Evolutionary Rough Feature Selection in Gene Expression Data

Mohua Banerjee, Sushmita Mitra, *Senior Member, IEEE*, and Haider Banka

***Abstract*—An evolutionary rough feature selection algorithm is used for classifying microarray gene expression patterns. Since the data typically consist of a large number of redundant features, an initial redundancy reduction of the attributes is done to enable faster convergence. Rough set theory is employed to generate reducts, which represent the minimal sets of nonredundant features capable of discerning between all objects, in a multiobjective framework. The effectiveness of the algorithm is demonstrated on three cancer datasets.**

***Index Terms*—Bioinformatics, feature selection, genetic algorithms (GAs), microarray data, rough sets, reduct generation, soft computing.**

I. Introduction

COMPUTATIONAL molecular biology is an interdisciplinary subject involving fields as diverse as biology, computer science, information technology, mathematics, physics, statistics, and chemistry. Aspects of this subject that relate to information science are the focus of bioinformatics [1], [2]. One needs to analyze and interpret the vast amount of data that are available, involving the decoding of around 24 000–30 000 human genes. Specifically, high-dimensional feature selection is important for characterizing gene expression data involving many attributes—indicating that data mining methods hold promise in this direction.

Unlike a genome, which provides only static sequence information, microarray experiments produce gene expression patterns that provide dynamic information about cell function. This information is useful while investigating complex interactions within the cell. For example, data mining methods can ascertain and summarize the set of genes responding to a certain level of stress in an organism [1]. Microarray technologies have been utilized to evaluate the level of expression of thousands of genes in colon, breast, and blood cancer classification [3]–[6] as well as clustering [7], [8].

In addition to the combinatorial approach for solutions, there also exists scope for soft computing; especially, for generating low-cost, low-precision, good solutions. *Soft computing* is a consortium of methodologies that works synergistically and provides flexible information-processing capability for handling real-life ambiguous situations [9]. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions. Recently, various soft computing methodologies (such as fuzzy logic, neural networks, genetic algorithms (GAs), and rough sets) have been applied to handle the different challenges posed by data mining [10], involving large heterogeneous datasets.

One of the important problems in extracting and analyzing information from large databases is the associated high complexity. Feature selection is helpful as a preprocessing step for reducing dimensionality, removing irrelevant data, improving learning accuracy, and enhancing output comprehensibility. There are two basic categories of feature selection algorithms, viz., filter and wrapper models. The filter model selects feature subsets independently of any learning algorithm and relies on various measures of the general characteristics of the training data. The wrapper model uses the predictive accuracy of a predetermined learning algorithm to determine the goodness of the selected subsets and is computationally expensive. Use of fast-filter models for the efficient selection of features, based on correlation for relevance and redundancy analysis, has been reported in literature [11], [12] for high-dimensional data.

Microarray data are a typical example presenting an overwhelmingly large number of features (genes), the majority of which are not relevant to the description of the problem and could potentially degrade the classification performance by masking the contribution of the relevant features. The key informative features represent a base of reduced cardinality for subsequent analysis aimed at determining their possible role in the analyzed phenotype. This highlights the importance of feature selection with particular emphasis on microarray data. Recent approaches in this direction include probabilistic neural networks [13], support vector machines [14], neuro-fuzzy computing [15], and neuro-genetic hybridization [16].

Rough set theory [17] provides an important and mathematically established tool for this sort of dimensionality reduction in large data. A basic issue addressed in relation to many practical applications of knowledge databases is the following. An information system consisting of a domain U of objects/observations and a set A of attributes/features induces a partitioning (classification) of U by A. A block of the partition would contain those objects of U that share identical feature values, i.e., they are *indiscernible* with respect to the given set A of features. But the whole set A may not always be necessary to define the classification/partition of U. Many of the attributes may be superfluous, and we may find the *minimal* subsets of attributes, which give the same classification as the whole set A. These subsets are called *reducts* in rough set theory. In terms of feature selection, therefore, reducts correspond to the *minimal feature sets* that are *necessary* and *sufficient* to represent a *correct* decision about the classification of the domain. One is thus provided

Manuscript received January 5, 2005; revised September 9, 2005. This work was supported in part by the Council of Scientific and Industrial Research research under Grant 22/0346/02/EMR-II. This paper was recommended by Associate Editor Y. Jin.

M. Banerjee is with the Department of Mathematics and Statistics, Indian Institute of Technology, Kanpur 208016, India (e-mail: mohua@iitk.ac.in).

S. Mitra and H. Banka are with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India (e-mail: sushmita@isical.ac.in; hbanka_r@isical.ac.in).

Digital Object Identifier 10.1109/TSMCC.2007.897498

with another angle of addressing the problem of dimensionality, based on the premise that the initial set of features may render objects of the domain indiscernible, due to lack of complete information.

The task of finding reducts is reported to be nondeterministic polynomial time (NP)-hard [18]. The high complexity of this problem has motivated investigators to apply various approximation techniques to find near-optimal solutions [19]. This includes work on computing reducts, specifically for feature selection [19], [20]. Zhong [21], for instance, proposes a greedy heuristics and applies it on cancer (not microarray) data. Others have mainly based their methods on the filter or the wrapper approach. Apart from these, there are some studies reported, e.g., [22] and [23], where GAs [24] have been applied to find reducts.

GAs provide an efficient search technique in a large solution space based on the theory of evolution. It involves a set of evolutionary operators such as selection, crossover, and mutation. A population of chromosomes is made to evolve over generations by optimizing a fitness function, which provides a quantitative measure of the fitness of individuals in the pool. When there are two or more conflicting characteristics to be optimized, often the single-objective GA requires an appropriate formulation of the single fitness function in terms of an additive combination of the different criteria involved. In such cases, *multiobjective* GAs (MOGAs) [25] provide an alternative, more efficient, approach in searching for the optimal solutions.

Each of the studies in [22] and [23] employs a single-objective function to obtain reducts. The essential properties of a reduct are: 1) to classify among all elements of the universe with the same accuracy as the starting attribute set simultaneously and 2) to be of small cardinality. A close observation reveals that these two characteristics are of a conflicting nature. Hence the determination of reducts is better represented as a two-objective optimization problem. This idea was first mooted in [26], and a preliminary study was conducted.

In the present paper, we consider microarray data consisting of three sets of two-class cancer samples. Since such data typically contains a large number of features, most of which are not relevant, an initial redundancy reduction is done on the (attribute) expression values. The idea is to retain only *those* genes that play a major role in arriving at a decision about the output classes. This preprocessing aids faster convergence, mainly because the initial population is now located nearer to the optimal solution in the huge search space. Reducts or the minimal features are then generated from these reduced sets using MOGA. Among the different multiobjective algorithms, it is observed that nondominated sorting genetic algorithm (NSGA-II) [27] has the features required for a good MOGA. NSGA-II is adapted here to handle large datasets more effectively.

Section II describes the relevant preliminaries on rough set theory, MOGAs, and microarray gene expression data. We assume that the readers are sufficiently familiar with the basics of classical GA [24], and hence, do not go into its details here. The redundancy reduction to better handle the high-dimensional data, the basic notions used in the evolutionary rough feature selection algorithm, and the algorithm itself are described in Section III. The performance of the algorithm is demonstrated in Section IV on microarray gene expression data from bioinformatics involving very high-dimensional attributes. Comparative study and analysis of the results are also included. Finally, Section V concludes the paper.

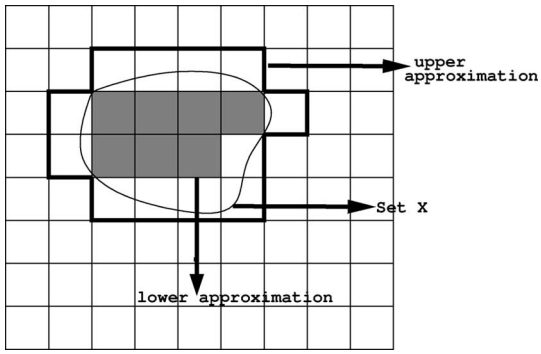


Fig. 1. Lower and upper approximations of a rough set.

