 samples are not classified correctly. The one-gene classifier causes five mismatches, whereas a four-gene classifier causes no mismatch in all training and test samples.
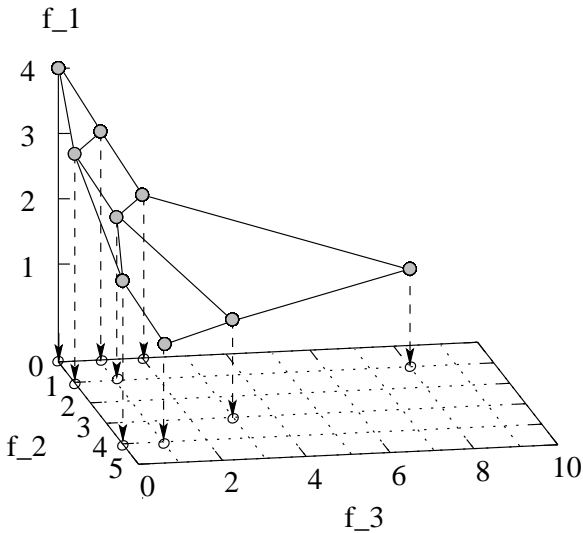

Fig. 4. Three-objective solutions obtained using NSGA-II for the Leukemia samples. Axis labels f_1, f_2, and f_3 indicate classifier size, number of mismatches in training samples and number of mismatches in test samples, respectively.

subsets cause more mismatches. Interestingly, four different four-gene solutions are found to provide 100% classification and these solutions are different from that obtained in the two-objective case. This indicates that there exist multiple gene-combinations for a 100% perfect classification, a matter we discuss in more detail later. The three-objective case also clearly shows the break up in the mismatches. For example, Fig. 3 shows that the two-gene solution has two mismatches, whereas Fig. 4 shows that there exist three two-gene solutions (four mismatches in training samples only, two mismatches in test samples only, and one mismatch each in training and test samples).

3.2. Maximization of gene subset size

Although four genes are enough to correctly classify all 72 samples, there may be larger sized subsets which can also correctly classify all 72 samples. In order to find the largest size gene subset (of 50
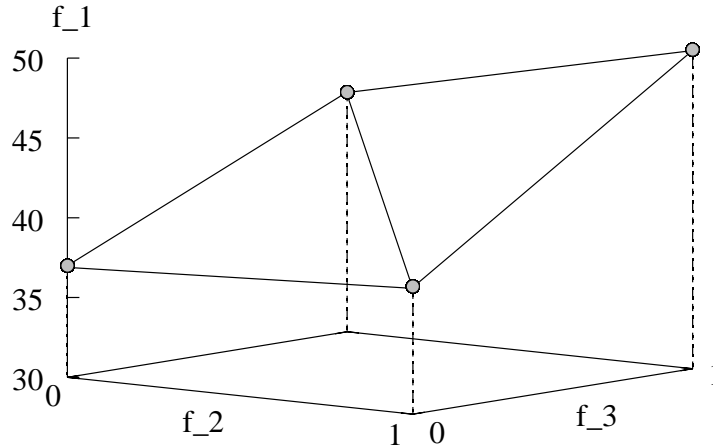
Fig. 5. Maximization of gene subset size for the Leukemia samples. A classifier having as many as 37 genes is able to correctly classify all 72 samples.

genes) which can correctly classify all 72 samples, we apply NSGA-II with the above three objectives, but instead of minimizing the first objective we maximize it. With identical NSGA-II parameters, the obtained non-dominated set has a solution having 37 genes which can correctly classify all 72 samples (Fig. 5). If more genes are added, the classification becomes less than perfect. The non-dominated set also involves a solution having all 50 genes which causes one mismatch each in the training and test samples. This outcome matches with that found exclusively for the 50 genes elsewhere (Golub et al., 1999).

When the first objective is maximized and other two objectives are minimized, the previously found solution with the perfectly classified four gene-subset is dominated by the perfectly classified 37 gene-subset solution. If one is interested in finding what would be the maximum size of gene-subset for a perfect classification, the procedure of this subsection becomes useful.

3.3. Modified domination criterion for multiple gene subset sizes

The above two subsections showed how solutions with smallest and largest number of gene subsets can be found by simply using minimization and maximization of the first objective, respectively. However, in order to find the entire spread of solutions on the

first objective axis simultaneously in one single simulation run, we can modify the domination criterion in NSGA-II.

Definition 1 (Biased dominance $\prec_i$ criterion). *Solution $\boldsymbol{x}^{(1)}$ biased-dominate ($\prec_i$) solution $\boldsymbol{x}^{(2)}$ if $f_j(\boldsymbol{x}^{(1)}) \leq f_j(\boldsymbol{x}^{(2)})$ for all objectives ($j = 1, 2, \dots, M$) and $f_k(\boldsymbol{x}^{(1)}) < f_k(\boldsymbol{x}^{(2)})$ for at least one objective other than the i-th objective ($k = 1, 2, \dots, M$ and $k \neq i$).*

This biased-domination definition differs from the original dominance definition (Deb, 2001) in that any two solutions with identical f_j values will not dominate each other. This way, multiple solutions lying along the f_i ($j \neq i$) axis can be all non-dominated to each other.

