[7] N. Otsu, “A threshold selection method from gray-level histogram,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, pp. 62–66, Jan. 1979.
[8] L. Gupta and T. Sortrakul, “A Gaussian mixture based image segmentation algorithm,” *Pattern Recognit.*, vol. 31, no. 3, pp. 315–325, 1998.
[9] E. R. Dougherty, *An Introduction to Morphological Image Processing*. Bellingham, WA: SPIE, 1992.
[10] L. Gupta, T. Sortrakul, A. Charles, and P. Kisatsky, “Robust automatic target recognition using a localized boundary representation,” *Pattern Recognit.*, vol. 28-10, pp. 1587–1598, 1995.
[11] H. Sakoe and S. Chiba, “Dynamic programming optimization for spoken word recognition,” *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-26, pp. 43–49, Feb. 1978.
[12] L. Gupta, D. L. Molfese, R. Tammana, and P. G. Simos, “Non-linear alignment and averaging for estimating the evoked potential,” *IEEE Trans. Biomed. Eng.*, vol. 43, pp. 348–356, Apr. 1996.
[13] L. Gupta and K. Malakapalli, “Robust partial shape classification using invariant breakpoints and dynamic alignment,” *Pattern Recognit.*, vol. 23-10, pp. 1103–1111, 1990.

Nonparametric Genetic Clustering: Comparison of Validity Indices

Sanghamitra Bandyopadhyay and Ujjwal Maulik

***Abstract*—Variable string length genetic algorithm (GA) is used for developing a novel nonparametric clustering technique when the number of clusters is not fixed *a priori*. Chromosomes in the same population may now have different lengths since they encode different number of clusters. The crossover operator is redefined to tackle the concept of variable string length. Cluster validity index is used as a measure of the fitness of a chromosome. The performance of several cluster validity indices, namely, Davies–Bouldin (DB) index, Dunn's index, two of its generalized versions and a recently developed index, in appropriately partitioning a data set, are compared.**

***Index Terms*—Clustering, cluster validity, Davies–Bouldin (DB) index, generalized Dunn's index, genetic algorithms (GAs), pattern recognition.**

I. Introduction

Genetic algorithms (GAs) [1], [2] are randomized search and optimization techniques guided by the principles of evolution and natural genetics, and have a large amount of implicit parallelism. They provide near optimal solutions of an objective or fitness function in complex, large, and multimodal landscapes. In GAs the parameters of the search space are encoded in the form of strings (or, *chromosomes*). A *fitness* function is associated with each string that represents the degree of *goodness* of the solution encoded in it. Biologically inspired operators like *selection*, *crossover*, and *mutation* are used over a number of generations for generating potentially better strings.

Clustering [3], [4] is a popular unsupervised pattern classification technique which partitions the input space into K regions based on some similarity/dissimilarity metric where the value of K may or may not be known *a priori*. The aim of any clustering technique is to evolve a partition matrix $U(X)$ of the given data set X (consisting of, say, n patterns, $X = \{x_1, x_2, \ldots, x_n\}$) such that

$$\sum_{j=1}^{n} u_{kj} \geq 1 \quad \text{for } k = 1, \ldots, K$$

$$\sum_{k=1}^{K} u_{kj} = 1 \quad \text{for } j = 1, \ldots, n \quad \text{and}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{n} u_{kj} = n.$$

The partition matrix $U(X)$ of size $K \times n$ may be represented as $U = [u_{kj}]$, $k = 1, \ldots, K$ and $j = 1, \ldots, n$, where u_{kj} is the membership of pattern x_j to cluster C_k. In crisp partitioning $u_{kj} = 1$ if $x_j \in C_k$; otherwise $u_{kj} = 0$. Cluster validity is a measure associated with different partitions that indicates their relative goodness.

In most real-life situations the number of clusters in a data set is not known beforehand. The real challenge is to be able to automatically evolve a proper value of the number of clusters and provide the appropriate clustering under this circumstance. Some attempts in this regard can be found in [4] and [5]. The ISODATA algorithm [4] uses a combination of splitting, merging and deleting clusters to adjust the number of cluster centers. Each of these operations depends on several user supplied parameters, which are often very difficult to estimate *a priori*. Recently, Ravi and Gowda [5] used a distributed GA based on the ISODATA technique for clustering symbolic objects. However, this method also suffers from the same limitations as present in the ISODATA clustering technique.