II. Preliminaries

In this section, we briefly discuss the basic concepts of rough set theory, MOGAs, and microarray gene expression data.

A. Rough Set Theory

Rough sets [17] constitute a major mathematical tool for managing uncertainty that arises from granularity in the domain of discourse—due to incomplete information about the objects of the domain. The granularity is represented formally in terms of an *indiscernibility* relation that partitions the domain. If there is a given set of *attributes* ascribed to the objects of the domain, objects having the same attribute values would be indiscernible and would belong to the same block of the partition. The intention is to approximate a *rough* (imprecise) concept in the domain by a pair of *exact* concepts. These exact concepts are called the lower and upper approximations and are determined by the indiscernibility relation. The lower approximation is a set of objects definitely belonging to the rough concept, whereas the upper approximation is a set of objects possibly belonging to the same. Fig. 1 shows a rough set with its approximations. The formal definitions of the aforementioned notions and others required for the present work are given as follows.

Definition 1: An information system $\mathcal{A} = (U, A)$ consists of a nonempty, finite set U of objects (cases, observations, etc.) and a non-empty, finite set A of attributes a (features, variables), such that $a : U \rightarrow V_a$, where V_a is a value set. We shall deal with information systems called decision tables, in which the attribute set has two parts ($A = C \cup D$) consisting of the *condition* and *decision* attributes (in the subsets C, D of A, respectively). In particular, the decision tables we take will have a single decision attribute d and will be *consistent*, i.e., whenever objects x, y are such that for each condition attribute $a, a(x) = a(y)$, then $d(x) = d(y)$.

Definition 2: Let $B \subset A$. Then a B-indiscernibility relation IND(B) is defined as

$$\text{IND}(B) = \{(x, y) \in U : a(x) = a(y), \qquad \forall a \in B\}. \quad (1)$$

It is clear that IND(B) partitions the universe U into equivalence classes

$$[x_i]_B = \{x_j \in U : (x_i, x_j) \in \text{IND}(B)\}, \qquad x_i \in U. \quad (2)$$

Definition 3: The B-lower and B-upper approximations of a given set $X(\subseteq U)$ are defined, respectively, as follows:

$\underline{B}X = \{x \in U : [x]_B \subseteq X\}$

$\overline{B}X = \{x \in U : [x]_B \cap X \neq \phi\}$.

The B-boundary region is given by $BN_B(X) = \overline{B}X \setminus \underline{B}X$.

1) Reducts: In a decision table $\mathcal{A} = (U, C \cup D)$, one is interested in eliminating redundant *condition* attributes and actually *relative* (D)-reducts are computed.

Let $B \subseteq C$, and consider the *B-positive region* of D, viz., $\mathrm{POS}_B(D) = \bigcup_{[x]_D} \underline{B}[x]_D$. An attribute $b \in B(\subseteq C)$ is D-dispensable in B if $\mathrm{POS}_B(D) = \mathrm{POS}_{B\setminus\{b\}}(D)$, otherwise b is D-indispensable in B. Here, B is said to be D-independent in $\mathcal{A}$, if every attribute from B is D-indispensable in B.

Definition 4: $B(\subseteq C)$ is called a D-reduct in $\mathcal{A}$, if B is D-independent in $\mathcal{A}$, and $\mathrm{POS}_C(D) = \mathrm{POS}_B(D)$.

Notice that, as decision tables with a single decision attribute d are taken to be consistent, $U = \mathrm{POS}_C(d) = \mathrm{POS}_B(D)$, for any d-reduct B.

2) Discernibility Matrix: D-reducts can be computed with the help of D-discernibility matrices [18]. Let $U = \{x_1, \cdots, x_m\}$. A D-discernibility matrix $M_D(\mathcal{A})$ is defined as an $m \times m$ matrix of the information system $\mathcal{A}$ with the (i, j)th entry c_{ij} given by

$$c_{ij} = \{a \in C : a(x_i) \neq a(x_j), \text{ and } (x_i, x_j) \notin \mathrm{IND}(D)\}, \quad i, j \in \{1, \cdots, m\}. \quad (3)$$

A variant of the discernibility matrix, viz., *distinction table* [22] is used in our paper to enable faster computation.

Definition 5: A distinction table is a *binary* matrix with dimensions $\frac{(m^2-m)}{2} \times N$, where N is the number of attributes in A. An entry $b((k, j), i)$ of the matrix corresponds to the attribute a_i and pair of objects (x_k, x_j) and is given by

$$b((k, j), i) = \begin{cases} 1, & \text{if } a_i(x_k) \neq a_i(x_j) \\ 0, & \text{if } a_i(x_k) = a_i(x_j). \end{cases} \quad (4)$$

The presence of a "1" signifies the ability of the attribute a_i to discern (or distinguish) between the pair of objects (x_k, x_j).

B. MOGA

Most real-world search and optimization problems typically involve multiple objectives. A solution that is better with respect to one objective requires a compromise in other objectives. Let us consider the decision-making problem regarding the purchase of a car. It is expected that an inexpensive car is likely to be less comfortable. If a buyer is willing to sacrifice cost to some extent, (s)he can find another car with a better comfort level than the cheapest one. Thus, in problems with more than one conflicting objective, there exists no single optimum solution. Rather, there exists a set of solutions, which are all optimal involving tradeoffs between conflicting objectives. For example, the various factors to be optimized in the problem of buying a car include the total finance available, distance to be driven each day, number of passengers riding in the car, fuel consumption and cost, depreciation value, road conditions where the car will be mostly driven, physical health of the passengers, social status, etc.

Unlike single-objective optimization problems, the MOGA tries to optimize two or more conflicting characteristics represented by fitness functions. Modeling this situation with a single-objective GA would amount to a heuristic determination of a number of parameters involved in expressing such a scalar-combination-type fitness function. MOGA, on the other hand, generates a set of *Pareto-optimal* solutions [25], which simultaneously optimize the conflicting requirements of the multiple fitness functions.

Among the different multiobjective algorithms, it is observed that NSGA-II [27] possesses all the features required for a good MOGA. It has been shown that this can converge to the global Pareto front, while simultaneously maintaining the diversity of the population. We describe here the characteristics of NSGA-II such as nondomination, crowding distance, and the crowding selection operator. This is followed by the actual algorithm.

1) Nondomination: The concept of optimality, behind the multiobjective optimization, deals with a set of solutions. The conditions for a solution to be *dominated* with respect to the other solutions are given as follows.

Definition 6: If there are M objective functions, a solution $x^{(1)}$ is said to *dominate* another solution $x^{(2)}$, if both conditions 1 and 2 are true.

1) The solution $x^{(1)}$ is *no worse* than $x^{(2)}$ in *all* the M objective functions.
2) The solution $x^{(1)}$ is *strictly better* than $x^{(2)}$ in *at least one* of the M objective functions.

Otherwise, the two solutions are *nondominating* to each other. When a solution i dominates a solution j, then rank $r_i < r_j$.

The major steps for finding the nondominated set in a population P of size $|P|$ are outlined as follows.

Step 1) Set solution counter $i = 1$ and create an empty nondominated set P'.
Step 2) **For** a solution $j \in P(j \neq i)$, check if the solution j dominates the solution i. **If** yes **then go to** step 4).
Step 3) **If** more solutions are left in P, increment j by one and **go to** step 2). **Else** set $P' = P' \cup \{i\}$.
Step 4) Increment i by one. **If** $i \leq |P|$ **then go to** step 2). **Else** declare P' as the nondominated set.

After all the solutions of P are checked, the members of P' constitute the nondominated set at the first level (front with rank $= 1$). In order to generate solutions for the next higher level (dominated by the first level), the aforementioned procedure is repeated on the reduced population $P = P - P'$. This is iteratively continued until $P = \emptyset$.

2) Crowding Distance: In order to maintain diversity in the population, a measure called crowding distance is used. This assigns the highest value to the boundary solutions and the average distance of two solutions [$(i + 1)$th and $(i - 1)$th] on either side of solution i along each of the objectives. The following algorithm computes the crowding distance d_i of each point in the front $\mathcal{F}$.