When we apply NSGA-II with identical parameter settings as in the previous subsection and with the above biased dominance criterion (or $\prec_1$) for f_1, all solutions ranging from four gene-subset to 36 gene-subset (in the interval of one) are found to produce a 100% perfect classification. Fig. 6 shows the obtained non-dominated solutions. Instead of using two NSGA-II applications as shown in the previous two subsections, a whole spectrum of solutions can be obtained in one single simulation run. This illustrates the flexibility of using NSGA-II in the gene subset identification task. The availability of multiple classifiers each causing 100% correct classification
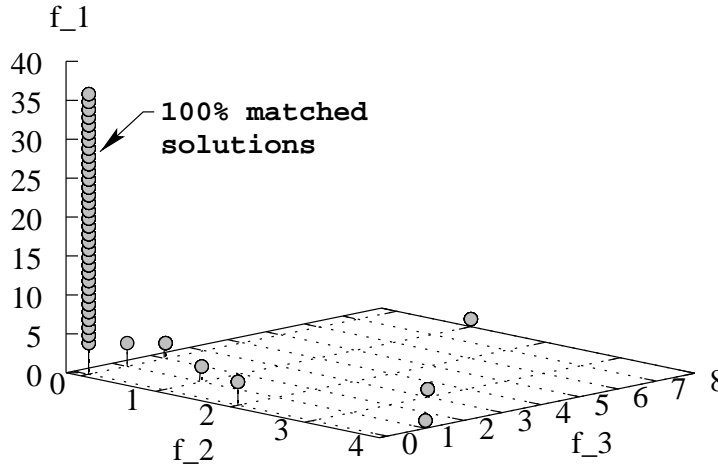
Fig. 6. NSGA-II solutions with modified domination criterion for the Leukemia samples. The power of NSGA-II is demonstrated by finding four to 36-gene classifiers, each providing 100% correct classification.

to all available samples should provide a plethora of information for further biological study.

3.4. Multimodal MOEAs for multiple solutions

We have observed in Section 3.1 that the gene subset identification task with multiple objectives may involve multimodal solutions, meaning that for a point in the objective space (Figs. 1–6), there may exist more than one solution in the decision variable space (Hamming space). When this happens, it becomes important and useful for a biologist to know which all gene combinations may provide an identical classification. In this subsection, we suggest a modified NSGA-II procedure for identifying such multimodal solutions. We define two multimodal solutions in the context of multiobjective optimization as follows:

Definition 2 (Multimodal solutions). *If for two different solutions* $\boldsymbol{x}^{(1)}$ *and* $\boldsymbol{x}^{(2)}$ *satisfying* $\boldsymbol{x}^{(1)} \neq \boldsymbol{x}^{(2)}$*, all objective values are the same, or* $f_i(\boldsymbol{x}^{(1)}) = f_i(\boldsymbol{x}^{(2)})$ *for all* $i = 1, 2, \ldots, M$*, then solutions* $\boldsymbol{x}^{(1)}$ *and* $\boldsymbol{x}^{(2)}$ *are multimodal solutions.*

In such problems, more than one solutions in the decision variable space maps to one point in the objective space. We are not aware of any MOEA which has been attempted for this task or for any other

similar problem. Fortunately, the gene subset identification problem results in multiple gene subset combinations each producing an identical outcome in terms of their mismatches in the training and test data sets.

Recall that in an NSGA-II iteration, the parent P_t and offspring Q_t populations are combined to form an intermediate population R_t of size $2N$. Thereafter, R_t is sorted according to a decreasing order of non-domination level ($F_1, F_2, \ldots$). In the original NSGA-II, solutions from each non-dominated level (starting from the best set) are accepted until a complete set cannot be included without increasing the designated population size. A crowding operator was then used to choose the remaining population members. We follow an identical procedure here until the non-dominated sorting is finished. Thereafter, we use a slightly different procedure. But before we describe the procedure, let us present two definitions:

Definition 3 (Duplicate solutions). *Two solutions* $\boldsymbol{x}^{(1)}$ *and* $\boldsymbol{x}^{(2)}$ *are duplicates to each other if* $\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(2)}$*.*

It follows that duplicate solutions have identical objective values.

Definition 4 (Distinct objective solutions). *Two solutions* $\boldsymbol{x}^{(1)}$ *and* $\boldsymbol{x}^{(2)}$ *are distinct objective solutions if* $f_i(\boldsymbol{x}^{(1)}) \neq f_i(\boldsymbol{x}^{(2)})$ *for at least one i.*
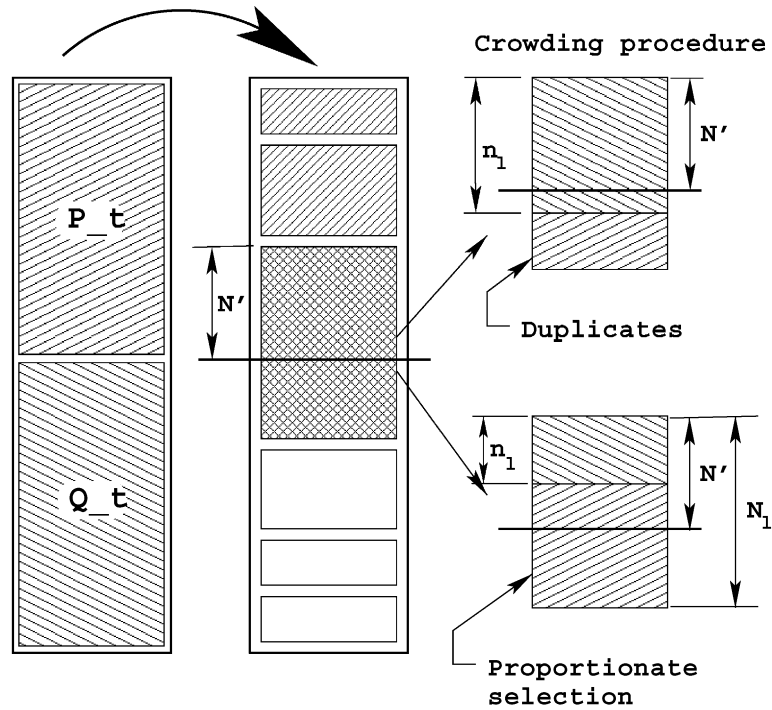
Fig. 7. Schematic of the multimodal NSGA-II procedure is shown. The original NSGA-II is modified in choosing N' solutions from N_l. Refer to the text for details.

