Abstract—Recently, the problem of clustering categorical data, where no natural ordering among the elements of a categorical attribute domain can be found, has been gaining significant attention from researchers. With the growing demand for categorical data clustering, a few clustering algorithms with focus on categorical data have recently been developed. However, most of these methods attempt to optimize a single measure of the clustering goodness. Often, such a single measure may not be appropriate for different kinds of datasets. Thus, consideration of multiple, often conflicting, objectives appears to be natural for this problem. Although we have previously addressed the problem of multiobjective fuzzy clustering for continuous data, these algorithms cannot be applied for categorical data where the cluster means are not defined. Motivated by this, in this paper a multiobjective genetic algorithm-based approach for fuzzy clustering of categorical data is proposed that encodes the cluster modes and simultaneously optimizes fuzzy compactness and fuzzy separation of the clusters. Moreover, a novel method for obtaining the final clustering solution from the set of resultant Pareto-optimal solutions in proposed. This is based on majority voting among Pareto front solutions followed by $k$-nn classification. The performance of the proposed fuzzy categorical data-clustering techniques has been compared with that of some other widely used algorithms, both quantitatively and qualitatively. For this purpose, various synthetic and real-life categorical datasets have been considered. Also, a statistical significance test has been conducted to establish the significant superiority of the proposed multiobjective approach.

Index Terms—Categorical attributes, fuzzy clustering, multiobjective genetic algorithm, Pareto optimality.

I. INTRODUCTION

CLUSTERING [1]–[3] is a popular unsupervised pattern-classification approach in which a given dataset is partitioned into a number of distinct groups based on some similarity/dissimilarity measures. If each data point is assigned to a single cluster, then the clustering is called crisp clustering. On the other hand, if a data point has certain degrees of belongingness to each cluster, the partitioning is called fuzzy.

Most of the clustering algorithms are designed for datasets where the dissimilarity between any two points of the dataset can be computed using standard distance measures such as Euclidean distance. However, many real-life datasets are categorical in nature, where no natural ordering can be found among the elements in the attribute domain. In such situations, the clustering algorithms, such as K-means [1], fuzzy C-means (FCM) [4], etc., cannot be applied. The K-means algorithm computes the center of a cluster by computing the mean of the set of feature vectors belonging to that cluster. However, as categorical datasets do not have any inherent distance measure, computing the mean of a set of feature vectors is meaningless. A variation of the K-means algorithm, namely partitioning around medoids (PAM) or K-medoids [3], has been proposed for such datasets. In PAM, instead of the cluster center, the cluster medoid, i.e., the most centrally located point in a cluster, is determined. Unlike cluster center, a cluster medoid must be an actual data point. Another extension of the K-means is the K-modes, algorithm [5], [6]. Here, the cluster centroids are replaced by cluster modes (described later). A fuzzy version of the K-modes algorithm, i.e., fuzzy K-modes, is also proposed in [7]. Recently, a Hamming distance (HD) vector-based categorical data clustering algorithm (CCDV) has been developed in [8]. Hierarchical algorithms, such as average linkage [1], are also widely used to cluster categorical data. Some other developments in this area are available in [9]–[11]. However, all these algorithms rely on optimizing a single objective to obtain the partitioning. A single objective function may not work uniformly well for different kinds of categorical data. Hence, it is natural to consider multiple objectives that need to be optimized simultaneously.

Genetic algorithms (GAs) [12]–[14] are popular search and optimization strategies guided by the principle of Darwinian evolution. Although genetic algorithms have been previously used in data clustering problems [15]–[17], as earlier, most of them use a single objective to be optimized, which is hardly equally applicable to all kinds of datasets. To solve many real-world problems, it is necessary to optimize more than one objective simultaneously. Clustering is an important real-world problem, and different clustering algorithms usually attempt to optimize some validity measure such as the compactness of the clusters, separation among the clusters, or a combination of both. (The problem of clustering categorical data poses an additional level of complexity because it is not possible to define the mean of a cluster.) However, as the relative importance of different clustering criteria is unknown, it is better to optimize
compactness and separation separately rather than combining them into a single measure to be optimized. Motivated by this fact, in this paper, the problem of fuzzy partitioning of categorical dataset is modeled as one of multiobjective optimizations (MOOs) [13], [18]–[20], where search is performed over a number of often conflicting objective functions. Multiobjective genetic algorithms (MOGAs) are used in this regard in order to determine the appropriate cluster centers (modes) and the corresponding partition matrix. Non-dominated sorting GA-II (NSGA-II) [21], which is a popular elitist MOGA, is used as the underlying optimization method. The two objective functions, i.e., the global fuzzy compactness of the clusters and fuzzy separation, are optimized simultaneously. Unlike single objective optimization, which yields a single best solution, in MOO the final solution set contains a number of Pareto-optimal solutions, none of which can be dominated or further improved on any one objective without degrading another [13], [18]. This paper also proposes a novel method for selecting the final clustering solution from the set of Pareto-optimal solution based on majority voting among the Pareto front solutions, followed by $k$-nn classification.