1) Let the number of solutions in $\mathcal{F}$ be $l = |\mathcal{F}|$ and assign $d_i = 0$ for $i = 1, 2, \ldots, l$.
2) **For** each objective function $f_k, k = 1, 2, \ldots, M$, sort the set in its worse order.
3) Set $d_1 = d_l = \infty$.
4) **For** $j = 2$ to $(l - 1)$ increment d_j by $f_{k_{j+1}} - f_{k_{j-1}}$.

3) Crowding Selection Operator: Crowded tournament selection operator is defined as follows. A solution i wins tournament with another solution j if any one of the following is true.

- Solution i has better rank, i.e., $r_i < r_j$.
- Both the solutions are in the same front, i.e., $r_i = r_j$, but solution i is less densely located in the search space, i.e., $d_i < d_j$.

4) NSGA-Ii: The multiobjective algorithm NSGA-II is characterized by the use of the aforementioned three characteristics while generating the optimal solution. Let us now outline the main steps of NSGA-II [27].

1) Initialize the population randomly.
2) Calculate the multiobjective fitness function.
3) Rank the population using the dominance criteria of Section II-B1.
4) Calculate the crowding distance based on Section II-B2.
5) Do selection using crowding selection operator of Section II-B3.
6) Do crossover and mutation (as in the conventional GA) to generate children population.
7) Combine parent and children population.
8) Replace the parent population by the best members of the combined population. Initially, members of lower fronts replace the parent population. When it is not possible to accommodate all the members of a particular front, then that front is sorted according to the crowding distance. Selection of individuals is done on the basis of higher crowding distance. The number selected is that required to make the new parent population size the same as that of the old one.

C. Microarray and Gene Expression Data

Microarrays are used in the medical domain to produce molecular profiles of diseased and normal tissues of patients. Such profiles are useful for understanding various diseases and aid in more accurate diagnosis, prognosis, treatment planning, as well as drug discovery.

DNA microarrays (gene arrays or gene chips) [1] usually consist of thin glass or nylon substrates containing specific DNA gene samples spotted in an array by a robotic printing device. Researchers spread fluorescently labeled m-RNA from an experimental condition onto the DNA gene samples in the array. This m-RNA binds (hybridizes) strongly with some DNA gene samples and weakly with others, depending on the inherent double helical characteristics. A laser scans the array and sensors to detect the fluorescence levels (using red and green dyes), indicating the strength with which the sample expresses each gene.

The logarithmic ratio between the two intensities of each dye is used as the gene expression data. The relative abundance of the spotted DNA sequences in a pair of DNA or RNA samples is assessed by evaluating the differential hybridization of the two samples to the sequences on the array. Gene expression levels can be determined for samples taken: 1) at multiple time instants of a biological process (different phases of cell division) or 2) under various conditions (tumor samples with different histopathological diagnosis). Each sample corresponds to a high-dimensional row vector of its gene expression profile.

III. Evolutionary Reduct Generation

Over the past few years, there has been a good amount of study in effectively applying GAs to find reducts. We describe here the reduct generation procedure, incorporating initial redundancy reduction, in a multiobjective framework. NSGA-II is adapted to handle large datasets more effectively. We focus our analysis to two-class problems.

A. Redundancy Reduction for Microarray Data

Gene expression data typically consist of a small number of samples with a very large number of features of which many are redundant. We consider here two-class problems, particularly, diseased and normal samples, or two varieties of diseased samples. In other words, there is a single decision attribute d having only two members in its value set V_d. We first do a redundancy reduction on the (attribute) expression values to retain only *those* genes that play a highly decisive role in choosing in favor of either output class. Note that this preprocessing phase is a simple, fast, heuristic thresholding with the objective of generating an initial crude redundancy reduction among features. Subsequent reduct generation with MOGA (as explained in Sections III-B–C) determines the actual, refined minimal feature sets that are necessary and sufficient to represent a correct classification decision.

Normalization leads to scaling of intensities, thereby enabling the comparison of expression values between different microarrays within an experiment. *Preprocessing* aims at eliminating the ambiguously expressed genes (neither too high nor too low) as well as the constantly expressed genes across the tissue classes. During reduct generation, we select an *appropriate minimal* set of differentially expressed genes, across the classes, for subsequent efficient classification.

1) Attributewise normalization by

$$a'_j(x_i) = \frac{a_j(x_i) - \min_j}{\max_j - \min_j}, \quad \forall i \tag{5}$$

where $\max_j$ and $\min_j$ correspond to the maximum and minimum gene expression values for attribute a_j over all samples. This constitutes the normalized gene dataset, i.e., (continuous) attribute value table.

2) Choose thresholds Th_i and Th_f, based on the idea of quantiles [10]. Let the N patterns be sorted in the ascending order of their values along the jth axis. In order to determine the partitions, we divide the measurements into a number of small class intervals of equal width δ and count the corresponding class frequencies fr_c. The position of the kth partition value ($k = 1, 2, 3$ for four partitions) is calculated as

$$\mathrm{Th}_k = l_c + \frac{R_k - \mathrm{cfr}_{c-1}}{fr_c} * \delta \tag{6}$$

where l_c is the lower limit of the cth class interval, $R_k = \frac{N*k}{4}$ is the rank of the kth partition value, and cfr_{c-1} is the cumulative frequency of the immediately preceding class interval such that $\mathrm{cfr}_{c-1} \leq R_k \leq \mathrm{cfr}_c$. Here, we use $\mathrm{Th}_i = \mathrm{Th}_1$ and $\mathrm{Th}_f = \mathrm{Th}_3$.

3) Convert the attribute value table to binary (0/1) form as follows:

$$\textbf{If } a'(x) \leq \mathrm{Th}_i \textbf{ Then put } \text{“0”},$$
$$\textbf{Else If } a'(x) \geq \mathrm{Th}_f \textbf{ Then put } \text{“1”},$$
$$\textbf{Else put } \text{“}*\text{”}(\mathrm{don't\ care}).$$

TABLE I
USAGE DETAILS OF THE TWO-CLASS MICROARRAY DATA

Data used	# Attributes	Classes	# Samples
Colon	2000	Colon cancer	40
		Normal	22
Lymphoma	4026	Other type	54
		B-cell lymphoma	42
Leukemia	7129	ALL	47
		AML	25

4) Find the average occurrences of "*" over the entire attribute value table. Choose this as the threshold Th_a.
5) Remove from the table those attributes for which the number of "*'s" are $\geq \mathrm{Th}_a$. This is the *modified* (reduced) *attribute value table* $\mathcal{A}_r$.

B. d-Distinction Table

For a decision table $\mathcal{A}$ with N condition attributes and a single decision attribute d, the problem of finding a d-reduct is equivalent to finding a minimal subset of columns $R(\subseteq \{1, 2, \cdots, N\})$ in the distinction table [cf. Section II, Definition 5, (4)], satisfying

$$\forall(k,j)\exists i \in R : b((k,j),i) = 1, \quad \text{whenever } d(x_k) \neq d(x_j).$$

So, in effect, we may consider the distinction table to consist of N columns, and rows corresponding to only those object pairs (x_k, x_j) such that $d(x_k) \neq d(x_j)$. Let us call this shortened distinction table, a *d-distinction table*. Note that, as $\mathcal{A}$ is taken to be consistent, there is no row with all zero entries in a d-distinction table.