Our aim in this paper is to develop a nonparametric clustering technique which will not assume any particular underlying distribution of the data set, while it will be able to evolve a proper value of number of clusters as well as provide the appropriate clustering automatically. Variable string length genetic algorithm (VGA) [6], with real encoding of the cluster centers in the chromosome [7], is used as the underlying search tool for this purpose. Several cluster validity indices viz., Davies–Bouldin (DB) index [8], Dunn's index [9], two of its generalized versions [10], and a newly developed validity index are utilized for computing the fitness of the chromosomes. The results provide a comparison of these indices in terms of their utility in determining the appropriate clustering of the data. Several artificial and real-life data sets with different characteristics are used for performing the experiments.

II. Clustering Using Variable String Length Genetic Algorithms

In this section, we describe the use of VGAs for automatically clustering a data set. This involves determination of the number of clusters as well as the appropriate clustering of the data. The technique, described below, is subsequently referred to as the *VGA-clustering* (VGA-based clustering).

String Representation and Population Initialization: In *VGA-clustering*, the chromosomes are made up of real numbers (representing the coordinates of the centers). If chromosome i encodes the centers of K_i clusters in N dimensional space $K_i \geq 2$, then its length l_i is taken to be $N * K_i$.

Each string i in the population initially encodes the centers of a number K_i of clusters, where K_i is given by $K_i = \text{rand}(\,) \bmod K^*$. Here, $\text{rand}(\,)$ is a function returning an integer, and K^* is a soft estimate of the upper bound of the number of clusters. Note that K^* is used only for the generation of the initial population. The actual number of clusters in the data set is not related to K^*, and may be any number greater than, equal to or less than K^*. The K_i centers encoded in a chromosome are randomly selected points from the data set.

Fitness Computation: The different cluster validity indices that have been utilized for computing the fitness of chromosomes are described in the next section. For each chromosome, the centers encoded in it are first extracted, and then a partition is obtained by assigning the points to a cluster corresponding to the closest center. The cluster centers encoded in the chromosome are then replaced by the centroids of the corresponding clusters. Given the above partition, and the number of clusters, the value of the cluster validity index is computed. The fitness of a chromosome is then defined as a function of the corresponding cluster validity index.

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. We have used the roulette wheel strategy for implementing the proportional selection scheme.

Crossover: For the purpose of crossover, the cluster centers are considered to be indivisible, i.e., the crossover points can only lie in between two clusters centers. The crossover operator, applied stochastically with probability μ_c, must ensure that information exchange takes place in such a way that both the offspring encode the centers of at least two clusters. For this, the operator is defined as follows.

Let parents P_1 and P_2 encode K_1 and K_2 cluster centers, respectively. $\mathcal{C}_1$, the crossover point in P_1, is generated as $\mathcal{C}_1 = \mathrm{rand}(\ ) \bmod K_1$. As before, $\mathrm{rand}(\ )$ is a function that returns an integer. Let $\mathcal{C}_2$ be the crossover point in P_2, and it may vary in between $[\mathrm{LB}(\mathcal{C}_2), \mathrm{UB}(\mathcal{C}_2)]$, where $\mathrm{LB}(\ )$ and $\mathrm{UB}(\ )$ indicate the lower and upper bounds of the range of $\mathcal{C}_2$, respectively. $\mathrm{LB}(\mathcal{C}_2)$ and $\mathrm{UB}(\mathcal{C}_2)$ are given by

$$\mathrm{LB}(\mathcal{C}_2) = \min[2, \max[0, 2 - (K_1 - \mathcal{C}_1)]] \quad \text{and} \tag{1}$$

$$\mathrm{UB}(\mathcal{C}_2) = [K_2 - \max[0, 2 - \mathcal{C}_1)]]. \tag{2}$$

Therefore, $\mathcal{C}_2$ is given by

$$\mathcal{C}_2 = \mathrm{LB}(\mathcal{C}_2) + \mathrm{rand}(\ ) \bmod (\mathrm{UB}(\mathcal{C}_2) - \mathrm{LB}(\mathcal{C}_2)).$$

Mutation: Each chromosome undergoes mutation with a fixed probability μ_m. Since floating point representation is considered in this paper, we use the following mutation. A number δ in the range [0, 1] is generated with uniform distribution. If the value at a gene position is v, after mutation it becomes $(1 \pm 2 * \delta) * v$, when $v \neq 0$, and $\pm 2 * \delta$, when $v = 0$. The '$+$' or '$-$' sign occurs with equal probability.

