# Genetic clustering for automatic evolution of clusters and application to image classification

Sanghamitra Bandyopadhyay[a,*], Ujjwal Maulik[b]

[a] *Machine Intelligence Unit, Indian Statistical Institute, 203, B.T. Road, Calcutta - 700 035, India*
[b] *Department of Computer Science and Technology, Kalyani Government Engineering College, Kalyani, Nadia, India*

Received 22 September 2000; accepted 11 May 2001

**Abstract**

In this article the searching capability of genetic algorithms has been exploited for automatically evolving the number of clusters as well as proper clustering of any data set. A new string representation, comprising both real numbers and the do not care symbol, is used in order to encode a variable number of clusters. The Davies–Bouldin index is used as a measure of the validity of the clusters. Effectiveness of the genetic clustering scheme is demonstrated for both artificial and real-life data sets. Utility of the genetic clustering technique is also demonstrated for a satellite image of a part of the city Calcutta. The proposed technique is able to distinguish some characteristic landcover types in the image.

*Keywords:* Clustering; Davies–Bouldin index; Genetic algorithms; Real encoding; Satellite image classification

## 1. Introduction

Genetic algorithms (GAs) [1–4] belong to a class of search techniques that mimic the principles of natural selection to develop solutions of large optimization problems. GAs operate by maintaining and manipulating a population of potential solutions called chromosomes. Each chromosome has an associated fitness value which is a qualitative measure of the goodness of the solution encoded in it. This fitness value is used to guide the stochastic selection of chromosomes which are then used to generate new candidate solutions through crossover and mutation. Crossover generates new chromosomes by combining sections of two or more selected parents. Mutation acts by randomly selecting genes which are then altered; thereby preventing suboptimal solutions from persisting and increases diversity in the population. The process of selection, crossover and mutation continues for a fixed number of generations or until a termination condition is satisfied. GAs have applications in fields as diverse as VLSI design, pattern recognition, image processing, neural networks, machine learning, etc. [5–12].

Clustering [13–17] is a popular unsupervised pattern classification technique which partitions the input space into $K$ regions based on some similarity/dissimilarity metric. The number of partitions/clusters may or may not be known a priori. Let the input space $S$ be represented by $n$ points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, and the $K$ clusters be represented by $C_1, C_2, \ldots, C_K$. Then

$$C_i \neq \emptyset \quad \text{for } i = 1, \ldots, K,$$

$$C_i \cap C_j = \emptyset \quad \text{for } i = 1, \ldots, K, \quad j = 1, \ldots, K \quad \text{and}$$

$$i \neq j, \quad \text{and}$$

$$\bigcup_{i=1}^{K} C_i = S.$$

Several algorithms for clustering data when the number of clusters is known a priori are available in the literature

* Corresponding author.
*E-mail addresses:* sanghami@isical.ac.in (S. Bandyopadhyay), ujjwal_maulik@kucse.wb.nic.in (U. Maulik).

viz., $K$-means [17], branch and bound procedure [18], maximum likelihood estimate technique [19], graph theoretic approaches [20]. The $K$-means algorithm, one of the most widely used ones, attempts to solve the clustering problem by optimizing a given metric. The branch and bound procedure uses a tree search technique for searching the entire solution space for classifying a given set of points into a fixed number of clusters, along with a criterion for eliminating subtrees which do not contain the optimal result. In this scheme, the number of nodes to be searched becomes huge as the size of the data set becomes large; therefore a proper choice of the criterion for eliminating subtrees becomes crucial [21]. The maximum likelihood estimate technique performs clustering by computing the posterior probabilities of the classes after assuming a particular distribution of the data set. In the graph theoretic approach, a direct tree is formed among the data set by estimating the density gradient at each point. The clustering is realized by finding the valley of the density function. It is known that the quality of the result depends wholly on the quality of the estimation technique for the density gradient, particularly in the low-density area of the valley. Recently, a genetic algorithm based clustering technique has been developed [22] which does not assume any particular underlying distribution of the data set while it is conceptually simple as the $K$-means algorithm. Moreover, it does not suffer from the limitation of the $K$-means algorithm, which is known to get stuck at sub-optimal solutions depending on the choice of the initial cluster centers. However, as in the $K$-means algorithm, the methodology proposed in Ref. [22] is applicable to the cases where the number of clusters is known a priori.

In most real life situations the number of clusters in a data set is not known a priori. The real challenge in this situation is to be able to automatically evolve a proper value of $K$ as well as providing the appropriate clustering. In this article, we propose a GA based clustering technique which can automatically evolve the appropriate clustering of a data set. The chromosome encodes the centres of a number of clusters, whose value may vary. Modified versions of crossover and mutation operators are used. Cluster validity index like Davies–Bouldin index [23] is utilized for computing the fitness of the chromosomes.

The effectiveness of the genetic clustering technique is demonstrated on four artificial and two real life data sets having different characteristics. Another interesting real life application of the clustering technique is provided for automatically classifying a SPOT satellite image into distinct landcover regions.

## 2. Genetic clustering

In this section, an attempt has been made to use genetic algorithms for automatically clustering a data set. This includes determination of number of clusters as well as appropriate clustering of the data. The methodology is explained first followed by the description of the implementation results.

### 2.1. The methodology

The genetic clustering technique is subsequently referred to as the genetic clustering for unknown $K$ (*GCUK-clustering*), where $K$ denotes the number of clusters. A flowchart of the method is provided in Fig. 1. The different steps of *GCUK-clustering* are now discussed in detail.

#### 2.1.1. String representation

In *GCUK-clustering*, the chromosomes are made up of real numbers (representing the coordinates of the centres) as well as the don't care symbol '#'. The value of $K$ is assumed to lie in the range $[K_{min}, K_{max}]$, where $K_{min}$ is chosen to be 2 unless specified otherwise. The length of a string is taken to be $K_{max}$ where each individual gene position represents either an actual center or a don't care symbol.