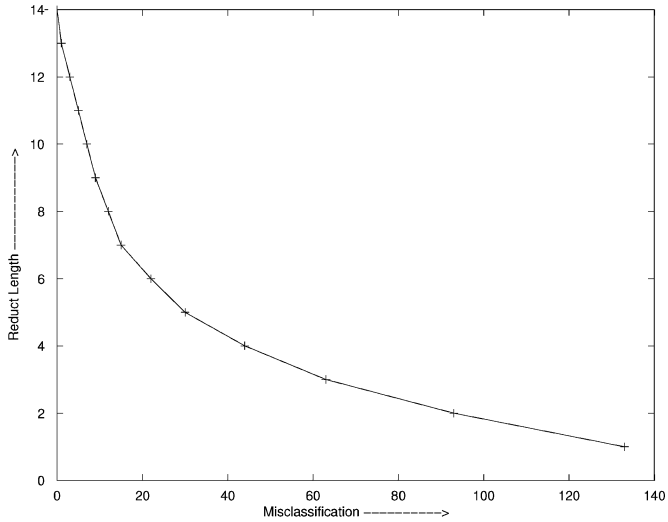
Accordingly, to find d-reducts in the present case, the reduced attribute value table $\mathcal{A}_r$ (as obtained in Section III is used for generating the d-distinction table. As mentioned earlier, d has the two output classes as the only members in its value set V_d.

- As object pairs corresponding to the same class do not constitute a row of the d-distinction table, there is a considerable reduction in its size, thereby leading to a decrease in computational cost.
- Additionally,
  If either of the objects in a pair has "*" as an entry under an attribute in table $\mathcal{A}_r$
  Then in the distinction table, put "0" at the entry for that attribute and pair.
- The entries "1" in the matrix correspond to the attributes of interest for arriving at a classification decision.

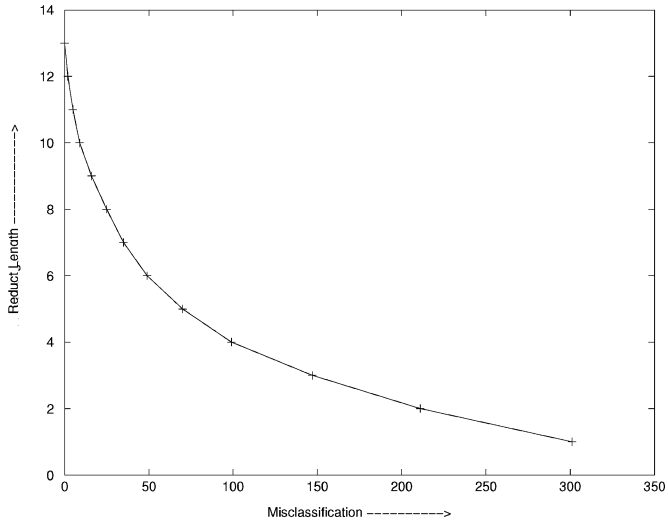
Let the number of objects initially in the two classes be m_1 and m_2, respectively. Then, the number of rows in the d-distinction table becomes $(m_1 * m_2) < \frac{m*(m-1)}{2}$, where $m_1 + m_2 = m$. This reduces the complexity of fitness computation to $O(N * m_1 * m_2)$.

C. Using MOGA

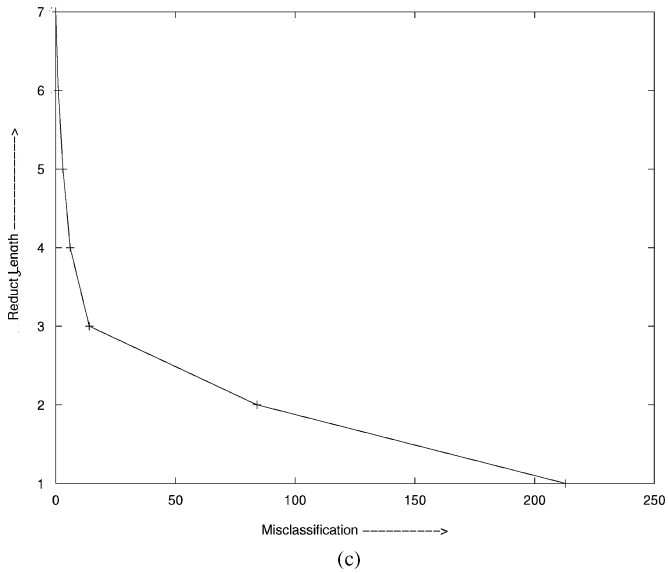
Algorithms reported in literature, e.g., in [22] and [23], vary more or less in defining the fitness function and typically use combined single-objective functions. Upon closely observing the nature of the reduct, we find that one needs to concentrate on generating a minimal set of attributes that are necessary and sufficient in order to arrive at an acceptable (classification)



(a)



(b)



(c)

Fig. 2. Pareto optimal front, with a certain random seed, for (a) colon, (b) lymphoma, and (c) leukemia datasets.

TABLE II
CLASSIFICATION OF GENE EXPRESSION TEST DATA WITH SELECTED LOW-CARDINALITY SOLUTIONS FROM THE BEST FRONT

Dataset	Population size	No. of attributes	k-nearest neighbors classification (%) on test set											
			$k=1$			$k=3$			$k=5$			$k=7$		
			C1	C2	Net	C1	C2	Net	C1	C2	Net	C1	C2	Net
Colon:	50	10	80.0	90.9	83.9	75.0	90.9	80.6	75.0	81.8	77.4	75.0	72.7	74.2
# Genes 2000	100	9	90.0	90.9	90.3	90.0	90.9	90.3	90.0	81.8	87.1	90.0	63.6	80.6
Reduce to 1102	200	8	85.0	90.9	87.1	90.0	81.8	87.1	90.0	90.9	90.3	95.0	81.8	90.3
	300	8	75.0	72.7	74.2	80.0	72.7	77.4	80.0	63.6	74.2	80.0	63.6	74.2
Lymphoma:	50	2	92.6	90.5	91.7	96.3	95.2	95.8	96.3	95.2	95.8	96.3	95.2	95.8
# Genes 4026	100	3	92.6	90.5	91.7	96.3	95.2	95.8	96.3	95.2	95.8	96.3	95.2	95.8
Reduce to 1867	200	3	96.3	90.5	93.8	96.3	95.2	95.8	96.3	95.2	95.8	96.3	95.2	95.8
	300	2	92.6	90.5	91.7	96.3	95.2	95.8	96.3	95.2	95.8	96.3	95.2	95.8
Leukemia:	50	3	100.0	85.7	94.1	100.0	78.6	91.2	100.0	78.6	91.2	100.0	71.4	88.2
# Genes 7129	100	3	100.0	78.6	91.2	95.0	85.7	91.2	100.0	78.6	91.2	100.0	78.6	91.2
Reduce to 3783	150	2	90.0	71.4	82.4	90.0	100.0	94.1	90.0	85.7	88.2	90.0	85.7	88.2
	180	2	95.0	71.4	85.3	100.0	71.4	88.2	100.0	71.4	88.2	100.0	71.4	88.2

decision. These two characteristics of reducts, being conflicting to each other, are well-suited for multiobjective modeling. This idea was explored and a preliminary study using simple datasets was done in [26]. In order to optimize the pair of conflicting requirements, the fitness function of [22] was split in a two-objective GA setting. We use these two objective functions in this paper in a modified form.

The reduct candidates are represented by binary strings of length N, where N is the number of condition attributes. In the bit representation, a "1" implies that the corresponding attribute is present while "0" means that it is not. So, if there are three attributes a_1, a_2, a_3 (i.e., $N = 3$), $\vec{\nu} = (1, 0, 1)$ in the search space of the GA would actually indicate the reduct candidate $\{a_1, a_3\}$. As we are looking for the *minimal* nonredundant attribute sets, an objective then is to obtain a minimal number of 1's in a solution. We note that a reduct is a minimal set of attributes that *discerns between all objects*(4). Now, $\vec{\nu}$ would discern between an object pair (k, j) (say), provided at least one of the attributes present in $\vec{\nu}$ assigns a 1 to the pair, i.e., in the d-distinction table $b((k, j), i) = 1$ for some a_i in $\vec{\nu}$. Thus, the second objective is to maximize the number of such object pairs for a solution.