First, we delete the duplicate solutions from each non-domination set in R_t. Thereafter, each set is accepted as usual until the last front F_l which can be accommodated. Let us say that solutions remaining to be filled before this last front is considered is N' and the number of non-duplicate solutions in the last front is N_l ($>N'$). We also compute the number of distinct objective solutions in the set F_l and let us say it is n_l (obviously, $n_l \leq N_l$). This procedure is illustrated in Fig. 7.

If $n_l \geq N'$ (the top case shown in the figure), we use the usual crowding distance procedure to choose N' most dispersed and distinct solutions from n_l solutions. In this case, even if there exist multimodal solutions to any n_l solutions, they are ignored due to lack of space in the population. The major modification to NSGA-II is made when $n_l < N'$ (bottom case in the figure). This means that although there are fewer distinct solutions than the population slots, the distinct solutions are multimodal. However, the total number of multimodal solutions of all distinct solutions (N_l) is more than the remaining population slots. Thus,

we need to make a decision of choosing a few solutions. The purpose here is to have at least one copy of each distinct objective solution and as many multimodal copies of them so as to fill up the population. Here, we choose a strategy in which every distinct objective solution is allowed to have a proportionate number of multimodal solutions as they appear in F_l. To avoid loosing any distinct objective solutions, we first allocate one copy of each distinct objective solution, thereby allocating n_l copies. Thereafter, the proportionate rule is applied to the remaining solutions ($N_l - n_l$) to find the accepted number of solutions for the ith distinct objective solution as follows:

$$\alpha_i = \frac{N' - n_l}{N_l - n_l}(m_i - 1), \qquad (2)$$

where m_i is the number of multimodal solutions of the i-th distinct objective solution in F_l, such that $\sum_{i=1}^{n_l} m_i = N_l$. It is true that $\sum_{i=1}^{n_l} \alpha_i = N' - n_l$. The final task is to choose ($\alpha_i + 1$) multimodal solutions from m_i copies for the i-th distinct objective solution. Although a sharing strategy (Goldberg and

Richardson, 1987) can be used to choose the maximally different multimodal solutions, here we simply choose them randomly. Along with the duplicate-deletion strategy, the random acceptance of a specified number multimodal solutions to each distinct objective solution ensures a good spread of solutions in both objective and decision variable space. In the rare occasions of having less than N non-duplicate solutions
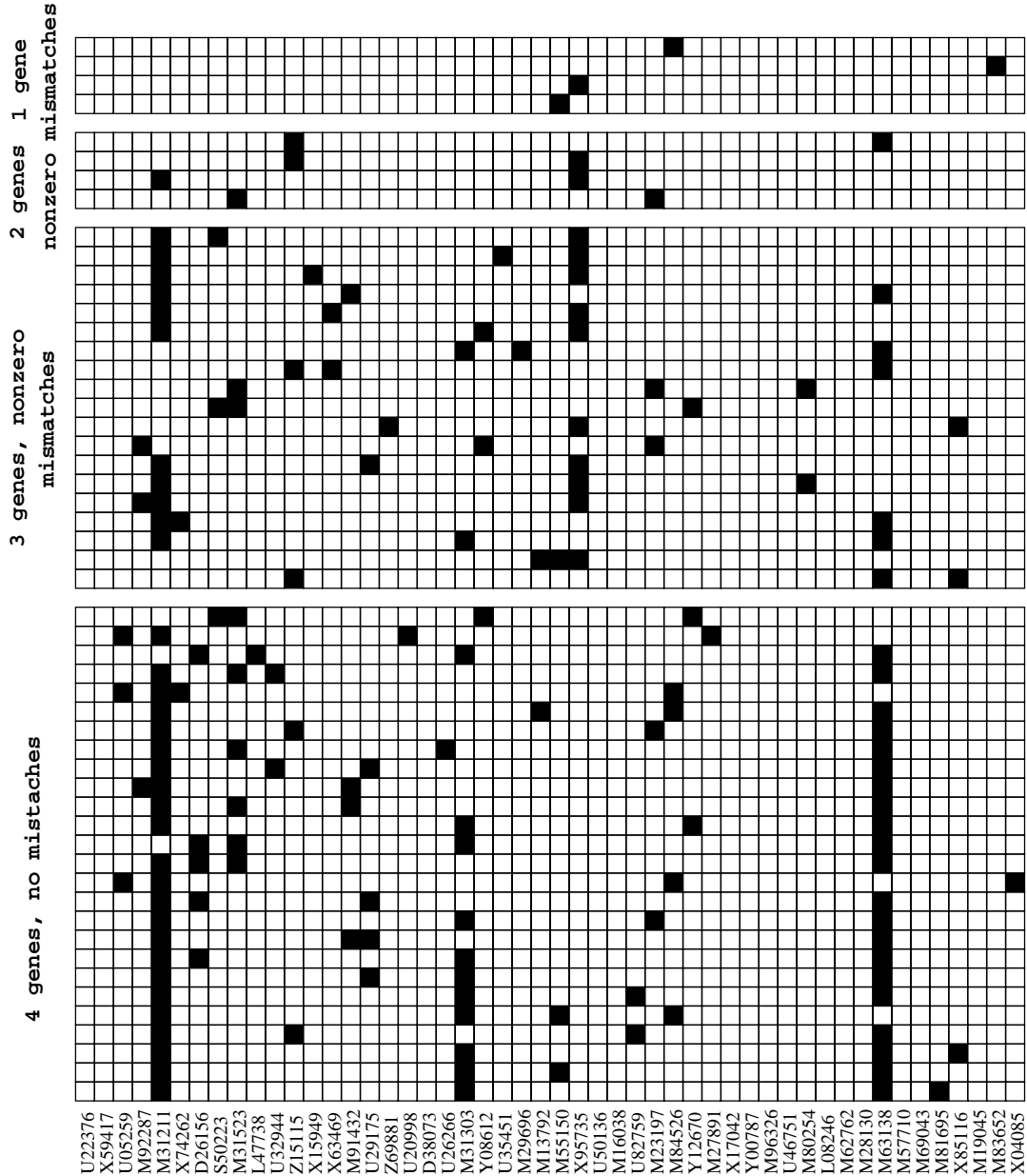

Fig. 8. Multimodal solutions obtained using NSGA-II for the Leukemia samples. Each row shows a classifier obtained from the 50-gene Leukemia data set. The lower set of 26, four-gene classifiers can all make 100% correct classifications to all 72 available samples. For each classifier, the participating genes are marked in solid boxes.

in R_t, new random solutions are used to fill up the population. For a problem having many multimodal solutions, the latter case will occur often and the above systematic preservation of distinct objective solutions and then their multimodal solutions will maintain a rich collection of multimodal Pareto-optimal solutions.