#### 2.1.2. Population initialization

For each string $i$ in the population ($i = 1, \ldots, P$, where $P$ is the size of the population), a random number $K_i$ in the range $[K_{min}, K_{max}]$ is generated. This string is assumed to encode the centres of $K_i$ clusters. For initializing these centres, $K_i$ points are chosen randomly from the data set. These points are distributed randomly in the chromosome. Let us consider the following example.

*Example*: Let $K_{min} = 2$ and $K_{max} = 10$. Let the random number $K_i$ be equal to 4 for chromosome $i$. Then this chromosome will encode the centres of 4 clusters. Let the 4 cluster centres (4 randomly chosen points from the data set) be

(10.0, 5.0) (20.4, 13.2) (15.8, 2.9) (22.7, 17.7).

On random distribution of these centres in the chromosome, it may look like

# (20.4, 13.2) # # (15.8, 2.9) # (10.0, 5.0) (22.7, 17.7) # #.

#### 2.1.3. Fitness computation

The fitness of a chromosome is computed using the Davies–Bouldin [23] index. This index is a function of the ratio of the sum of *within-cluster scatter* to *between-cluster separation*. The scatter within $C_i$, the $i$th cluster, is computed as

$$S_{i,q} = \left( \frac{1}{|C_i|} \sum_{x \in C_i} \{ ||x - z_i||_2^q \} \right)^{1/q}, \quad (1)$$

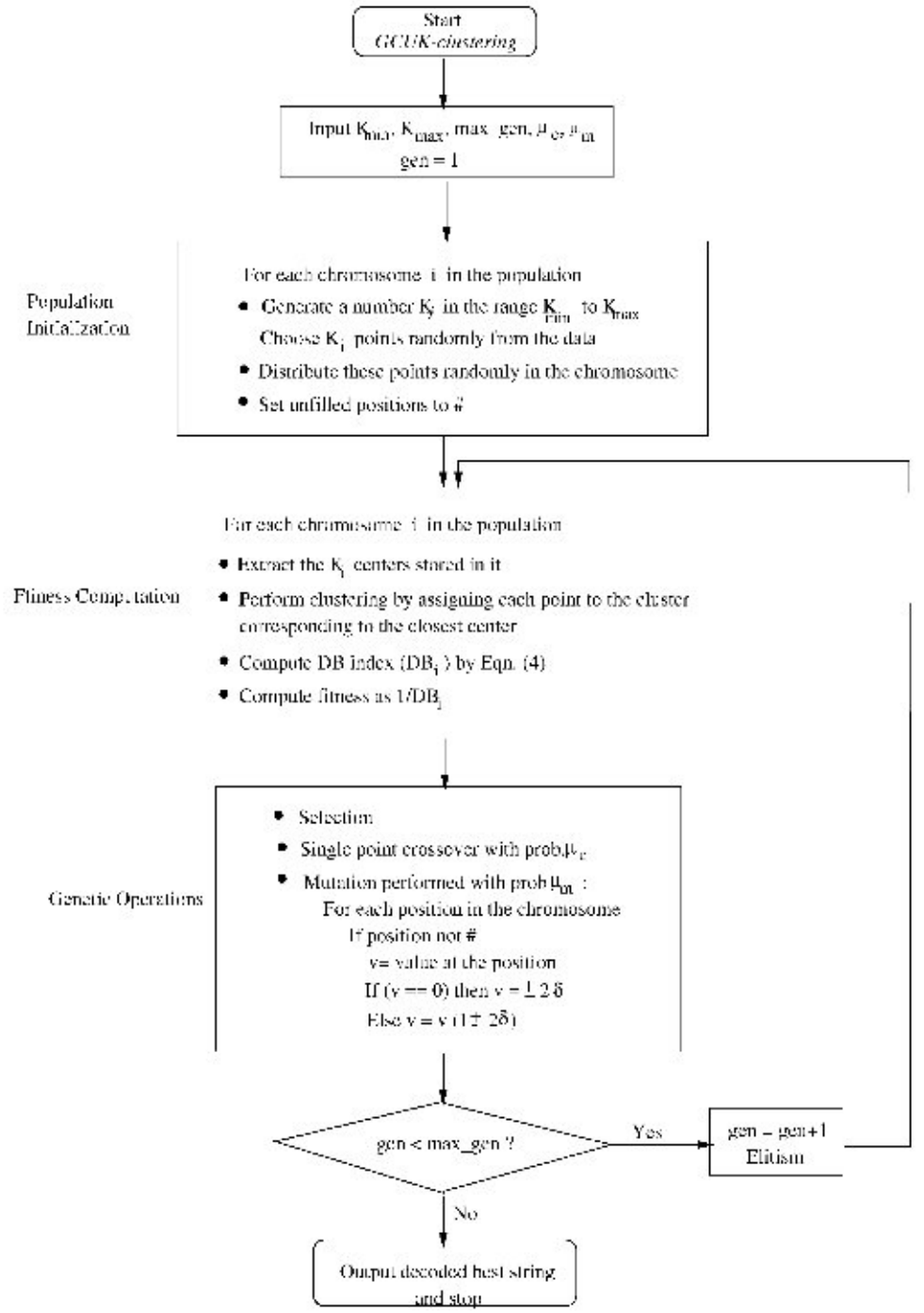


Fig. 1. Flowchart of *GCUK-clustering*.

where, $z_i$ is the centroid of $C_i$, and is defined as $z_i = 1/n_i \sum_{x \in C_i} x$, and $n_i$ is the cardinality of $C_i$, i.e., the number of points in cluster $C_i$. The distance between cluster $C_i$ and $C_j$ is defined as

$$d_{ij,t} = \left\{ \sum_{s=1}^{p} |z_{is} - z_{js}|^t \right\}^{1/t} = ||z_i - z_j||_t. \quad (2)$$

$S_{i,q}$ is the $q$th root of the $q$th moment of the points in cluster $i$ with respect to their mean, and is a measure of the dispersion of the points in cluster $i$. Specifically, $S_{i,1}$, used in this article, is the average Euclidean distance of the vectors in class $i$ to the centroid of class $i$. $d_{ij,t}$ is the Minkowski distance of order $t$ between the centroids that characterize clusters $i$ and $j$. Subsequently we compute

$$R_{i,qt} = \max_{j,j\neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}. \quad (3)$$

The Davies–Bouldin (DB) index is then defined as

$$DB = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt}. \quad (4)$$

The objective is to minimize the DB index for achieving proper clustering. The fitness function for chromosome $j$ is defined as $1/DB_j$, where $DB_j$ is the Davies–Bouldin index computed for this chromosome. Note that maximization of the fitness function will ensure minimization of the DB index.