Accordingly, two fitness functions f_1 and f_2 are considered for each individual. We have

$$f_1(\vec{\nu}) = \frac{N - L_{\vec{\nu}}}{N} \tag{7}$$

$$f_2(\vec{\nu}) = \frac{C_{\vec{\nu}}}{(m^2 - m)/2} \tag{8}$$

where $\vec{\nu}$ is the reduct candidate, $L_{\vec{\nu}}$ represents the number of 1's in $\vec{\nu}$, m is the number of objects, and $C_{\vec{\nu}}$ indicates the number of object combinations $\vec{\nu}$ can discern between. The fitness function f_1 gives the candidate credit for containing less attributes (fewer 1's), while the function f_2 determines the extent to which the candidate can discern among objects.

Thus, by generating a reduct, we are focusing on that minimal set of attributes, which can essentially distinguish between all patterns in the given set. In this manner, a reduct is mathematically more meaningful as the most appropriate set of nonredundant features selected from a high-dimensional data.

Crowding binary tournament selection of Section II-B3 is used. One-point crossover is employed with probability $p_c = 0.7$. Probability p_m of mutation on a single position of individual was taken as 0.05. Mutation of one position means replacement of "1" by "0," or "0" by "1." The probability values were chosen after several experiments.

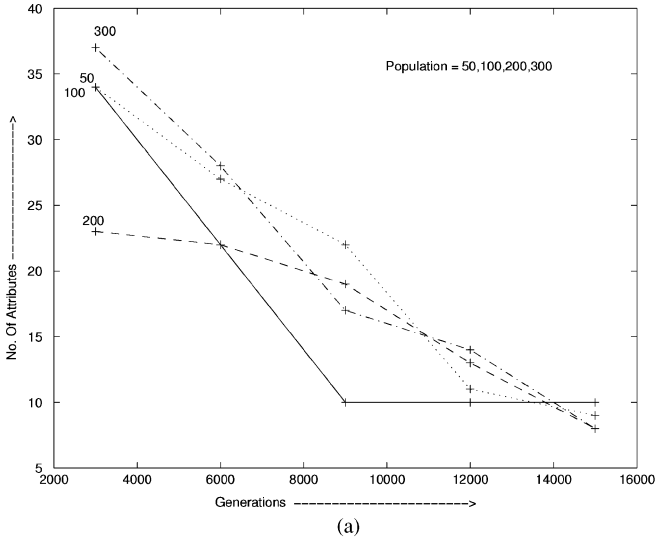
D. Algorithm

In this paper, NSGA-II is modified to effectively handle large datasets. Since we are interested in interclass distinction, the fitness function of (8) is modified as

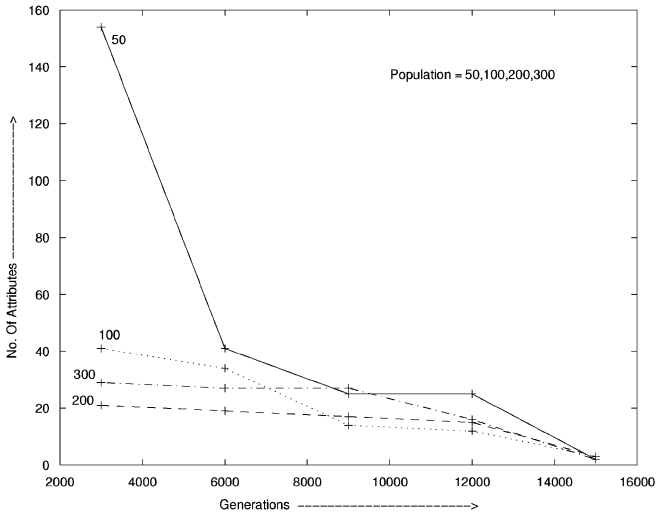
$$f_2(\vec{\nu}) = \frac{C_{\vec{\nu}}}{m_1 * m_2} \tag{9}$$

where m_1 and m_2 are the number of objects in the two classes. The basic steps of the proposed algorithm are summarized as follows.

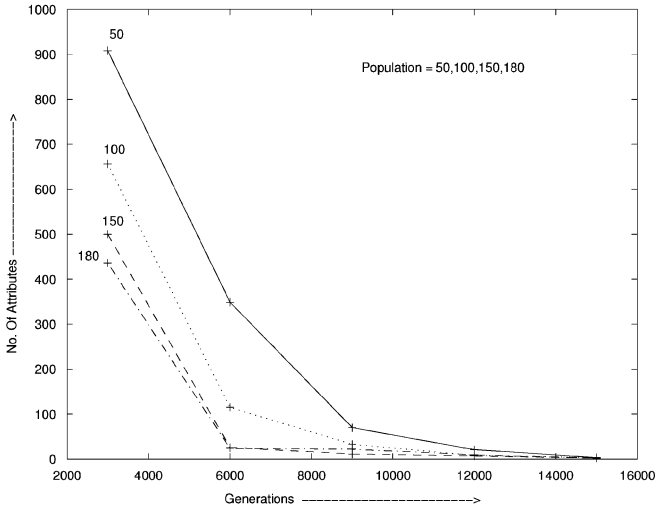
- Step 1) Redundancy reduction is made for the high-dimensional microarray data, as described in Section III-A, to get the reduced attribute value table $\mathcal{A}_r$.
- Step 2) d-distinction table is generated from $\mathcal{A}_r$ for the two classes being discerned.
- Step 3) A random population of size n is generated.
- Step 4) The two fitness values f_1 and f_2, for each individual, are calculated using (7) and (9).
- Step 5) Nondomination sorting is done, as discussed in Section II-B1, to identify different fronts.
- Step 6) Crowding sort based on crowding distance is performed to get widespread solutions.
- Step 7) Offspring solution of size n is created using *fitness* tournament selection, crossover, and mutation operators. This is a modification of crowded tournament selection of Section II-B3 with f_1 being accorded a higher priority over f_2 during solution selection from the same front. Specifically, for $r_i = r_j$ we favor solution i if $f_{1_i} < f_{1_j}$ (instead of $d_i < d_j$).
- Step 8) Select the best populations of size $n/2$, each from both the parent and offspring solutions, based on nondominated sorting to generate a combined population of size n. This modification enables effective handling of larger population sizes in the case of large datasets along with computational gain.
- Step 9) Steps 4)–7) are **repeated** for a prespecified number of generations.



(a)



(b)



(c)

Fig. 3. Plot of minimal reduct size versus generations, with different population sizes, for multiobjective optimization on (a) colon, (b) lymphoma, and (c) leukemia datasets.

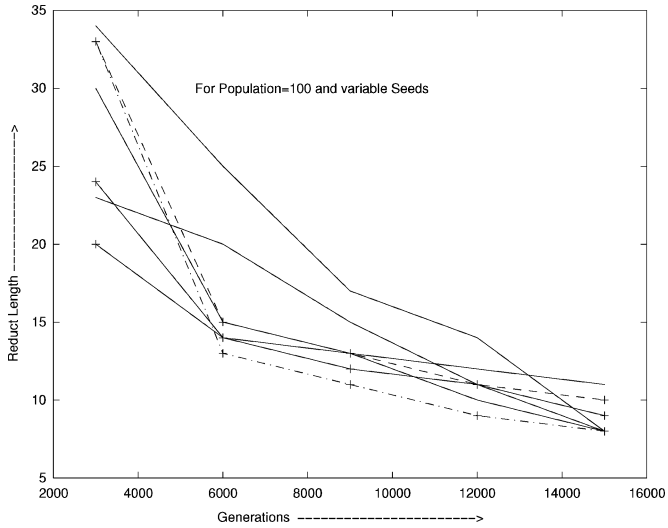


Fig. 4. Plot of minimal reduct size versus generations, over nine runs with different seed values, for multiobjective optimization on colon dataset.

IV. Experimental Results

We have implemented the proposed minimal feature selection algorithm on microarray data consisting of three different cancer samples. Availability of literature about performance of other related algorithms on these datasets, as summarized in Table I, prompted us to select them for our study. All results are averaged over several (three to five) runs involving different random seeds. No significant change was observed in the performance using different seeds.