Multiobjective optimization has recently been gaining popularity. There are some instances in the literature that applied multiobjective techniques for data clustering. One of the earliest approaches in this field is found in [22], where objective functions representing compactness and separation of the clusters were optimized in a crisp clustering context and with a deterministic method. In [23], a tabu search-based multiobjective clustering technique has been proposed, where the partitioning criteria are chosen as the within-cluster similarity and between-cluster dissimilarity. This technique uses solution representation based on cluster centers, as in [15]. However, experiments are mainly based on artificial distance matrices. A series of works on multiobjective clustering has been proposed in [24]–[26], where the authors have adopted chromosome encoding of length equal to the number of data points. The two objectives that were optimized are overall deviation (compactness) and connectivity. The algorithm in [24] is capable of handling categorical data, whereas the other two papers deal with numeric and continuous datasets. These methods have advantages that they can automatically evolve the number of clusters and can also be used to find non-convex shaped clusters. It may be noted that the chromosome length in these works is equal to the number of points to be clustered. Hence, as discussed in [27], when the length of the chromosomes becomes equal to the number of points $n$ to be clustered, the convergence becomes slower for the large values of $n$. This is due to the reason that the chromosomes, and hence the search space, in such cases become large. However, in [25], a special mutation operator is used to reduce the effective search space by maintaining a list of $L$ nearest neighbors for each data point, where $L$ is a user-defined parameter. This allows faster convergence of the algorithm toward the global Pareto optimal front, making it scalable for larger datasets. The algorithm needs to compute the cluster means, which is computationally less costly than computation of cluster modes, to find the value of one of the objective functions (overall cluster deviation). Moreover, this algorithm uses special initialization routines based on the minimum spanning tree method and is intended for crisp clustering of continuous data. In contrast, the method proposed in this paper uses a center (mode) based encoding strategy for fuzzy clustering of the categorical data. The computation of the cluster modes is costlier than that of the cluster means, and the algorithm needs to compute the fuzzy membership matrices that takes a reasonable amount of time. However, as fuzzy clustering is better equipped to handle overlapping clusters [28], the proposed technique can handle both overlapping and non-overlapping clusters. The experimental results also indicate that the incorporation of fuzziness significantly improves the performance of clustering.

In the context of multiobjective fuzzy clustering, in [29], a multiobjective evolutionary technique has been proposed that integrates NSGA-II with FCM clustering to simultaneously reduce the dimensionality and find the best partitioning. However, this method does not use NSGA-II in the clustering step directly (where FCM is used in its traditional form). NSGA-II is used on the upper level to determine the features to be selected as well as the parameters of FCM. Moreover, this method is only applicable for continuous numeric datasets, not for categorical data. In [19] and [30], we have addressed the problem of multiobjective fuzzy clustering using NSGA-II with a similar center-based encoding technique. These algorithms optimize two cluster validity measures, namely, FCM error function $J_{m}$ [4] and Xie-Beni ($XB$) index [31]. The selection of the solution from the final non-dominated set has been done using a third cluster validity measure, such as $T$ index [2] or Silhouette index [32], and thus, it is sensitive to the choice of the third validity measure. Most importantly, these techniques can only be applied for clustering continuous data, such as remote sensing imagery [19] and microarray gene expression data [30], and cannot be applied for clustering categorical data where the cluster means are not defined.

The main contribution of the present paper is that it proposes a fuzzy multiobjective algorithm for clustering categorical data. As far as our knowledge goes, none of the previous works has addressed the issue of multiobjective fuzzy clustering in the categorical domain. Unlike the works in [19] and [30], where chromosomes encode the cluster means (centers), here, the chromosomes encode the cluster modes, and hence, they differ in the chromosome updation process. Two fuzzy objective functions, viz., fuzzy compactness and fuzzy separation, have been simultaneously optimized resulting in a set of non-dominated solutions. Subsequently, a novel technique based on majority voting among the non-dominated Pareto-optimal solutions followed by $k$-nn classification is proposed to obtain the final clustering solution from the set of non-dominated solutions. Thus, the requirement of the third cluster validity measure for selecting the final solution from the Pareto-optimal set and the resulting bias are eliminated. Moreover, unlike [29], where NSGA-II is used to select the clustering parameters of FCM (which is essentially a single objective clustering that minimizes cluster variance), here, NSGA-II has directly been used in the clustering stage. This enables the algorithm to come out of the local optima, whereas FCM is known to often fall in local optima. Furthermore, the use of NSGA-II in
the clustering stage allows the method to suitably balance the different characteristics of clustering, unlike single objective techniques. Thus, by using NSGA-II directly for clustering, a Pareto optimal front of non-dominated solutions is generated, which allows us to use k-nn classification to find the most promising clustering solution from it.