#### 2.1.4. Genetic operations

The following genetic operations are performed on the population of strings for a number of generations.

*Selection*: Conventional proportional selection is applied on the population of strings. Here, a string receives a number of copies that is proportional to its fitness in the population.

*Crossover*: During crossover each cluster centre is considered to be an indivisible gene. Single point crossover, applied stochastically with probability $\mu_c$, is explained below with an example.

*Example*: Suppose crossover occurs between the following two strings:

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # | | (20.4, 13.2) | # | # | (15.8, 2.9) | | # | (10.0, 5.0) | | (22.7, 17.7) | # | | # | | | | | | |
| (13.2, 15.6) | | # | | # | # | (5.3, 13.7) | | # | (10.5, 16.2) | | (7.9, 15.3) | # | (18.3, 14.5) | | | | | | |

Let the crossover position be 5 as shown above. Then the offspring are

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # | | (20.4, 13.2) | # | # | (15.8, 2.9) | | # | (10.5, 16.2) | | (7.9, 15.3) | # | (18.3, 14.5) | | | | | | | |
| (13.2, 15.6) | | # | | # | # | (5.3, 13.7) | | # | (10.0, 5.0) | | (22.7, 17.7) | # | | # | | | | | |

*Mutation*: Each valid position (i.e., which is not '#') in a chromosome is mutated with probability $\mu_m$ in the following way. A number $\delta$ in the range [0, 1] is generated with uniform distribution. If the value at that position is $v$, then after mutation it becomes

$$v \times (1 \pm 2\delta), \quad v \neq 0$$

$$\pm 2\delta, \quad v = 0.$$

The '+' or '−' sign occurs with equal probability.

#### 2.1.5. Termination criterion

In this article the processes of fitness computation, selection, crossover, and mutation are executed for a maximum number of iterations. The best string having the largest fitness (i.e., smallest DB index value) seen up to the last generation provides the solution to the clustering problem. We have implemented elitism at each generation by preserving the best string seen up to that generation in a location outside the population. Thus on termination, this location contains the centres of the final clusters.

### 2.2. Implementation results

The experimental results showing the effectiveness of *GCUK-clustering* algorithm are provided for four artificial and two real life data sets. The artificial data sets are (*Data_3_2*, *Data_5_2*, *Data_6_2* and *Data_4_3*), where the first three data sets are in two dimensions with 3, 5 and 6 clusters, respectively, and the last one is in three dimensions with 4 clusters. Figs. 2–5 show the four data sets. Table 1 presents the number of points, dimensions and the number of clusters in each data.

Two real-life data sets considered are *Iris* and *Breast Cancer*. These are described below:

*Iris Data*: This data represents different categories of irises having four feature values. The four feature values represent the sepal length, sepal width, petal length and the petal width in centimeters [24]. It has three classes Setosa, Versicolor and Virginica, with 50 samples per class. It is known that two classes Versicolor and Virginica have a large amount of overlap while the class Setosa is linearly separable from the other two.

*Breast Cancer*: Here, we use the Wisconsin Breast Cancer data set available at [http://www.ics.uci.edu/mlearn/MLRepository.html]. Each pattern has nine features corresponding to *clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli* and *mitoses*. There are two categories in the data: malignant and benign. The two classes are known to be linearly inseparable. There are a total of 683 points in the data set.

*GCUK-clustering* is implemented with the following parameters: $\mu_c = 0.8$, $\mu_m = 0.001$. The population size $P$

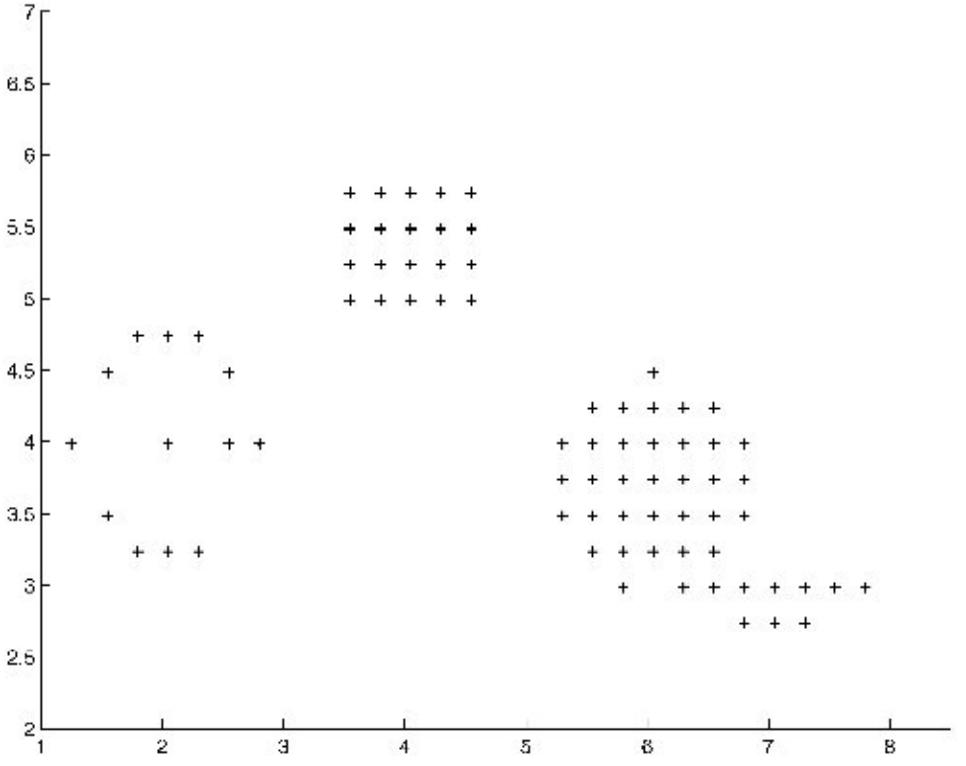


Fig. 2. Data_3_2.