First, We have applied the multimodal NSGA-II to the 50-gene Leukemia data set under the condition that minimization of all three objective functions. With 500 population sizes running for 500 generations, we have obtained the same non-dominated front as shown in Fig. 4. However, each distinct objective solution has a number of multimodal solutions. For the solution with four gene-subset causing 100% classification on both training and test samples, the multimodal NSGA-II has found 26 different solutions. All these four-gene solutions are shown in Fig. 8.

The figure highlights an interesting aspect. Among different four-gene combinations, three genes (accession numbers M31211, M31303, and M63138) frequently appear in the obtained gene subsets. Of the 26 solutions, these three genes appear together in eight of them. Such information about frequently appearing genes in high-performing classifiers is certainly useful to biologists. Interestingly, two of these three genes also appear quite frequently in other trade-off non-dominated solutions with non-zero mismatches, as shown in Fig. 8.

It is also interesting to note that when multimodal NSGA-II was not used (in Section 3.1), only four distinct solutions with 100% correct classification were obtained. These four solutions are also rediscovered

in the set of 26 multimodal solutions shown in Fig. 8. This illustrates the efficiency of the proposed multimodal NSGA-II approach in finding and maintaining multimodal Pareto-optimal solutions.

3.5. Complete Leukemia data set

In the case of the complete Leukemia samples each having 7129 genes, we first use the filtering procedure with $\theta = 20$, $\beta_1 = 500$ and $\beta_2 = 5$. These values were recommended and used in (Golub et al., 1999). This reduces the number useful genes to 3859 from 7129 genes available in the raw data. Thereafter, we normalize the gene expression values of these genes using the procedure depicted in Fig. 1. Because of the large string length requirement, we have chosen a population of size 1000 and iterated the NSGA-II for 1000 times. We have used a mutation probability of 0.0005, so that on an average about one bit gets mutated in the complete string.

The perfect classification is obtained with a classifier having only three genes. However, the number of three-gene combinations resulting in a 100% correct classification is 352, meaning that any one of these 352 three-gene combinations will produce a 100% classification of training as well as test samples. Recall that the 50-gene study above on the same Leukemia samples has resulted in four-gene subsets. Table 1 also shows other classifiers having smaller number of genes resulted in less than 100% correct classifications.

To investigate the effect of mutation probability in the obtained gene subset size, we have rerun the

Table 1
Multimodal solutions for three disease samples

Leukemia samples					Lymphoma samples					Colon samples			
f_1	f_2	f_3	α		f_1	f_2	f_3	α		f_1	f_2	f_3	α
3	0	0	352		5	0	0	121		7	0	1	25
2	1	0	10		4	0	1	17		6	1	1	4
2	0	2	2		3	1	0	1		4	0	2	3
1	5	0	1		3	0	3	2		3	4	3	1
1	3	1	1		2	1	2	1		3	0	4	1
					2	2	1	1		2	3	4	2
					1	4	5	2		2	2	6	2
										1	4	5	1

Parameters f_1, f_2, f_3, and α represent gene subset size, mismatches in training samples, mismatches in test samples, and the number of multimodal solutions obtained with (f_1, f_2, f_3) values.
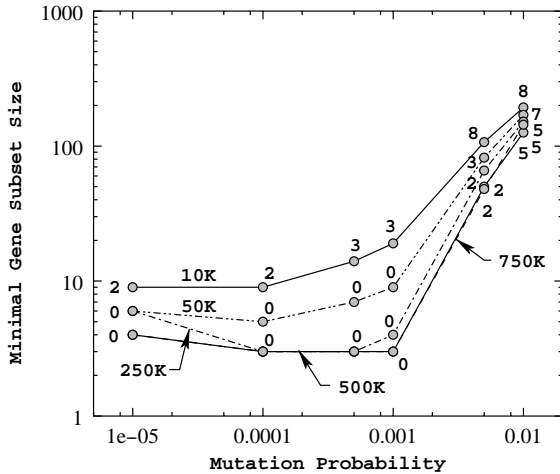
Fig. 9. The effect of mutation probability on the obtained minimum gene subset size. The number near each point indicates the total number of mismatched samples obtained with the gene subset.

above NSGA-II with different mutation probabilities. The minimum gene subset size obtained at the end of 10,000, 50,000, 250,000, 500,000, and 750,000 evaluations are recorded and plotted in Fig. 9. It is clear that with the increase in the number of evaluations (or generations), NSGA-II reduces the minimum gene subset size for any mutation probability. However, after 500,000 evaluations, there is no change in the optimal gene subset size for most mutation probabilities, meaning that there is no need to continue running NSGA-II after these many evaluations. When an optimum mutation probability ($p_m \sim 1/\ell$) is chosen, about 250,000 evaluations are enough. The figure also indicates the total number of mismatched samples in each case. It is clear that a 100% correct classification cannot be obtained with less than three genes.

3.6. Complete Lymphoma data set

Next, we apply the multimodal NSGA-II to the Lymphoma data set having 4026 genes. In this case, the printed gene expression values are already log-transformed. However, some gene expression values are missing. For these genes, the expression values are derived based on suggestion given in Ben-Dor et al. (2000). Thus, we consider all 4026 genes in our study and normalize the gene expression values

as before. There are a total of 96 samples available for this cancer disease. We have randomly divided 50% of them for training and 50% of them for testing purposes. With the same NSGA-II parameters as in the 3859-gene Leukemia case, we obtain a total of 144 solutions (Table 1). As few as five out of 4026 genes are enough to correctly identify the class all 96 samples. Interestingly, there are 121 such five-gene combinations to classify the task perfectly. Some other solutions with smaller gene subsets are also shown in the table.