III. Some Cluster Validity Indices

This section contains the description of several cluster validity indices that have been used for computing the fitness of the chromosomes in the *VGA-clustering* scheme.

A. DB Index [8]

This index is a function of the ratio of the sum of *within-cluster scatter* to *between-cluster separation*. The scatter within the ith cluster is computed as

$$S_{i,q} = \left(\frac{1}{|C_i|}\sum_{x\in C_i}\{\|x - z_i\|_2^q\}\right)^{1/q} \tag{3}$$

and the distance between cluster C_i and C_j is defined as

$$d_{ij,t} = \|z_i - z_j\|_t. \tag{4}$$

$S_{i,q}$ is the qth root of the qth moment of the $|C_i|$ points in cluster C_i with respect to their mean z_i, and is a measure of the dispersion of the points in the cluster. 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 z_i and z_j that characterize clusters C_i and C_j. Subsequently, we compute

$$R_{i,qt} = \max_{j, j\neq i}\left\{\frac{S_{i,q} + S_{j,q}}{d_{ij,t}}\right\}. \tag{5}$$

The Davies–Bouldin DB index is then defined as

$$DB = \frac{1}{K}\sum_{i=1}^{K} R_{i,qt}. \tag{6}$$

The objective is to minimize the DB index for achieving proper clustering. Therefore, the fitness of 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.

B. Dunn's Index [9]

Let S and T be two nonempty subsets of $\mathcal{R}^N$. Then the diameter $\triangle$ of S and set distance δ between S and T are

$$\triangle(S) = \underbrace{\max}_{x,y\in S}\{d(x,y)\} \quad \text{and}$$

$$\delta(S,T) = \underbrace{\min}_{x\in S, y\in T}\{d(x,y)\}$$

where $d(x,y)$ is the distance between points x and y. For any partition, Dunn defined the following index:

$$\nu_D = \underbrace{\min}_{1\le i\le K}\left\{\underbrace{\min}_{1\le j\le K, j\neq i}\left\{\frac{\delta(C_i, C_j)}{\underbrace{\max}_{1\le k\le K}\{\triangle(C_k)\}}\right\}\right\}. \tag{7}$$

Larger values of ν_D correspond to good clusters, and the number of clusters that maximizes ν_D is taken as the optimal number of clusters.

C. Generalized Dunn's Index ν_{GD} [10]

The generalized Dunn's index was developed in [10] after demonstrating the sensitivity of the original index ν_D, given by (7), to changes in cluster structure, since not all of the data was involved in the computation of the index. Let δ_i be any positive, semi-definite, symmetric set distance function and $\triangle_j$ be any positive, semi-definite diameter function. Then the generalized Dunn's index, ν_{GD} is defined as

$$\nu_{\mathrm{GD}} = \underbrace{\min}_{1\le s\le K}\left\{\underbrace{\min}_{1\le t\le K, t\neq s}\left\{\frac{\delta_i(C_s, C_t)}{\underbrace{\max}_{1\le k\le K}\{\triangle_j(C_k)\}}\right\}\right\}. \tag{8}$$

Five set distance functions and three diameter functions are defined in [10]. Of these, we have used two combinations δ_3 and $\triangle_3$ (which is recommended in [10] as being most useful for cluster validation) in one and δ_5 and $\triangle_3$ in the other. The three measures viz., δ_3, δ_5, and $\triangle_3$, are defined as follows:

$$\triangle_3(S) = 2\left(\frac{\sum_{x\in S} d(x, z_S)}{|S|}\right) \tag{9}$$

$$\delta_3(S,T) = \frac{1}{|S||T|}\sum_{x\in S, y\in T} d(x,y) \quad \text{and} \tag{10}$$

$$\delta_5(S,T) = \frac{1}{|S| + |T|}\left(\sum_{x\in S} d(x, z_T) + \sum_{y\in T} d(y, z_S)\right). \tag{11}$$

Here $z_S = (1/|S|)\sum_{x\in S} x$, and $z_T = (1/|T|)\sum_{y\in T} y$. The two generalized Dunn's indices ν_{33} and ν_{53} are generated by replacing $\triangle_i$

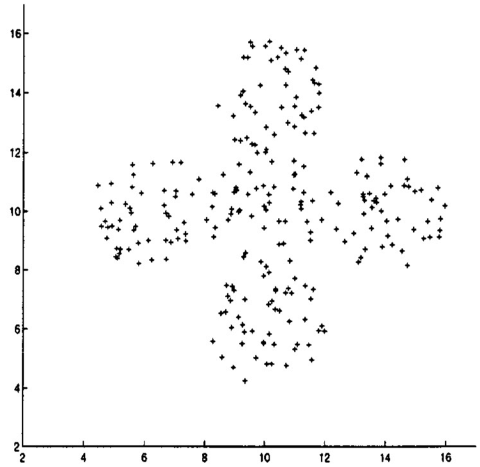

Fig. 1. *AD_5_2*.