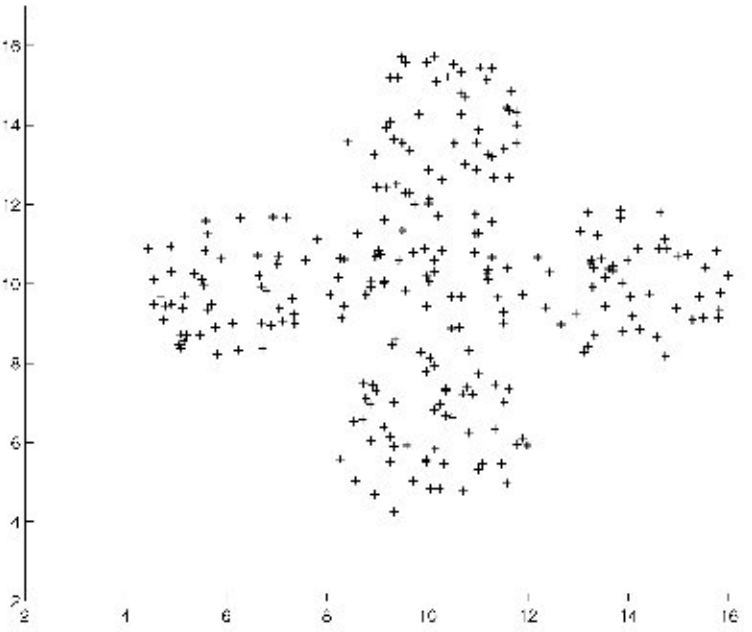


Fig. 3. Data_5_2.

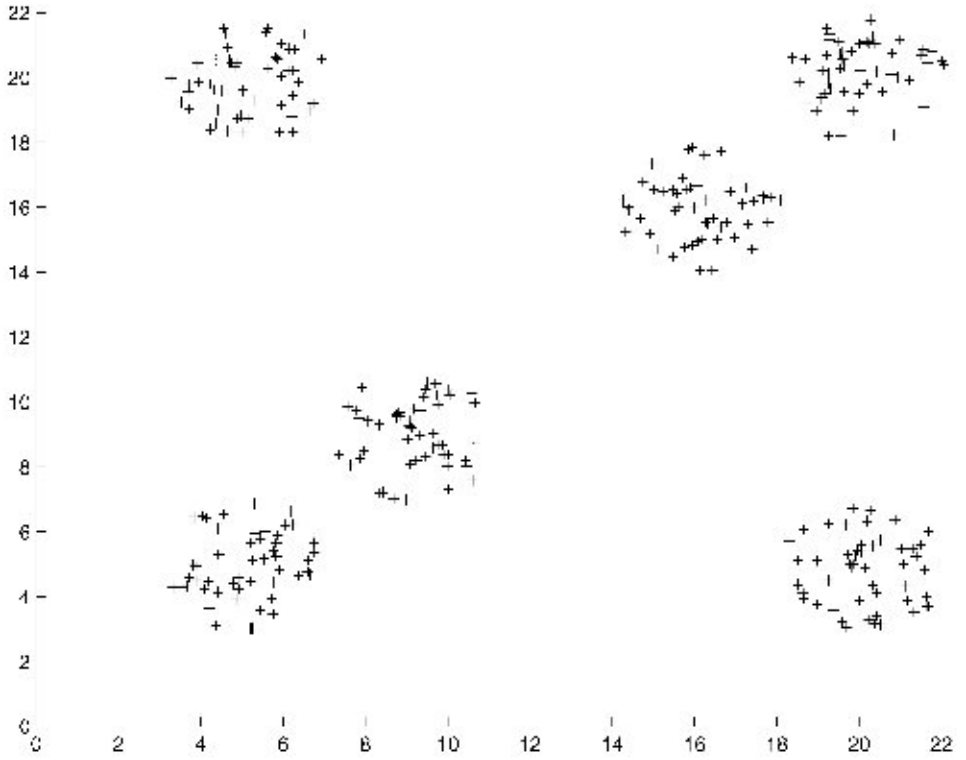


Fig. 4. Data_6_2.

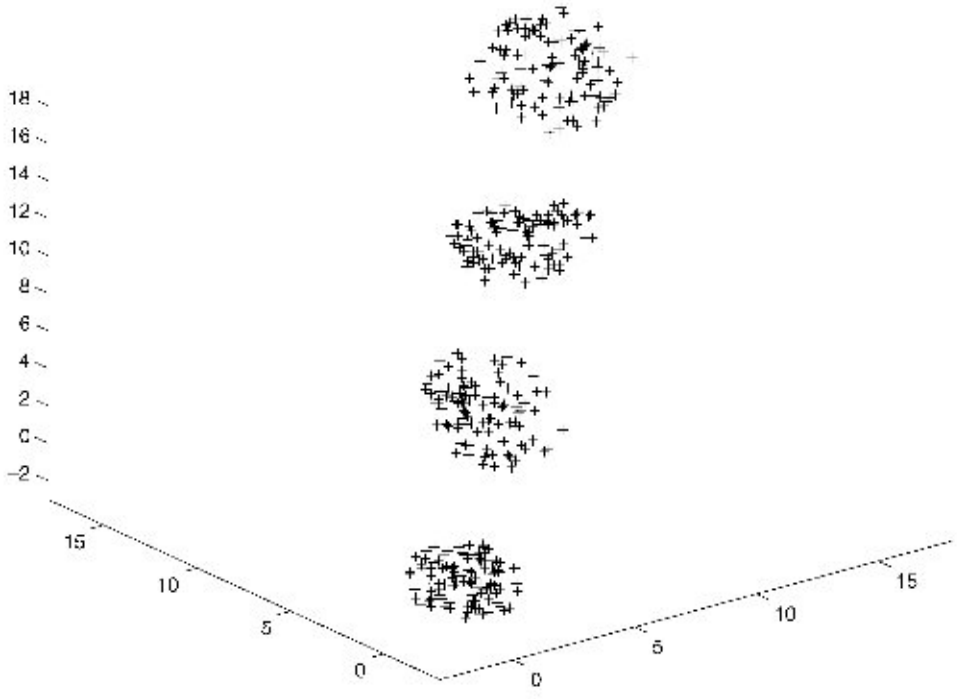


Fig. 5. Data_4_3.