The *colon cancer* data[1] are a collection of 62 gene expression measurements from colon biopsy samples. There are 22 normal (class $C2$) and 40 colon cancer (class $C1$) samples having 2000 genes (features). Fifty percent of the samples ($20 + 11 = 31$) was considered as the training set, while the remaining 50% ($20 + 11 = 31$) constituted the test set.

The *lymphoma* dataset[2] provides expression measurements from 96 normal and malignant lymphocyte samples, containing 42 cases of diffused large B-cell lymphoma (DLBCL) (class $C2$) and 54 cases of other types (class $C1$). There are 4026 genes present. Here, also, 50% of the samples ($27 + 21 = 48$) was considered as the training set, while the remaining 50% ($27 + 21 = 48$) constituted the test set.

The *leukemia* dataset[3] is a collection of gene expression measurements from 38 leukemia samples. There are 27 cases of acute lymphoblastic leukemia (ALL) and 11 cases of acute myeloblastic leukemia (AML). An independent test set, which composed of 20 ALL and 14 AML samples, was used for evaluating the performance of the classifier. The gene expression measurements were taken from high-density oligonucleotide microarrays containing 7129 genes (attributes).

After the initial redundancy reduction, by the procedure outlined in Section III-A, the feature sets were reduced to the following:

[1]http://microarray.princeton.edu/oncology
[2]http://llmpp.nih.gov/lymphoma/data/figure1/figure1.cdt
[3]http://www.genome.wi.mit.edu/MPR

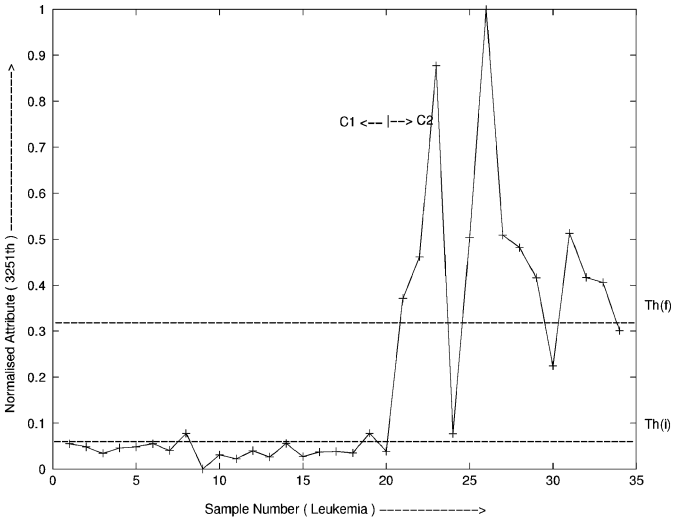


Fig. 5. Plot of normalized gene expression value of attributes (a) $U46499$, (b) $M28130$, and (c) $Y00787$ for leukemia dataset.

TABLE III
COMPARATIVE PERFORMANCE AS NUMBER OF MISCLASSIFICATIONS

Dataset	*Leukemia*		*Colon*		
# Genes:	2	3	8	9	10
# Misclassification for: Evolutionary-Rough	2	2	3	3	5
# Genes:	2	4	2	4	8
# Misclassification for: RSA	4	2	8.5 $\pm$ 0.58	6.5 $\pm$ 1.73	5.0 $\pm$ 1.41

1) colon dataset: 1102 attributes for the normal and cancer classes;
2) lymphoma dataset: 1867 attributes for normal and malignant lymphocyte cells;
3) leukemia dataset: 3783 attributes for classes ALL and AML.

A. Reduct Generation and Classification

The MOGA of Section III-D is run on the d-distinction table by using the fitness functions of (7) and (8), with different population sizes, to generate reducts upon convergence. Here, the two fitness functions $f_1(\vec{\nu})$ and $f_2(\vec{\nu})$ offset each other, such that the priority accorded to $f_1(\vec{\nu})$ in step 7) of the proposed algorithm of Section III-D allows weaker reducts (with less than 100% discrimination on object pairs from the d-distinction table) to appear in the best front. This can also be observed from the Pareto optimal front provided in Fig. 2. Note that reducts of sizes 9–14, 11–13, and 6–7 for the aforementioned three datasets are capable of 100% discrimination, while some of the other weaker reducts (consisting of less number of attributes) are incapable of perfect discrimination.

Sample results are provided in Table II on the three sets of two-class microarray gene expression data after 15 000 generations. The corresponding recognition scores (in percentage) (on test set) by the powerful k-nearest neighbors (k-NN) classifier [28], for different values of k, are also presented in the table. We do not use other classifiers such as decision-tree that typically deal with symbolic (nonnumeric) data. Neural nets were not explored since our objective was to focus on the classification ability of the reducts generated and not to further improve upon the recognition at the expense of increased computational complexity.

As the number of attributes decreases, it is observed that the k-NN performance on the test set improves. This is mainly due to the elimination of the existing large redundancy inherent in gene expression data. However, this classification is different from the discrimination over the d-distinction table, as shown in the Pareto optimal front of Fig. 2 for training data.

It is observed that the number of features get reduced considerably with the evolutionary progression of reduct generation. A larger initial population size leads to a smaller size of reducts faster, and hence, a correspondingly higher fitness value. This is depicted in Fig. 3. However, the associated computational complexity also increases with a larger size of population, thereby resulting in a limitation in terms of available space and time.

Fig. 4 shows the effect on reduct length as MOGA progresses on *colon* data, for nine runs with different seed values, over

TABLE IV
COMPARATIVE PERFORMANCE ON GENE EXPRESSION DATA USING SINGLE-OBJECTIVE GA

Dataset	Minml. reduct size	k-nearest neighbors classification (%) on test set											
		$k=1$			$k=3$			$k=5$			$k=7$		
		C1	C2	Net	C1	C2	Net	C1	C2	Net	C1	C2	Net
Colon	15	75.0	63.6	71.0	70.0	36.4	58.1	75.0	0.0	48.4	90.0	9.1	61.3
Lymphoma	18	85.2	71.4	79.2	81.5	90.5	85.4	92.6	81.0	87.5	92.6	85.7	89.6
Leukemia	19	90.0	50.0	73.5	90.0	57.1	76.5	95.0	14.3	61.7	100.0	14.3	64.7

different number of generations involving a sample population size of 100 chromosomes.

The Pareto optimal front, with a population size of 100 chromosomes, is illustrated in Fig. 2 for the three datasets. Here, we plot the reduct size versus the number of misclassifications over the training set (as obtained from the object pairs in the d-distinction table) at the end of 15 000 generations. As mentioned earlier, the best front contains minimal as well as weaker reducts based on the formulation of the fitness function.

We found that with an increase in the number of generations, the $f_1(\vec{\nu})$ component of (7) gains precedence over the $f_2(\vec{\nu})$ component of (8) in the fitness function. Thereby the number of minimal reducts (having 100% discrimination between training samples) decreases, as compared to those having less number of attributes but incapable of perfect discrimination.

Fig. 5 depicts sample normalized gene expression values of a set of three genes for *leukemia* data. The partitions Th_i and Th_f of (6) are marked parallel to the abscissa. The samples are listed classwise and sequentially, such that one can observe the marked change in gene expression values of these attributes corresponding to the two output classes. This also serves to highlight the importance of these selected features (in reduct) in arriving at a good discriminatory decision.

B. Comparison

Feature selection has been reported in the literature [13]–[15]. Huang [13] used a probabilistic neural network for feature selection based on correlation with class distinction. In the case of the *leukemia* data, there is 100% correct classification with a ten-gene set. For the *colon* data, a ten-gene set produces a classification score of 79.0%. From Table II, we obtain a correct classification of 90.3% with a nine-gene set, whereas the reduced attribute size comes down to two or three for the *leukemia* data.

Chu *et al*. [15] employ a t-test-based feature selection with a fuzzy neural network. A five-gene set provides 100% correct classification for *lymphoma* data. We determine from Table II a misclassification on just two samples from the test data using a two-gene set for *lymphoma*.