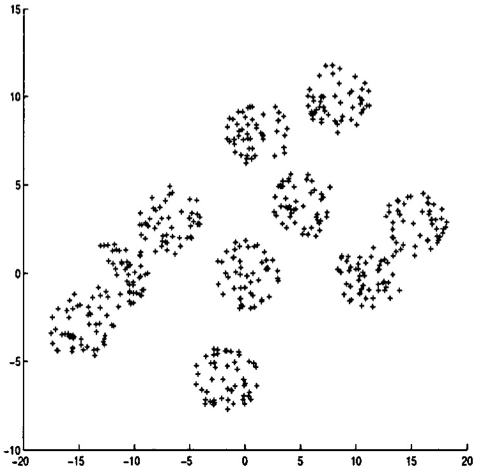

Fig. 2. *AD_10_2*.

by $\triangle_3$ (in the denominator) and δ_i by δ_3 and δ_5, respectively (in the numerator) in (8).

D. Results

The *VGA-clustering* algorithm is applied on the real and artificial data sets described below.

The artificial data sets are *AD_5_2*, *AD_10_2*, and *AD_4_3*, where the first two data sets are in two dimensions with five and ten clusters, respectively, and the last one is in three dimensions with four clusters. Figs. 1–3 show the three data sets. Table I presents the number of points, dimensions, and the number of clusters in each data. Note that the artificial data sets are generated in such a way that they present different degrees of difficulty for clustering. For example, although *AD_4_3* (Fig. 3) appears to be visually simple, the presence of noisy points in it (situated exactly between two distinct clusters) can mislead the clustering technique, and/or the validity indices. On the other hand, *AD_5_2* can be seen to be highly overlapped, and *AD_10_2* is overlapping with large number of clusters.

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

Iris Data: This data represents different categories of irises characterized by four feature values [11]. 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.

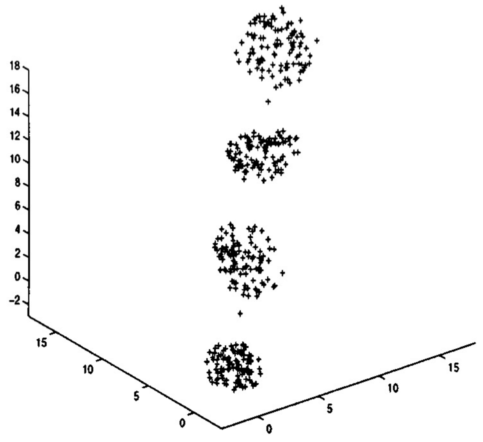

Fig. 3. *AD_4_3*.

TABLE I
DESCRIPTION OF THE DATA SETS

Name	No. of points	No. of clusters	No. of dim.	Points per clusters
AD_5_2	250	5	2	50
AD_10_2	500	10	2	50
AD_4_3	402	4	3	101,101, 100,100
Iris	150	3	4	50
Cancer	683	2	9	444,239

Breast Cancer: Here we use the Wisconsin Breast Cancer data set consisting of 683 sample points, 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.

VGA-clustering is implemented with the following parameters: $\mu_c = 0.8, \mu_m = 0.05$, and $K^* = 20$ (see Section II). The population size is taken to be 50 and a maximum of 50 iterations are executed.

As can be seen from Fig. 5 for *AD_5_2* data, while the two generalized versions of the Dunn's index identify, incorrectly, four clusters to be appropriate, the *DB* and Dunn's index correctly identify five clusters in the data (see Fig. 4).