Table 1
Description of the data sets

| Name | # points | # clusters | # dimensions | Points per cluster |
|---|---|---|---|---|
| Data_3_2 | 76 | 3 | 2 | 43,20,13 |
| Data_5_2 | 250 | 5 | 2 | 50 per cluster |
| Data_6_2 | 300 | 6 | 2 | 50 per cluster |
| Data_4_3 | 400 | 4 | 3 | 100 per cluster |
| Iris | 150 | 3 | 4 | 50 per cluster |
| Cancer | 683 | 2 | 9 | 444,239 |

is taken to be 50. The values of $K_{min}$ and $K_{max}$ are taken to be 2 and 10, respectively. *GCUK-clustering* is executed for a maximum of 100 iterations. Note that it is shown in Refs. [13,25] that if exhaustive enumeration is used to solve a clustering problem with $n$ points and $K$ clusters, then one requires to evaluate $1/K \sum_{j=1}^{K} (-1)^{K-j} j^n$ partitions. For a data set of size 10 with 2 clusters, this value is $2^9 - 1$ (= 511), while that of size 50 with 2 clusters is $2^{49} - 1$ (i.e., of the order of $10^{15}$). If the number of clusters cannot be specified a priori, then the search space will be even larger. The utility of GAs becomes evident in such situations, where we find that reasonably good results are obtained while evaluating significantly smaller number of partitions.

Table 2 provides the number of clusters and coordinates of the corresponding centres found by the *GCUK-clustering* technique for the four artificial data sets and one real life data *Cancer*. Also included are the actual values as obtained from the labelled data. It is evident from the table that *GCUK-clustering* is able to evolve the proper number of clusters in all these cases, and the computed centres are also close to the actual ones (see Figs. 6–9 demonstrating the clustering obtained for the four artificial data sets).

For Iris, *GCUK-clustering* provided two clusters, one corresponding to the class Setosa, and the other to the combination of Versicolor and Virginica. This is understandable from the fact that the latter two classes are significantly overlapping. Moreover, it was found that the DB index for the two clusters (0.39628) was smaller than that for 3 clusters (0.74682) and hence the former was preferred over the latter. In this connection one may also note that several indices have been found to provide two clusters for Iris [26,27].

## 3. Application to satellite image classification

The satellite image of a part of the city of Calcutta considered for the experiment was obtained by the French satellites Systems Probataire d'Observation de la Terre (SPOT) [28], launched in 1986 and 1990. The image is in the multispectral mode having two bands:

Red of wavelength 0.61–0.68 μm, and
Near infra red of wavelength 0.79–0.89 μm.

Fig. 10 shows the image in the near infra red band. It is known that the region captured has a river (the *Hooghly*) cutting through it, along with several other water bodies. Besides this there are regions belonging to vegetation, habitation, concrete etc. Our aim in this article is to cluster the image (in the two bands) using *GCUK-clustering* so that the landcover types are automatically identified.

Table 2
Actual and computed values for the data sets

| Data set | # clusters | | Center coordinates | |
|---|---|---|---|---|
| | Actual | Computed | Actual | Computed |
| Data_3_2 | 3 | 3 | [(6.2267, 3.5581), (4.0000, 5.3750), (1.9961, 4.0385)] | [(6.2267, 3.5581), (4.0000, 5.3750), (1.9961, 4.0385)] |
| Data_5_2 | 5 | 5 | [(5.8908, 9.8194), (9.8956, 10.1716), (10.0138, 6.3268), (10.2572, 13.9434), (13.9876, 10.1164)] | [(5.8100, 9.7929), (9.6533, 10.4652), (10.0217, 6.4375), (10.2843, 14.0613) (13.8056, 10.0818)] |
| Data_6_2 | 6 | 6 | [(4.9666, 5.0322), (4.9972, 19.9180), (9.0534, 9.0644), (15.9590, 15.9976), (19.8848, 20.2904), (19.9932, 4.8666)] | [(4.9666, 5.0322), (4.9972, 19.9180), (9.0534, 9.0644), (15.9590, 15.9976) (19.8848, 20.2904), (19.9932, 4.8666)] |
| Data_4_3 | 4 | 4 | [(0.1087, 0.0303, −0.0189), (4.7990, 5.2694, 4.9617), (9.8937, 9.9876, 10.1315), (15.0780, 14.9186, 15.1325)] | [(0.1087, 0.0303, −0.0189), (4.7990, 5.2694, 4.9617), (9.8937, 9.9876, 10.1315), (15.0780, 14.9186, 15.1325)] |
| Cancer | 2 | 2 | [(2.9640, 1.3063, 1.4144, 1.3468, 2.1081, 1.3468, 2.0833, 1.2613, 1.0653), (7.1883, 6.5774, 6.5607, 5.5858, 5.3264, 7.6276, 5.9749, 5.8577, 2.6025)] | [(3.0509, 1.2965, 1.4248, 1.3473, 2.0951, 1.3053, 2.0907, 1.250, 1.1128), (7.1645, 6.7792, 6.7186, 5.7316, 5.4632, 7.9264, 6.0952, 6.039, 2.5628)] |