3.7. Complete Colon data set

For the Colon disease, we have 62 samples each with 2000 genes. In this case, expression values for all genes are available. Thus, we simply log-transform and then normalize the expression values of all 2000 genes as described in Section 2.1. Next, we apply the multimodal NSGA-II on this data set. We have randomly chosen 50% samples for training and the rest for testing. Identical NSGA-II parameters to those used in the Leukemia case except a mutation probability of 0.001 are used here. A total of 39 solutions are obtained. Of them, 25 solutions contain seven genes and each can correctly identify all 31 training samples but misses to identify only one test sample. Table 1 shows these and other more mismatched solutions obtained using the multimodal NSGA-II.

Interestingly, in this data set, the NSGA-II fails to identify a single solution with 100% correct classification. Although we have chosen different combinations of 31 training and 31 test samples and rerun NSGA-II, no improvement to this solution is observed. In order to investigate if there exists at all any solution (classifier) which will correctly classify all training and test samples, we have used a single-objective, binary-coded genetic algorithm for minimizing the sum of mismatches in the training and test samples and without any care of minimizing the gene subset size. The resulting minimum solution corresponds to one of the 25 solutions found using the NSGA-II. The obtained classifier contains seven genes and with an overall mismatch of one sample. This study supports the best solution obtained using the multimodal NSGA-II discussed in the previous paragraph.

4. Classification with multiple training data sets

In the earlier study (Liu and Iba, 2002) and many studies related to the machine learning, the available data set is partitioned into two classes: training and test sets. In these cases, one particular training set is considered during the learning phase and a classifier is developed. However, we stress here the importance of using multiple training sets (may be of the same size) for generating a better classifier.

As with previous sections, we use only one particular training set for evaluating a classifier. For the Leukemia samples, this training set is already suggested and used in the literature, whereas for other two cases, no particular training set is pre-specified. For these two cases, we choose a training set randomly from all available samples. The rest are declared as the test set. However, in this study, we use H (we have used $H = 100$ in all cases) different training sets, instead of one. Thus, for evaluating a classifier consisting of a few genes, we follow the classification procedure used in the previous section for all H training sets independently. For each case j, we note the number of mismatches τ_{train}^j and τ_{test}^j in both training and test samples, respectively. Thereafter, we take an average of these mismatches and calculate the two objective values as follows:

$$f_2 = \frac{1}{H} \sum_{j=1}^{H} \tau_{\text{train}}^j, \quad f_3 = \frac{1}{H} \sum_{j=1}^{H} \tau_{\text{test}}^j.$$

Since a number of training cases are used during the development of the classifier, the obtained classifier is also expected to be more generic than those obtained in the previous sections. However, the development of such a classifier comes at the expense of more computational efforts. Since H different training cases are considered in the evaluation of one classifier, the computational time is expected to increase by at least H times.

4.1. Leukemia samples

The NSGA-II parameters are the same as before. Fig. 10 shows that 21 different four-gene classifiers are discovered by the NSGA-II.

All these classifiers make 100% correct classification to 100 different training and test sets. It is interesting to note that 21 classifiers needed a total of 27 distinct genes, of which three of them appear in more than 50% of the obtained 21 classifiers. These genes have accession numbers: L07633, U82759, and M27891. Of these three genes, the first and the third genes appear in all but one classifier. Based on this computational study, it can be concluded that these two genes are important in making the correct classification of the two Leukemia diseases. It is now a matter of investigation whether the same conclusion can be drawn from a biological viewpoint. However, it is remarkable that from 7219 genes available for the Leukemia samples, this computational approach is able to identify the two most important genes for further biological study.

4.2. Lymphoma samples

Next, we apply the multimodal NSGA-II with identical parameters as those used in Section11. Here, we have used $H = 100$ different training sets (randomly chosen from the available 96 samples). In this case, we obtain only six, 12-gene classifiers, each capable of making 100% correct classification on all 100 training and test data. Fig. 11 shows these six classifiers.

It can be observed that of the 12 genes, five of them (with accession numbers 2801X, 848X, 721X, 1610X, and 1636X) are common to all six classifiers and six other genes (with accession numbers 3093X, 2424X,

1483X, 1818X, 1639X, and 493X) appear in all but one classifier. These results strongly suggest that these 11 genes are responsible for making correct classification for the two-class Lymphoma cancer samples. Compared to the study with just one training set (in which five-gene classifiers were adequate), here 12-gene classifiers are required to have 100% classifications. With respect to the one training set study, we observe that there are only three genes (with accession numbers 2801X, 3955X, and 1639X) common between the two observed sets of classifiers.

4.3. Colon samples

With identical parameter settings to those used in Section 3.7, we apply NSGA-II to the complete
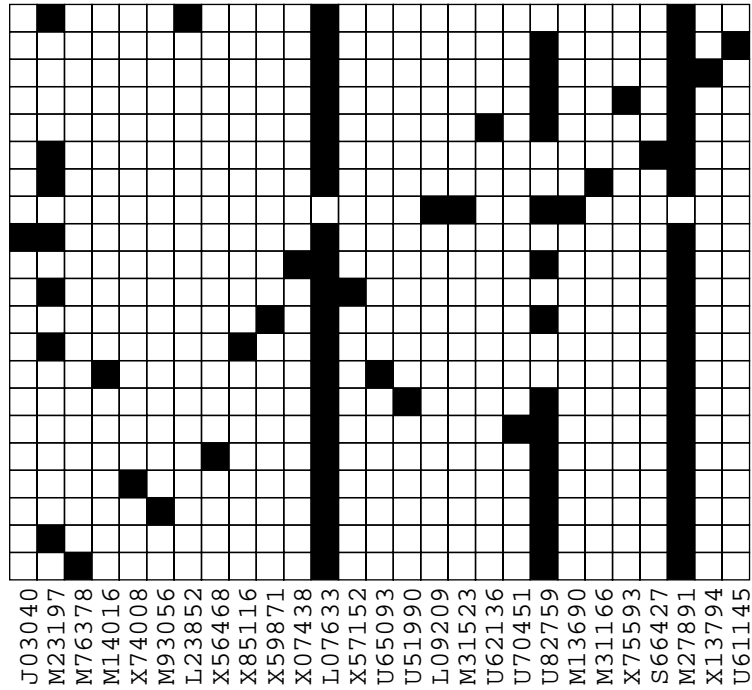
Fig. 10. Multimodal NSGA-II solutions (with 100% correct classification) for the complete Leukemia samples. Each row indicates a classifier having four genes.