AD_10_2 is generated with ten physical clusters, some of which are compact and separated, while others are overlapping. The clusterings obtained when *DB* and Dunn's indices are used are shown in Fig. 6 when both of them provided eight clusters (resulting from merging of some physically separate clusters to yield cluster numbers 2 and 5 in the figure). Interestingly, and quite unexpectedly, both ν_{33} and ν_{53} provided two clusters, which are shown in Fig. 7. Since this indicates a gross deviation from the actual scenario, we investigated the values of the indices for different partitionings of *AD_10_2*, including the one that we know to be correct, i.e., with ten clusters. It was found that the values of ν_{33} were 0.986 103, 1.421 806, and 1.388 861 0 for the clusterings in Figs. 6 (into eight clusters) and 7 (into two clusters), and the correct partitioning into ten clusters, respectively. Similarly, the values of ν_{53} were 0.963 560, 1.367 821, and 1.347 380, respectively, for the three cases. Thus, among the partitionings that were investigated, the values of ν_{33} and ν_{53} actually got maximized for the case of two clusters (i.e., corresponding to Fig. 7). This indicates that it is not the limitation of the *VGA-clustering* technique which was responsible for providing two

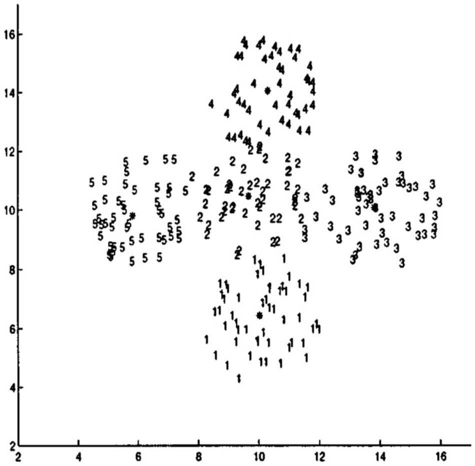

Fig. 4. *AD_5_2* clustered into five clusters when *DB* and ν_D index are used for computing fitness. The centers are shown with '$*$'.

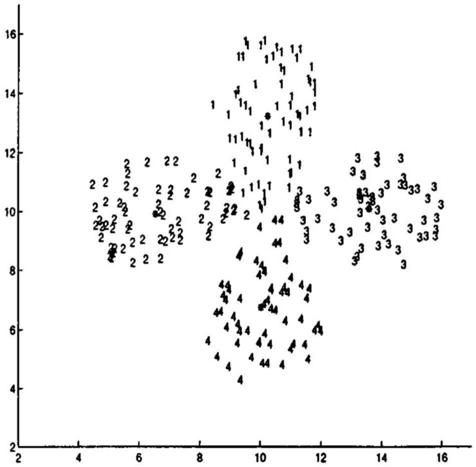

Fig. 5. *AD_5_2* clustered into four clusters when ν_{33} and ν_{53} are used for computing fitness. The centers are shown with '$*$'.

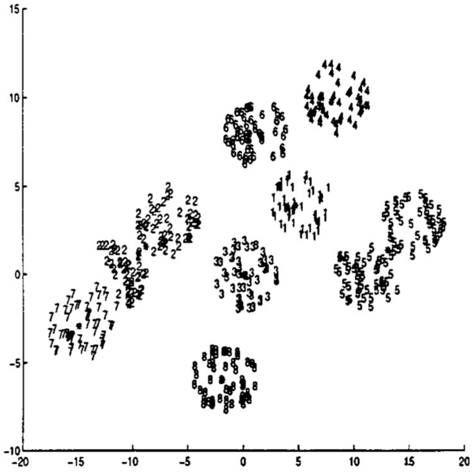

Fig. 6. *AD_10_2* clustered into eight clusters when *DB* and ν_D indices are used for computing fitness. The centers are shown with '$*$'.

clusters; on the contrary, this is the result of a limitation of the index definition.

Fig. 8 shows that the genetic clustering with *DB* index, ν_{33} and ν_{53}, can correctly evolve the appropriate partitioning for *AD_4_3*, while the

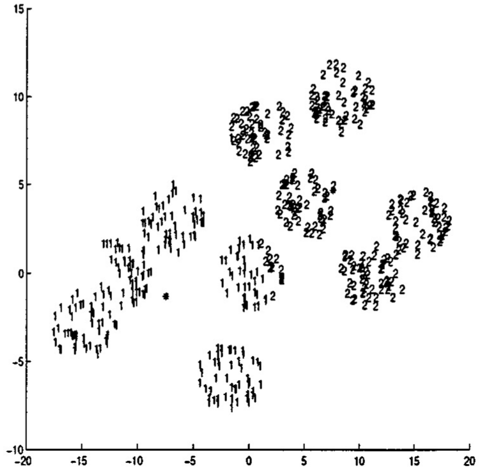

Fig. 7. *AD_10_2* clustered into two clusters when ν_{33} and ν_{53} indices are used for computing fitness. The centers are shown with '$*$'.