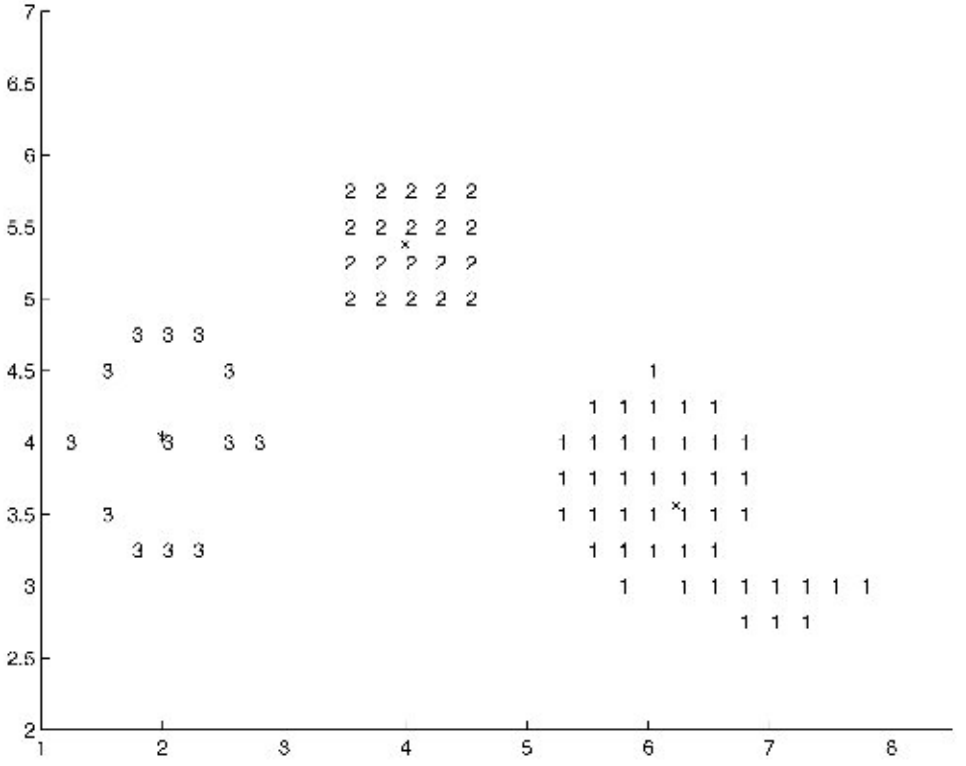

Fig. 6. Clustered Data_3_2 using *GCUK-clustering*. The centres are shown with '*'.

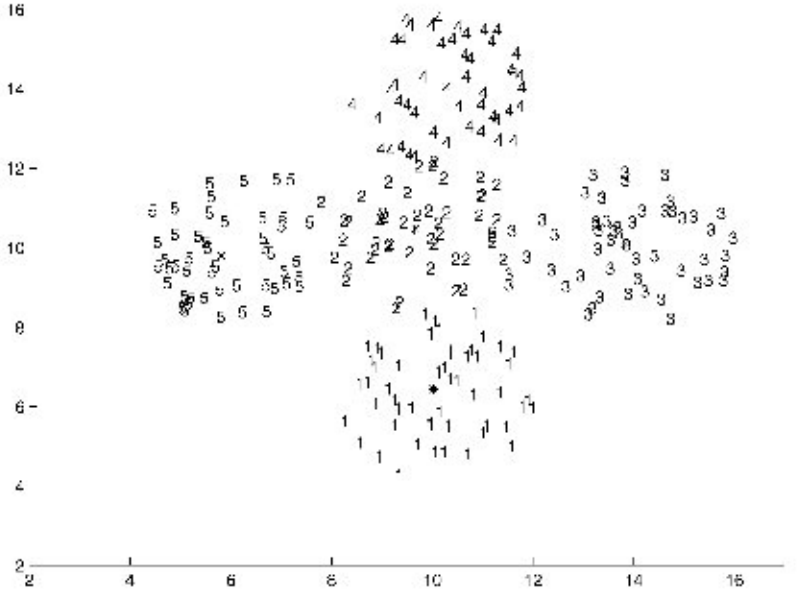

Fig. 7. Clustered Data_5_2 using *GCUK-clustering*. The centres are shown with '*'.

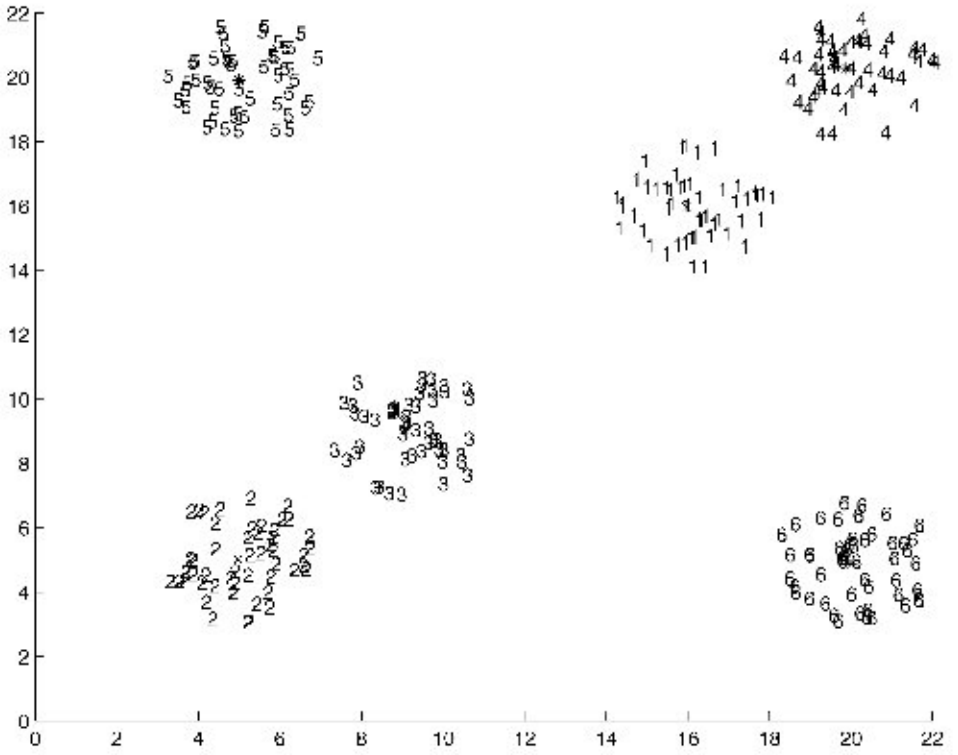

Fig. 8. Clustered Data_6_2 using *GCUK-clustering*. The centres are shown with '*'.