Cao *et al*. [14] apply saliency analysis to support vector machines for gene selection in tissue classification. The importance of genes is ranked by evaluating the sensitivity of the output to the inputs in terms of the partial derivative. The recursive saliency analysis (RSA) algorithm is developed to remove irrelevant genes in the case of *leukemia* and *colon* data. Table III lists a comparative study of RSA with the proposed evolutionary rough method in terms of the number of misclassification on the test data.

We also made a comparative study with some other logically similar fitness functions [22], [23], involving combinations of

$$f_1(\vec{\nu}) = \frac{1}{L_{\vec{\nu}}} \tag{10}$$

$$f_2(\vec{\nu}) = \begin{cases} \dfrac{C_{\vec{\nu}}}{(m^2-m)/2}, & \text{if } C_{\vec{\nu}} < (m^2-m)/2 \\ \left(\dfrac{C_{\vec{\nu}}}{(m^2-m)/2} + \dfrac{1}{2}\right) * \frac{1}{2}, & \text{if } C_{\vec{\nu}} = (m^2-m)/2 \end{cases} \tag{11}$$

as adapted to the multiobjective framework. There was no observable improvement in performance here while using the multiobjective algorithm, as compared to that employing the functions of (7) and (8).

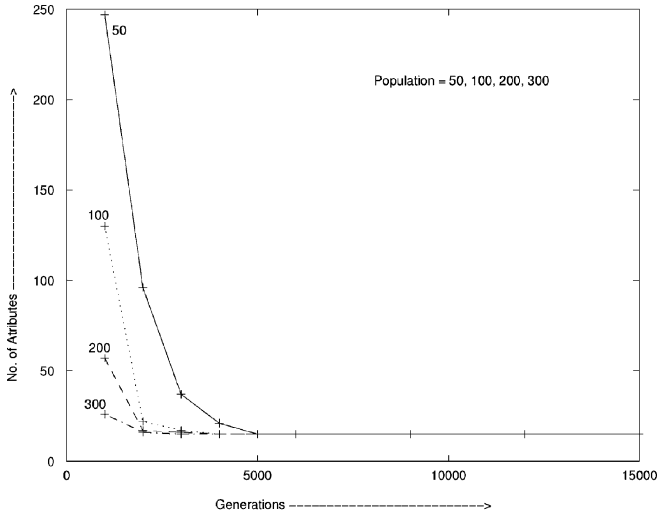
Reduct generation with a single-objective (classical) GA [22] was also investigated for different population sizes. The fitness function

$$F_t = \alpha_1 f_1(\vec{\nu}) + \alpha_2 f_2(\vec{\nu}) \tag{12}$$

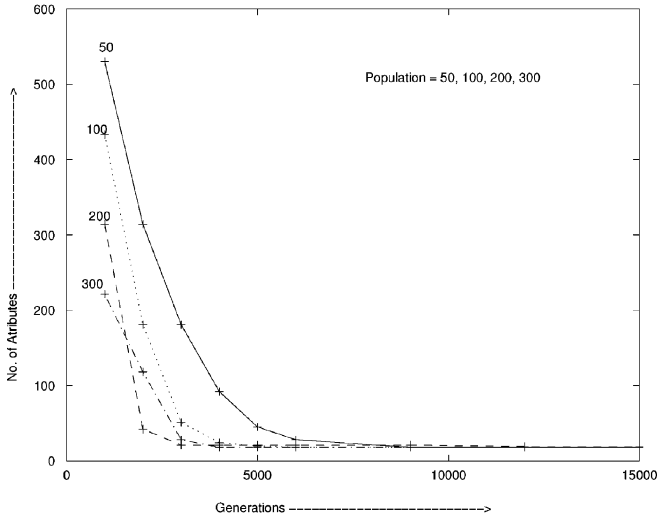
was used, in terms of (7) and (8), with the parameters $\alpha_1 = \alpha_2 = 1$. Additionally, we investigated with $0 < \alpha_1, \alpha_2 < 1$ for $\alpha_1 = 1 - \alpha_2$. Sample results are provided in Table IV for population size of 100, with optimal values being generated for $\alpha_1 = 0.9$. It is observed that, for different choices of α_1 and α_2, the size of the minimal reduct was 15, 18, and 19 for *colon, lymphoma,* and *leukemia* data, respectively. This is more than the reduct size generated by the proposed multiobjective approach illustrated in Table II. Moreover, the classification performance in Table IV is also observed to be poorer.

The variation in reduct size with generations and different populations, using single-objective optimization, is depicted in Fig. 6 for the three datasets. Comparison is provided for the same set of parameter initializations, with runs over the same number of generations, using similar objective functions. It is observed that the plots stabilize at a larger reduct size in the case of single-objectives, as compared to the case of multiobjective optimization. Moreover, there is a convergence to noticeably homogeneous solutions in the population, in the case of the single-objective GA, resulting in stagnation of performance.

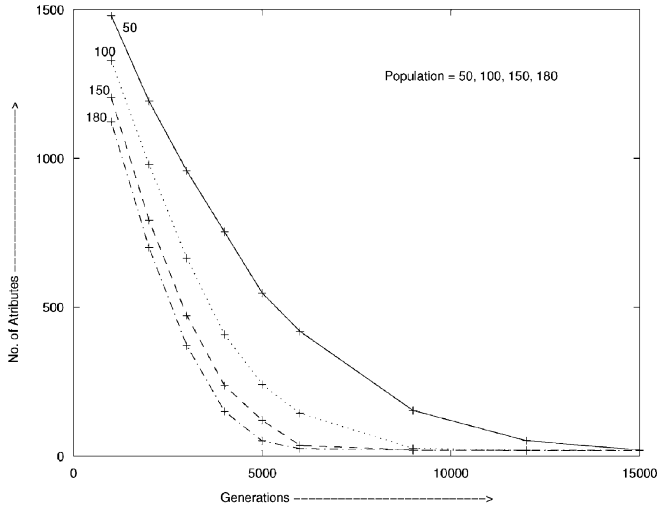
A principal component analysis (PCA)-based [28] data reduction technique has been very popular in data mining [10]. Eigenvectors of the covariance matrix of a dataset identify a linear projection that produces uncorrelated features. PCA allows extraction of the most relevant eigenvalues and eigenvectors, which provide a good approximation for the discrimination. We employed PCA to generate an optimal set of n_o transformed features (eigenvalues), subject to a threshold of 99% approximation for the decision function. These features were subsequently evaluated using a k-NN classifier (for k = 1, 3, 5, 7) on the test



(a)



(b)



(c)

Fig. 6. Plot of minimal reduct size versus generations, with different population sizes, for single-objective optimization on (a) colon, (b) lymphoma, and (c) leukemia datasets.

set. The best performance (percentage recognition scores) for the three cancer datasets were as follows:

1) *colon*: $k = 1$ with $n_o = 7$ eigenvalues, $C1 = 90.0, C2 = 66.7, Net = 80.7$;
2) *lymphoma*: $k = 3$ with $n_o = 8$ eigenvalues, $C1 = 95.5, C2 = 96.3, Net = 95.8$;
3) *leukemia*: $k = 1$ with $n_o = 8$ eigenvalues, $C1 = 100.0, C2 = 64.3, Net = 85.3$.

Our evolutionary rough strategy is found to work better in all the cases.

V. Conclusion

We have described an evolutionary rough feature selection algorithm using redundancy reduction for effective handling of high-dimensional microarray gene expression data. This serves as an interesting study in bioinformatics. The NSGA-II has been modified to more effectively handle large data.

Since microarray data typically consist of a large number of redundant attributes, we have done an initial preprocessing for redundancy reduction. The objective was to retain only those genes that play a major role in discerning between objects. This preprocessing aids faster convergence along the search space. Moreover, a reduction in the rows (object pairs) of the distinction table was made by restricting comparisons only between objects belonging to different classes—giving the d-distinction table. This is intuitively meaningful, since our objective here is to determine the reducts that can discern between objects belonging to different classes. A further reduction in computational complexity is thereby achieved.