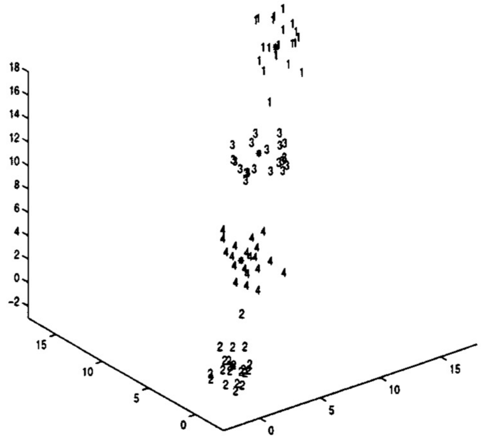

Fig. 8. *AD_4_3* clustered into four clusters when *DB*, ν_{33}, and ν_{53} are used for computing fitness. The centers are shown with '$*$'. (Only 20 points per class are plotted for the sake of clarity.)

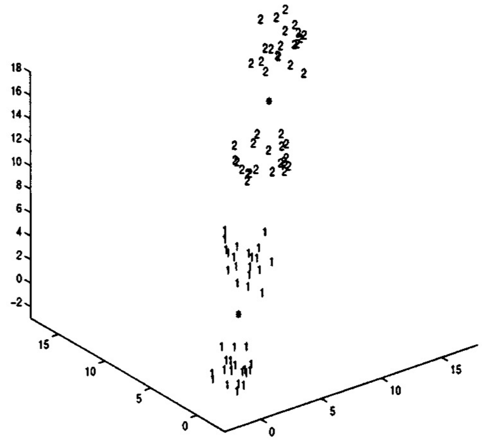

Fig. 9. *AD_4_3* clustered into two clusters when ν_D is used for computing fitness. The centers are shown with '$*$'. (Only 20 points per class are plotted for the sake of clarity.)

original Dunn's index ν_D fails in this regard (Fig. 9). On investigation it was found that ν_D for two clusters is larger than that with four clusters, and hence the former is preferred over the latter. Note that this problem

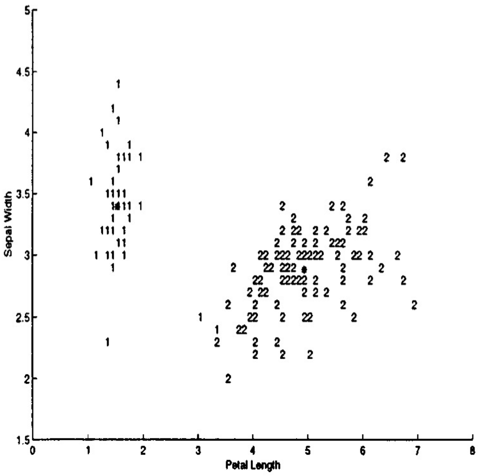

Fig. 10. *Iris* data clustered into two clusters when *DB* index is used for computing fitness. Centers are shown with '$*$'.

TABLE II
ACTUAL AND COMPUTED NUMBER OF CLUSTERS FOR THE DATA SETS

Data Set	Actual # clusters	# clusters with VGA clustering using				
		DB	ν_D	ν_{33}	ν_{53}	$\mathcal{I}(K)$
AD_5_2	5	5	5	4	4	5
AD_10_2	10	8	8	2	2	10
AD_4_3	4	4	2	4	4	4
Iris	3	2	2	2	2	3
Cancer	2	2	2	2	2	2

arises since all the points are not taken into consideration when computing ν_D by (7). This limitation is removed in the two generalized versions of the Dunn's index used in this paper.

For *Iris*, the *VGA-clustering* always provides two clusters irrespective of the validity index used for computing the fitness. Although it is known that the data has three physical classes, two of them have a significant amount of overlap. Thus many automatic clustering methods reported in the literature have often provided two clusters for this data [10], [12]. Fig. 10 shows for the purpose of demonstration the *Iris* data set partitioned into two clusters when *DB* index is used for computing the fitness. As seen from Table II (columns 3–6), the *VGA-clustering* produces two clusters in all the cases for *Cancer* data.

Table II shows, in a nutshell, the actual number of clusters, and that provided by the genetic clustering when the different indices are used for computing the chromosome fitness. The *DB* index provided the correct number of clusters in three out of the five cases, while ν_D, ν_{33}, and ν_{53} do so in two cases each.