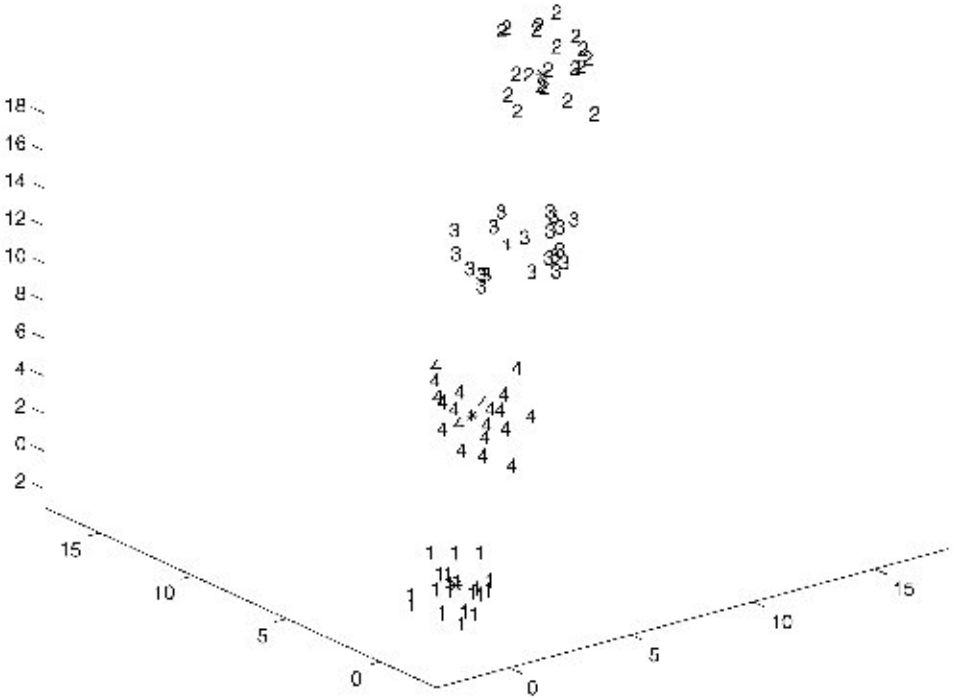

Fig. 9. Clustered Data_4_3 using *GCUK-clustering*. The centres are shown with '*'. (Only 20 points per class is plotted for the sake of clarity.)

Fig. 10. *SPOT* image of Calcutta in the near infra red band (the image is enhanced by a factor of 75%).

Fig. 11. Clustered *SPOT* image of Calcutta using the *GCUK-clustering* method. Three clusters were identified for this image.

The result of the application of the *GCUK-clustering* algorithm to this Calcutta image is provided in Fig. 11. It automatically yielded three clusters, shown in white, grey and black. As seen from the figure, the *Hooghly river* and many other water bodies (obtained in white) are distinctly demarcated from the rest. These water bodies could be verified correctly from the ground facts as belonging to several ponds, lakes, canals and dockyard. The grey region belongs to the vegetation and open space landcover types. Note that its predominance on the left bank of the river is borne out by the ground fact, since this belongs to the rural *Howrah* region. The concentration of the grey region on the right bank of the river near the middle of the image corresponds to the area of *race course* and several open, green regions in the heart of the city. The remaining portions cover mostly the concrete structures, habitation and roads.

## 4. Discussion and conclusions

Clustering is a well known exploratory data analysis tool where the objective is to partition the data into a number of clusters. In most real life situations the number of clusters is not known a priori. In this article, the searching capability of genetic algorithms is exploited for the formulation of clustering techniques for unknown number of clusters. For this purpose, the reciprocal of the DB index, a common cluster validation criteria, has been used for computing the fitness of the chromosomes.

The effectiveness of the clustering technique is demonstrated for several artificial and real life data sets with the number of clusters varying from two to six, and the number of dimensions varying from two to nine. Both overlapping and non-overlapping data sets are considered for this purpose. Another interesting real life application of the *GCUK-clustering* for classifying a *SPOT* image of Calcutta demonstrates that the said method is able to automatically identify several landcover types even when the size of the data set is significantly large (the *SPOT* image had 262144 pixels or data points in 2-D space).

In this article, mutation has been implemented as $v \times (1 \pm 2 \times \delta)$. Other forms like $v \pm (\delta + \varepsilon)v$, where $0 < \varepsilon < 1$ could also have been used. One may note in this context that similar sort of mutation operators for real encoding have been used mostly in the realm of evolutionary strategies (Chapter 8 of Ref. [3]). Although we have used the Davies–Bouldin index in this article for computing the fitness of a chromosome, other indices like Dunn's index and its generalized versions [26], Calinski–Harabasz index [29], C-index [30] etc. may be used for this purpose and their comparative performance can be studied.

Since the cluster centers are real numbers, a natural and conceptually straight forward way of encoding them in a chromosome is by using the floating point representation. This has been implemented in this article. In this context, a binary encoding may be implemented for the same problem, and the results may be compared with the present floating point form.

## 5. Summary of the article

GAs belong to a class of search techniques that mimic the principles of natural selection to develop solutions of large optimization problems. Clustering is a popular unsupervised pattern classification technique which partitions the input space into a number of regions based on some similarity/dissimilarity metric such that similar elements are placed in the same cluster while the dissimilar ones are placed in separate clusters. There are several tasks involved in clustering that require search in large and complex spaces, and therefore the application of GAs for this problem appears to be appropriate and natural.