Colon samples with $H = 100$ different training and test sets. In this case, no classifier with a 100% correct classification is found. However, the best-found classifier with 12 genes is able to make on an average 1.11 mismatches in 100 training sets (each having 31 samples) and 0.88 mismatches in 100 test sets (each having 31 samples). Recall that for the single training set case, a seven-gene classifier with only one mismatch in the test samples were found.
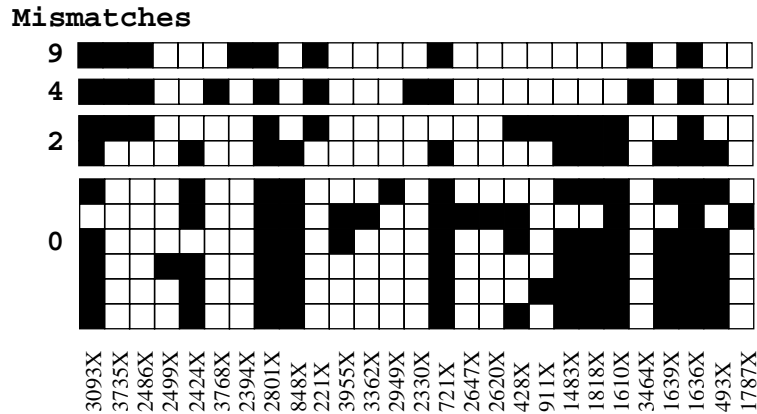

Fig. 11. Multimodal NSGA-II solutions for the complete Lymphoma samples. Each row in the lower set (marked as '0') indicates a 12-gene classifier capable of making 100% correct classifications. Other sets indicate classifiers making some classification error.

5. Classification with confidence

One of the difficulties with the above classification procedure is that the sign of the right term in Eq. (1) is checked to identify if a sample belongs to one class or another. For each (g) of the 50 genes in a particular Leukemia sample x, we have calculated the term (say the statistic $S(x, g)$) inside the summation in Eq. (1). For this sample, a correct prediction has been made. The statistic $S(x, g)$ values are plotted in Fig. 12.

It can be seen from the figure that for 27 genes, negative values of $S(x, g)$ emerged, thereby classifying individually that the sample belongs to AML (class B), whereas only 23 genes detects the sample to be ALL (class A). Eq. (1) finds the right side value to be 0.01, thereby declaring the sample to be an ALL sample (which is correct). But it has been argued elsewhere (Golub et al., 1999) that a correct prediction with such a small strength does not allow to make the classification with a reasonable confidence.

For a more confident classification, we may fix a prediction strength threshold θ and modify the classification procedure slightly. Let us define that the sum of the positive $S(x, g)$ values is S^A and the sum of the negative $S(x, g)$ values is S^B. Then, the prediction strength $|(S^A - S^B)/(S^A + S^B)|$ (Golub et al., 1999) is compared with θ. If it is more than θ, the classification is accepted, else the sample is considered to be undetermined for class A or B. For simplicity, we assume these undetermined samples to be identical to mismatched samples and include this sample to incre-

ment τ_1 or τ_2, as the case may be. This way, a 100% correctly classified gene subset will ensure that the prediction strength is outside $(-\theta, \theta)$. Fig. 12 illustrates this concept with $\theta = 0.3$ (30% threshold) and demands that a match will be scored only when the prediction strength falls outside the shaded area. In the sample illustrated in the figure, there are eight genes which will case a undetermined classification with less than 30% threshold, thereby causing this sample to be a 'mismatched' sample.

Here, we show the effect of classification with a threshold in prediction strength on the 50-gene Leukemia samples. All 72 samples are used in the study. With identical parameter settings as in Section 3.4, the multimodal NSGA-II with $\theta = 30\%$ threshold prediction strength finds a solution with four genes and one mismatch in the test samples. The smallest prediction strength observed in any sample is 32.1%. The sample on which the mismatch occurs, the prediction strength is 41.7%. Thus, the obtained four-gene subset does its best keeping the total mismatch down to one (in the test sample). Taking any more genes in the subset increases the number of mismatches. The multimodal NSGA-II has discovered two solutions having four genes and one mismatch in the test samples. These gene subsets are (X95735, M23197, X85116, M31211) and (X95735, M23197, X85116, M31523). Since these solutions have one mismatch in the test sample, they were not found to be non-dominated solutions in Section 3.4. However, some of the above genes are also found to be common to the frequently found classifier genes in Fig. 8.
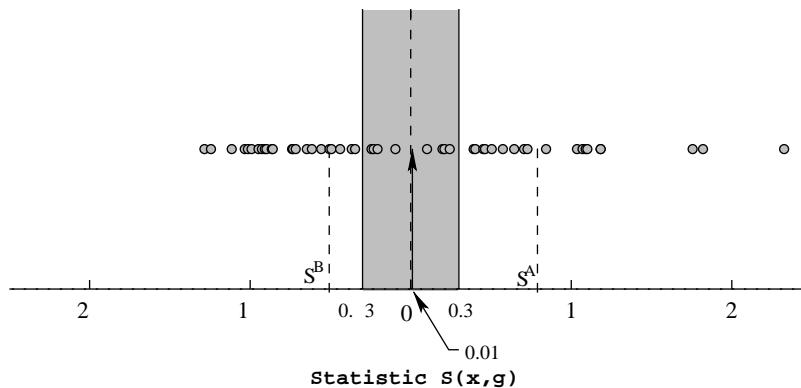

Fig. 12. The statistic $S(x, g)$ of a sample for the 50-gene Leukemia classifier. The mean statistic values for classes A and B are shown. Since the overall statistic is within the shaded region, the classifier is considered not being able to classify the sample to either class A or B.