Fig. 11 shows the variation of the number of clusters (of the best chromosome) deemed to be optimal when *DB* index is used with the number of generations while Fig. 12 shows the variation of the value of $(1/DB\ index)$ of the best chromosome with the number of generations. The figures show that although the final number of clusters $(=8)$ is attained by the *VGA-clustering* with *DB* index in generation seven, the appropriate partitioning keeps on evolving till around generation 30.

IV. NEW CLUSTER VALIDITY INDEX

In this section we describe a new index $\mathcal{I}$ developed recently by the authors. It is defined as follows:

$$\mathcal{I}(K) = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_K\right)^p \tag{12}$$

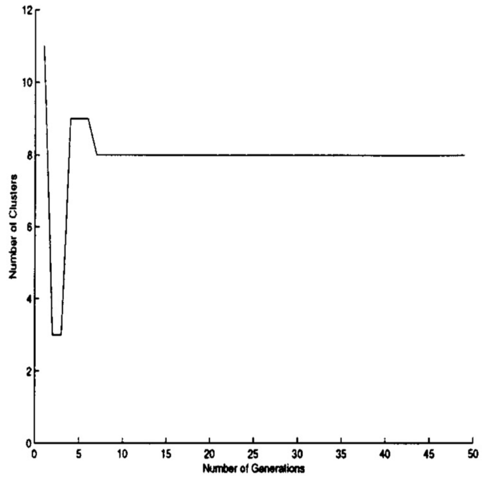

Fig. 11. Variation of the number of clusters with the number of generations for *AD_10_2* using *DB* index.

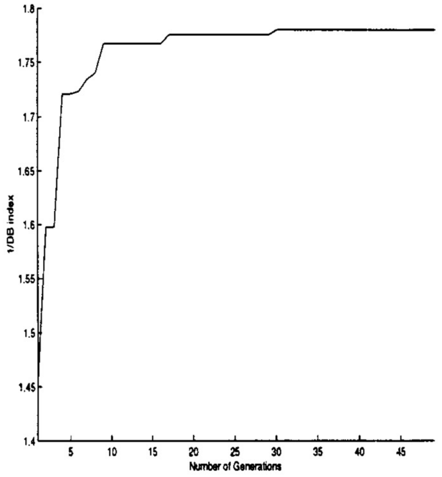

Fig. 12. Variation of the value of $(1/DB\ index)$ with the number of generations for *AD_10_2*.

where K is the number of clusters and p is any real number greater than or equal to 1. Note that p controls the contrast between different cluster configurations. Here

$$E_K = \sum_{k=1}^{K} \sum_{j=1}^{n} u_{kj} \|x_j - z_k\| \quad \text{and} \tag{13}$$

$$D_K = \max_{i,j=1}^{K} \|z_i - z_j\|. \tag{14}$$

n is the total number of points in the data set, $U(X) = [u_{kj}]_{K \times n}$ is a partition matrix for the data, and z_k is the center of the kth cluster.

As can be seen from (12), the new index $\mathcal{I}$ is a composition of three factors, namely, $1/K$, E_1/E_K, and D_K. The first factor decreases linearly as K increases. Therefore, this factor will try to reduce $\mathcal{I}$ as K is increased. The second factor consists of the ratio of E_1, which is constant for a given data set, and E_K, which decreases with increase in K. Hence, because of this term, $\mathcal{I}$ increases as E_K decreases. This, in turn, indicates that formation of more number of clusters, which are compact in nature, would be encouraged. Note that although the choice of E_1 does not affect the performance of $\mathcal{I}(K)$, it is used as some sort of normalizing factor in order to avoid extremely low values of the index. Finally, the third factor (which measures the maximum separation between two clusters) will increase with the value of K. However, note that this value is upper bounded by the maximum separation between

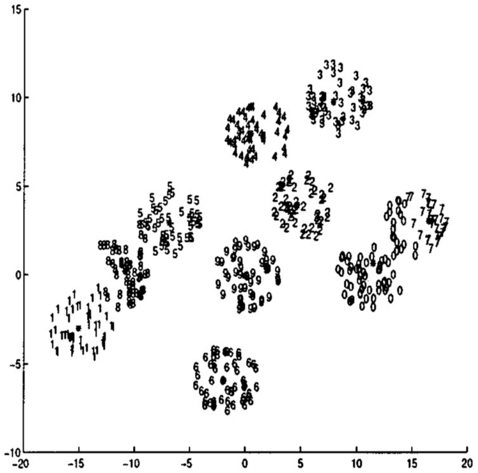

Fig. 13. *AD_10_2* clustered into ten clusters when $\mathcal{I}$ is used for computing fitness. The centers are shown with '$*$'.