Several algorithms for clustering data when the number of clusters is known a priori are available in the literature viz., the widely used $K$-means algorithm. However, in most real life situations the number of clusters in a data set is not known a priori. The real challenge in this situation is to be able to automatically evolve a proper value of the number of clusters as well as providing the appropriate clustering.

In this article, we propose a GA based clustering technique, *GCUK-clustering*, which can automatically evolve the appropriate clustering of a data set. The chromosome encodes the centres of a number of clusters, whose value may vary. Modified versions of crossover and mutation operators are used. Cluster validity index like Davies–Bouldin index is utilized for computing the fitness of the chromosomes.

The effectiveness of the clustering technique is demonstrated for several artificial and real life data sets with the number of clusters varying from two to six, and the number of dimensions varying from two to nine. Both overlapping and non-overlapping data sets are considered for this purpose. Another interesting real life application of the *GCUK-clustering* for classifying a *SPOT* image of Calcutta demonstrates that the said method is able to automatically identify several landcover types even when the size of the data set is significantly large (the *SPOT* image had 262,144 pixels or data points in 2-D space).

## References


[1] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.

[2] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[3] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York, 1992.

[4] J.L.R. Filho, P.C. Treleaven, C. Alippi, Genetic algorithm programming environments, IEEE Comput. 27 (1994) 28–43.

[5] D. Whitley, T. Starkweather, C. Bogart, Genetic algorithms and neural networks: optimizing connections and connectivity, Parallel Comput. 14 (1990) 347–361.

[6] S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, 1993.

[7] L.J. Eshelman (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, 1995.

[8] S. Bandyopadhyay, C.A. Murthy, S.K. Pal, Pattern classification using genetic algorithms, Pattern Recog. Lett. 16 (1995) 801–808.

[9] S. Bandyopadhyay, S.K. Pal, U. Maulik, Incorporating chromosome differentiation in genetic algorithms, Inform. Sci. 104 (3/4) (1998) 293–319.

[10] R.P. Dick, N.K. Jha, A multiobjective genetic algorithm for hardware software cosynthesis of distributed embedded systems, IEEE Trans. Comput. Aided Design Integrated Circuits Systems 17 (1998) 920–935.

[11] S.K. Pal, S. Bandyopadhyay, C.A. Murthy, Genetic algorithms for generation of class boundaries, IEEE Trans. Syst., Man, Cybern. 28 (1998) 816–828.

[12] S. Bandyopadhyay, S.K. Pal, Relation Between VGA-classifier and MLP: determination of network architecture, Fundam. Informat. 37 (1999) 177–196.

[13] M.R. Anderberg, Cluster Analysis for Application, Academic Press, New York, 1973.

[14] J.A. Hartigan, Clustering Algorithms, Wiley, New York, 1975.

[15] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, London, 1982.

[16] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[17] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.

[18] W.L.G. Koontz, P.M. Narendra, K. Fukunaga, A branch and bound clustering algorithm, IEEE Trans. Comput. C-24 (1975) 908–915.

[19] J.H. Wolfe, Pattern clustering by multivariate mixture analysis, Multivar. Behav. Res. 5 (1970) 329–350.

[20] W.L.G Koontz, P.M. Narendra, K. Fukunaga, A graph theoretic approach to nonparametric cluster analysis, IEEE Trans. Comput. C-25 (1975) 936–944.

[21] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1990.

[22] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recog. 33 (2000) 1455–1465.

[23] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Patt. Anal. Mach. Intell. 1 (1979) 224–227.

[24] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 3 (1936) 179–188.

[25] H. Spath, Cluster Analysis Algorithms, Ellis Horwood, Chichester, UK, 1989.

[26] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Systs., Man Cyberns. 28 (1998) 301–315.

[27] R. Kothari, D. Pitts, On finding the number of clusters, Patt. Recog. Lett. 20 (1999) 405–416.

[28] J.A. Richards, Remote Sensing Digital Image Analysis: An Introduction, Springer-Verlag, New York, 1993.

[29] R.B. Calinski, J. Harabasz, A dendrite method for cluster analysis, Comm. in Stat. 3 (1974) 1–27.

[30] G.W. Milligan, C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (2) (1985) 159–179.

**About the Author**—SANGHAMITRA BANDYOPADHYAY did her Bachelors in Physics and Computer Science in 1988 and 1991, respectively. Subsequently, she did her Masters in Computer Science from Indian Institute of Technology, Kharagpur in 1993 and Ph.D in Computer Science from Indian Statistical Institute, Calcutta in 1998. Dr. Bandyopadhyay is the first recipient of Dr. *Shanker Dayal Sharma Gold Medal* and *Institute Silver Medal* for being adjudged the best all round post graduate performer in 1994. She has worked in Los Alamos National Laboratory in 1997 as a graduate research assistant and in the University of New South Wales, Sydney, Australia, as a post doctoral fellow. Dr. Bandyopadhyay received the Indian National Science Academy (INSA) and the Indian Science Congress Association (ISCA) *Young Scientist Awards* in 2000. Her research interests include Soft Computation, Pattern Recognition, Parallel and Distributed Systems.

**About the Author**—UJJWAL MAULIK did his Bachelors in Physics and Computer Science in 1986 and 1989, respectively. Subsequently, he did his Masters and Ph.D in Computer Science in 1991 and 1997, respectively, from Jadavpur University, India. Dr. Maulik has visited Center for Adaptive Systems Application, Los Alamos, New Mexico, USA in 1997, and University of New South Wales, Sydney, Australia in 1999. He is a faculty member in the Department of Computer Science and Technology, Kalyani Government Engineering College, India. Currently, Dr Maulik is visiting University of Texas at Arlington, Texas, USA as a *BOYSCAST fellow* of the Department of Science and Technology (DST), Government of India. He has edited a book titled *Intelligent Computing and VLSI*, published by Allied Publishers, New Delhi in 2001. His research interests include Evolutionary Computation and Pattern Recognition, Computer Vision, Parallel and Distributed Systems.