However, if the threshold is reduced to $\theta = 20\%$, we obtain a solution with 100% correct classification with a gene subset of size five. In this case, there are eight samples in which the prediction strength is between 20 and 30%. Thus, while optimizing for the 30% threshold prediction strength, this five-gene solution was dominated by the four-gene solution and hence did not appear as one of the non-dominated solutions. Recall that the study presented in Section 3.4 used a prediction strength threshold of $\theta = 0\%$ and we obtained a 100% correct classification with only four genes. There are two five-gene solutions found with a 100% correct classification. They are (U05259, M31211, M23197, N96326, M83652) and (U05259, Y08612, M23197, X85116, M83652), which have three genes in common and share some common genes with the four-gene solutions found in the 30% threshold study.
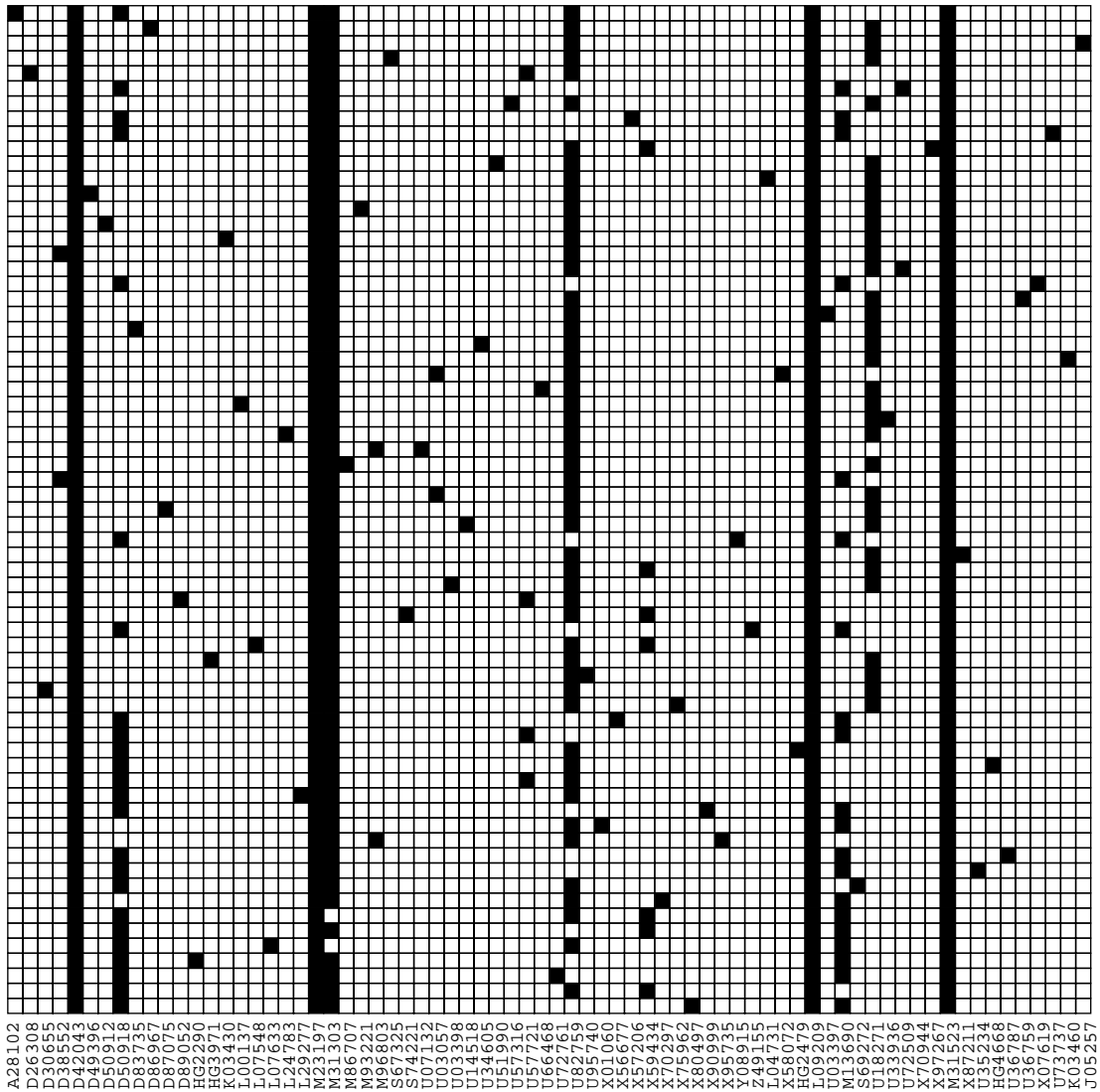

Fig. 13. Multimodal NSGA-II solutions for the complete Leukemia samples with prediction strength. Each row indicates an eight-gene classifier capable of making 100% correct classification with a prediction strength of 30%. The participating genes are marked in solid boxes.

This study shows that the outcome of the classification depends on the chosen prediction strength threshold. Keeping a higher threshold makes a more confident classification, but at the expense of some mismatches, while keeping a low threshold may make a 100% classification, but the classification may be performed with a poor confidence level.