Selection of the most frequently occurring attributes amongst the reducts may prove significant for biologists. This is because the attributes in the *core* [17] (the intersection of the reducts) could be the relevant genes responsible for a certain medical condition. For example, let us consider the results presented in Table II to illustrate selection of important attributes (or genes) in the reducts. It is found that generally gene IDs $Hsa.8147$ and $Hsa.1039$ occurred most frequently amongst the reducts in the case of *colon* data. Similar analysis on *lymphoma* data led to a focus on genes $1559X$ and $1637X$. In the case of *leukemia* data, the genes $U46499$, $M28130$, and $Y00787$ are found to be in the core. In the next phase, we plan to collaborate with biological experts toward validating these findings.

Microarray bioinformatics has aided in a massive parallelization of experimental biology [1], and the associated explosion of research has led to astonishing progress in our understanding of molecular biology. Future hybrid approaches, combining powerful algorithms and interactive visualization tools with the strengths of fast processors, hold further promise for enhanced performance.

Acknowledgement

The authors would like to thank the referees for their valuable comments.

References

[1] "Special issue on bioinformatics," *IEEE Comput.*, vol. 35, Jul. 2002.
[2] "Special issue on bioinformatics, part I: Advances and Challenges," *Proc. IEEE*, vol. 90, no. 11, Nov. 2002.

[3] S. B. Cho and H. H. Won, "Data mining for gene expression profiles from DNA microarray," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 13, pp. 593–608, 2003.
[4] H. H. Won and S. B. Cho, "Ensemble classifier with negatively correlated features for cancer classification," *J. KISS: Softw. Appl.*, vol. 30, pp. 1124–1134, 2003.
[5] S. Ando and H. Iba, "Artificial immune system for classification of cancer: Applications of evolutionary computing," in *Lecture Notes Computer Science*, vol. 2611, Berlin, Germany, Springer-Verlag, 2003, pp. 1–10.
[6] M. E. Futschik, A. Reeve, and N. Kasabov, "Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue," *Artif. Intell. Med.*, vol. 28, pp. 165–189, 2003.
[7] V. Roth and T. Lange, "Bayesian class discovery in microarray datasets," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 5, pp. 707–718, May 2004.
[8] Y. Turkeli, A. Ercil, and O. U. Sezerman, "Effect of feature extraction and feature selection on expression data from epithelial ovarian cancer," in *Proc. 25th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2003, vol. 4, pp. 3559–3562.
[9] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Commun. ACM*, vol. 37, pp. 77–84, 1994.
[10] S. Mitra and T. Acharya, *Data mining: Multimedia, Soft Computing, and Bioinformatics*. New York: Wiley, 2003.
[11] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," presented at the 20th Int. Conf. Mach. Learn. (ICML-2003). Washington, DC, 2003.
[12] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, 2004.
[13] C. J. Huang, "Class prediction of cancer using probabilistic neural networks and relative correlation metric," *Appl. Artif. Intell.*, vol. 18, pp. 117–128, 2004.
[14] L. Cao, H. P. Lee, C. K. Seng, and Q. Gu, "Saliency analysis of support vector machines for gene selection in tissue classification," *Neural Comput. Appl.*, vol. 11, pp. 244–249, 2003.
[15] F. Chu, W. Xie, and L. Wang, "Gene selection and cancer classification using a fuzzy neural network," in *Proc. 2004 Annu. Meet. North Amer. Fuzzy Inf. Process. Soc.*, vol. 2, pp. 555–559.
[16] M. Karzynski, A. Mateos, J. Herrero, and J. Dopazo, "Using a genetic algorithm and a perceptron for feature selection and supervised class learning in DNA microarray data," *Artif. Intell. Rev.*, vol. 20, pp. 39–51, 2003.
[17] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning About Data*. Dordrecht, The Netherlands: Kluwer, 1991.
[18] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," in *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*, R. Slowiński, Ed. Dordrecht, The Netherlands: Kluwer, 1992.
[19] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, "Rough sets: A tutorial," in *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, S. K. Pal and A. Skowron, Eds. Singapore: Springer-Verlag, 1999, pp. 3–98.
[20] S. Tsumoto, R. Slowinski, J. Komorowski, and J. Grzymala-Busse, Eds., Rough Sets and Current Trends in Computing (RSCTC). Berlin, Germany: Springer-Verlag, 2004.
[21] N. Zhong, J. Dong, and S. Ohsuga, "Using rough sets with heuristics for feature selection," *J. Intell. Inf. Syst.*, vol. 16, pp. 199–214, 2001.
[22] J. Wroblewski, "Finding minimal reducts using genetic algorithms," Inst. Comput. Sci., Warsaw Inst. Technol., Warsaw, Poland, Tech. Rep. 16/95, 1995.
[23] A. T. Bjorvand, "'Rough enough'—A system supporting the rough sets approach," in *Proc. 6th Scandinavian Conf. Artif. Intell.*, Helsinki, Finland, 1997, pp. 290–291.
[24] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[25] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. London, U.K.: Wiley, 2001.
[26] A. Anand, "Representation and learning of inexact information using rough set theory," M.S. thesis, Dept. Math., Indian Inst. Technol. Kanpur, Kanpur, India, 2002.
[27] K. Deb, S. Agarwal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[28] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. London, U.K.: Addison-Wesley, 1974.

Mohua Banerjee received the B.Sc.(Hons.) degree in mathematics, and the M.Sc., M.Phil., and Ph.D. degrees in pure mathematics from the University of Calcutta, Kolkata, India, in 1985, 1987, 1989, and 1995, respectively.

During 1995–1997, she was a Research Associate with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. In 1997, she joined as a Lecturer in the Department of Mathematics and Statistics, Indian Institute of Technology, Kanpur, where she is currently an Assistant Professor. Her current research interests include modal logics and rough set theory and its applications. She is a reviewer of various international journals.

Dr. Banerjee was awarded the Indian National Science Academy Medal for Young Scientists in 1995. She is a member of the Interim Advisory Board of the International Rough Set Society. She is also a member of the Working Group for the Centre for Research in Discrete Mathematics and Its Applications (CARDMATH), Department of Science and Technology (DST), Government of India. She has been an Associate of the Institute of Mathematical Sciences, Chennai, India, during 2003–2005.

Sushmita Mitra (S'91–M'92–SM'00) received the Ph.D. degree in computer science from the Indian Statistical Institute, Kolkata, in 1995.

She is currently a Professor with the Machine Intelligence Unit, Indian Statistical Institute. From 1992 to 1994, she was with the Die Leitseite der Rheinisch-Westfälischen Technischen Hochschule (RWTH), Aachen, Germany, as a DAAD Fellow. She was a Visiting Professor in the Computer Science Departments of the University of Alberta, in 2004, Meiji University, Tokyo, in 19991, 2004, and 2005, and Aalborg University Esbjerg, Denmark, in 2002 and 2003. She is the author of *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing* (New York: Wiley) and *Data Mining: Multimedia, Soft Computing, and Bioinformatics* (New York: Wiley). She has been a Guest Editor of various journal special issues and is an Associate Editor of *Neurocomputing*. She has more than 100 research publications in refereed international journals. According to the Science Citation Index (SCI), two of her papers have been ranked third and 15th in the list of top-cited papers in engineering science from India during 1992–2001. Her current research interests include data mining, pattern recognition, soft computing, image processing, and bioinformatics.

Dr. Mitra received the National Talent Search Scholarship (1978–1983) from the National Council of Educational Research and Training, India, the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award in 1994 for her pioneering work in neuro-fuzzy computing, and the CIMPA-INRIA-UNESCO Fellowship in 1996. She is a Fellow of the Indian National Academy of Engineering.

Haider Banka received the M.Sc. and M.Tech. degrees in computer science from the University of Calcutta, Kolkata, India, in 2001 and 2003, respectively.

During 2003 to 2004, he was a Lecturer in the Engineering College, Durgapur, India. Since 2004, he has been a Senior Research Fellow with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. He is a reviewer of several international journals. His current research interests include data mining, pattern recognition, soft computing, combinatorial optimization, and bioinformatics.