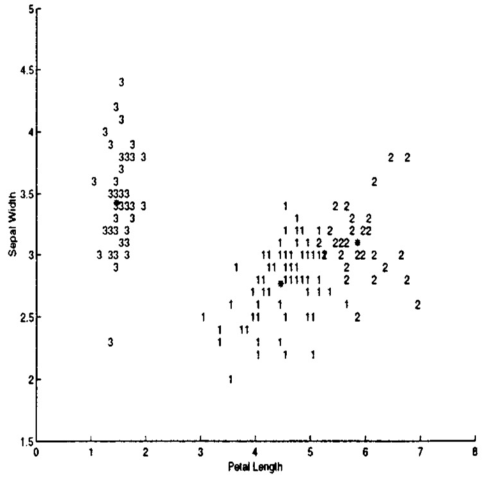

Fig. 14. *Iris* Data clustered into three clusters when $\mathcal{I}$ is used for computing fitness. The centers are shown with '$*$'.

two points in the data set. Thus, the three factors are found to compete with and balance each other critically. While the first factor will try to decrease K, the second and third factors will try to increase K (thereby encouraging formation of compact and well separated clusters).

A. Results of Genetic Clustering Using $\mathcal{I}(K)$

The variable string genetic clustering technique where the fitness is computed using the new index $\mathcal{I}$ is applied on the data sets that have already been described in the previous section. Table II (column 7) presents the number of clusters computed by the *VGA-clustering* when $\mathcal{I}(K)$ is used for computing the fitness of the chromosomes. We have kept p equal to 2. Encouragingly, in this case, the correct number of clusters is found for all the data sets; thereby indicating the significant superiority of the new index *vis-a-vis* the ones described in Section III.

Figs. 13 and 14 show the clusterings obtained for *AD_10_2* and *Iris*. Note that we are showing only the above two cases, since none of the indices mentioned in Section III have been able to provide the correct partitioning for these data sets.

V. Discussion and Conclusion

In this paper the searching capability of VGAs is exploited for the formulation of a clustering methodology when the number of clusters is not known *a priori*. The chromosomes encode the centers of the clusters. Instead of using conventional binary chromosomes, we have used real numbers to represent the cluster centers since it is conceptually closest to our problem domain. The crossover and mutation operators are newly defined for tackling variable string lengths and real encoding. DB index, Dunn's index, and its two generalized versions are used for measuring the goodness of a set of clusters.

In this context, a new cluster validity index is also described whose maximum value across the hierarchy indicates the optimum number of clusters. Use of this index encourages the formation of compact and separated clusters, while attempting to reduce the number of clusters. Comparative study of the different validity indices, when used in the *VGA-clustering* technique, for automatically clustering a data set demonstrates the significant superiority of the new index with respect to the other ones. Several artificial and real-life data sets with the number of dimensions varying from two to nine, and the number of clusters varying from two to ten have been considered. The new index is found to provide the correct number of clusters for all the data sets, while the other ones sometimes fail to do so. In this regard an extensive study, from both theoretical and empirical points of view, needs to be undertaken with respect to the new validity index as well as VGAs. The clustering algorithm may also be compared with other popular methods. Such investigations are currently being undertaken by the authors.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[2] *Handbook of Genetic Algorithms*, L. Davis, Ed., Van-Nostrand, NY, 1991.

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

[4] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Reading, MA: Addison-Wesley, 1974.

[5] T. V. Ravi and K. C. Gowda, "An ISODATA clustering procedure for symbolic objects using a distributed genetic algorithm," *Pattern Recognit. Lett.*, vol. 20, pp. 659–666, 1999.

[6] D. E. Goldberg, K. Deb, and B. Korb, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Syst.*, vol. 3, pp. 493–530, 1989.

[7] U. Maulik and S. Bandyopadhyay, "Genetic classifier based clustering technique," *Pattern Recognit.*, vol. 33, pp. 1455–1465, 2000.

[8] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 224–227, 1979.

[9] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, pp. 32–57, 1973.

[10] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 301–315, June 1998.

[11] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 3, pp. 179–188, 1936.

[12] R. Kothari and D. Pitts, "On finding the number of clusters," *Pattern Recognit. Lett.*, vol. 20, pp. 405–416, 1999.