5.1. Complete data with multiple training sets

The consideration of prediction strength into the development of an optimal classifier is important but involves additional computational effort. In the above subsection, we kept our discussion confined to a smaller search space with only 50 genes and to the use of a single training set. In this subsection, we present results with the completed filtered gene sets and with multiple training sets. We have used $H = 100$ different training sets. For space limitation, we discuss the results for only the Leukemia samples here.

With identical NSGA-II parameter setting, we find the optimal classifiers with 30% prediction strength. As many as 67 distinct solutions with 100% correct classification to all 100 sets of training and test samples are found. Fig. 13 shows these classifiers.

For this purpose, the number of genes required in the classifier is eight (compared to four genes required without the consideration of the prediction strength). Although four genes were enough to correctly classify earlier, a more confident classification requires a total of eight genes in the classifier. It is interesting to note that there are six genes which appear in more than half of the obtained classifiers. They have the following accession numbers: D42042, M23197, M31303, U82759, L09209, and M31523. Surprisingly, among the above genes, the first and fifth genes were not at all considered in the 50-gene study elsewhere (Golub et al., 1999). Here we discover that both of these genes appear in all 67 classifiers.

To investigate any similarity between the classifiers obtained with and without prediction strength, we compare the two sets of solutions. Sixty-seven different solutions obtained here involve a total of 72 different genes with a number of genes commonly appearing in 67 classifiers. The classifiers are shown in Fig. 13. It is observed that of the 72 genes, seven genes are common to those were found in the classifier list in Section 4.1. These common genes have the following

accession numbers: L07633, M23197, U51990, U82759, L09209, M13690, and M31523. It is interesting that the gene L07633 which appeared in 20 of 21 classifiers in the study without prediction strength appears only once among 67 classifiers with prediction strength consideration. Since the consideration of 30% prediction strength is more reliable than the study without the prediction strength, the study suggests that the presence of the six genes abundantly found in 67 classifiers is collectively important for making a correct classification of the two Leukemia disease samples. Similar results are also obtained for the other two cancer data sets and are omitted here for space restrictions.

6. Conclusions

The identification of small gene subsets responsible for classifying available disease samples to fall in one category or another has been addressed here. By treating the resulting optimization problem with two or more objectives, we have applied a multiobjective evolutionary algorithm (NSGA-II) and a number of its variants to find optimal gene-subsets for different classification tasks to three openly available cancer data sets: Leukemia, Lymphoma, and Colon. Compared to past studies, we have used multiple training/testing sets for the classification purpose to obtain a more general classifier. One remarkable finding is that compared to past studies, our study has discovered classifiers having only three or four genes, which can perfectly classify all Leukemia and Lymphoma samples. For the Colon cancer samples, we have found a classifier with all except one correctly classified samples. Different variants of NSGA-II have exhibited the flexibility with which such identifications can be made.

We have also suggested a *multimodal* NSGA-II by which multiple, multimodal, non-dominated solutions can be found simultaneously in one single simulation run. These solutions have identical objective values but they differ in their phenotypes. This study has shown that in the gene subset identification problem there exist a large number of such multimodal solutions, even corresponding to the 100% correctly classified gene subsets. In the Leukemia data set, the multimodal NSGA-II has discovered as many as 352

different three-gene combinations which correctly classified all 72 samples. Investigating these multiple solutions may provide intriguing information about crucial gene combinations responsible for classifying samples into one class or another. In the Leukemia data set, we have observed that among all 50 genes there are three genes which appear frequently on 26 different four-gene solutions capable of perfectly classifying all 72 samples. Similar such studies have also been performed on the complete Leukemia, Lymphoma and Colon cancer samples and similar observation about frequently appearing gene combinations have been obtained. Investigating these frequently appearing genes further from a biological point of view should provide crucial information about the causes of different classes of a disease. The proposed multimodal NSGA-II is also generic and can be applied to similar other multiobjective optimization problems in which multiple multimodal solutions are desired.

Finally, a more confident and reliable gene subset identification task is performed by using a minimum threshold in the prediction strength (Golub et al., 1999). With a 30% threshold, it has been found that four extra genes are needed to make a 100% correct classification compared to that needed with the 0% threshold. Surprisingly, there exist not many genes common between the two classifiers obtained with and without prediction strength. Although the genes commonly appearing in the two cases must be immediately investigated for their biological significance, this study clearly brings out the need for a collaborative effort between a computer algorithmist and a biologist in achieving a more coherent and meaningful classification task. What is importantly obtained in this study is a flexible and efficient evolutionary search procedure based on a multiobjective formulation of the gene subset identification problem for achieving a reliable classification task.

Acknowledgements

References

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

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In: Proceedings of National Academy of Science. Cell Biology, vol. 96. pp. 6745–6750.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., Yakhini, Z., 2000. Tissue classification with gene expression profiles. J. Comput. Biol. 7, 559–583.

Clarke, P.A., George, M., Cunningham, D., Swift, I., Workman, P., 1999. Analysis of tumor gene expression following chemotherapeutic treatment of patients with bowl cancer. In: Proceedings of Nature Genetics Microarray Meeting-99. p. 39.

Deb, K., 2001. Multi-objective Optimization Using Evolutionary Algorithms. Chichester, UK: Wiley.

Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evolut. Comput. 6 (2), 182–197.

Gershon, D., 2002. Microarray technology an array of opportunities. Nature 416, 885–891.

Goldberg, D.E., 1989. Genetic Algorithms for Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley.

Goldberg, D.E., Richardson, J., 1987. Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications. pp. 41–49.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S., 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286, 531–537.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artificial Intelligence Journal Special Issue on Relevance 97, 234–271.

Liu, J., and Iba, H., 2002. Selecting informative genes using a multiobjective evolutionary algorithm. In: Proceedings of the World Congress on Computational Intelligence (WCCI-2002). pp. 297–302.

Liu, J., Iba, H., Ishizuka, M., 2001. Selecting informative genes with parallel genetic algorithms in tissue classification. Genome Informatics 12, 14–23.

Miettinen, K., 1999. Nonlinear Multiobjective Optimization. Kluwer, Boston.