The major purpose of this paper is to establish that the problem of fuzzy clustering of categorical data can be posed as one of multiobjective optimization of fuzzy compactness and separation, and this leads to improved performance. NSGA-II is a widely used multiobjective optimization technique which is applied in this regard. However, any other multiobjective optimization technique within the evolutionary computation framework, such as SPEA2 [33] or AMOSA [20], could have been used.

Experiments have been carried out for four synthetic and four real-life categorical datasets. Comparison has been made among different algorithms, such as fuzzy K-modes, K-modes, K-medoids, average linkage, CCDV, the single objective GA (SGA)-based clustering algorithms, and the proposed NSGA-II based multiobjective fuzzy clustering scheme. The superiority of the multiobjective algorithm has been demonstrated both quantitatively and visually. Also, statistical significance tests are conducted in order to confirm that the superior performance of the proposed technique is significant and does not occur by chance.

The rest of the paper is organized as follows: the next section describes the problem of fuzzy clustering for categorical data. Section III discusses the basic concepts of multiobjective optimization. In Section IV, the proposed multiobjective fuzzy clustering technique is described in detail. Section V describes some clustering algorithms used for the comparison purpose. The experimental results are provided in Section VI. In Section VII, results for statistical significance tests are reported. Finally, Section VIII concludes the paper.

II. FUZZY CLUSTERING OF CATEGORICAL DATA

This section describes the fuzzy K-modes clustering algorithm [7] for categorical datasets. The fuzzy K-modes algorithm is the extension of the well-known FCM [4] algorithm in categorical domain. Let \( X = \{x_1, x_2, \ldots, x_n\} \) be a set of \( n \) objects having categorical attribute domains. Each object \( x_i \), \( i = 1, 2, \ldots, n \) is described by a set of \( p \) attributes \( A_1, A_2, \ldots, A_p \). Let \( \text{DOM}(A_j) \), \( 1 \leq j \leq p \) denote the domain of the \( j \)th attribute, and it consists of different \( q_j \) categories such as \( \text{DOM}(A_j) = \{a_{j1}^j, a_{j2}^j, \ldots, a_{jq_j}^j\} \). Hence, the \( i \)th categorical object is defined as \( x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}] \), where \( x_{ij} \in \text{DOM}(A_j) \), \( 1 \leq j \leq p \).

The cluster centers in the FCM are replaced by cluster modes in the fuzzy K-modes clustering. A mode is defined as follows: Let \( C_i \) be a set of categorical objects belonging to cluster \( i \). Each object is described by attributes \( A_1, A_2, \ldots, A_p \). The mode of \( C_i \) is a vector \( m_i = [m_{i1}, m_{i2}, \ldots, m_{ip}] \), \( m_j \in \text{DOM}(A_j) \), \( 1 \leq j \leq p \) such that the following criterion is minimized:

\[
D(m_i, C_i) = \sum_{x \in C_i} D(m_i, x). \tag{1}
\]

Here, \( D(m_i, x) \) denotes the dissimilarity measure between \( m_i \) and \( x \). Note that \( m_i \) is not necessarily an element of set \( C_i \).

The fuzzy K-modes algorithm partitions the dataset \( X \) into \( K \) clusters by minimizing the following criterion:

\[
J_m(U, Z : X) = \sum_{k=1}^{n} \sum_{i=1}^{K} u_{ik}^m D(z_{ik}, x_k). \tag{2}
\]

For probabilistic fuzzy clustering, the following are the conditions that must hold while minimizing \( J_m \):

\[
0 \leq u_{ik} \leq 1, \quad 1 \leq i \leq K, \quad 1 \leq k \leq n \tag{3}
\]

\[
\sum_{i=1}^{K} u_{ik} = 1, \quad 1 \leq k \leq n \tag{4}
\]

and

\[
0 < \sum_{k=1}^{n} u_{ik} < n, \quad 1 \leq i \leq K \tag{5}
\]

where \( m \) is the fuzzy exponent. \( U = [u_{ik}] \) denotes the \( K \times n \) fuzzy partition matrix, and \( u_{ik} \) denotes the membership degree of the \( k \)th categorical object to the \( i \)th cluster. \( Z = \{z_1, z_2, \ldots, z_K\} \) represents the set of cluster centers (modes).

Fuzzy K-modes algorithm is based on an alternating optimizing strategy. This involves iteratively estimating the partition matrix followed by computation of new cluster centers (modes). It starts with random initial \( K \) modes, and then, every iteration, it finds the fuzzy membership of each data point to every cluster using the following equation [7]:

\[
u_{ik} = \frac{1}{\sum_{j=1}^{K} \left( \frac{D(z_{ij}, x_k)}{D(z_{ij}, x_k)} \right)^{\frac{1}{m-1}}}, \quad \text{for } 1 \leq i \leq K, \quad 1 \leq k \leq n. \tag{6}\]

Note that while computing \( u_{ik} \) using (6), if \( D(z_{ij}, x_k) \) is equal to zero for some \( j \), then \( u_{ik} \) is set to zero for all \( i = 1, \ldots, K \), \( i \neq j \), while \( u_{ik} \) is set equal to 1.

Based on the membership values, the cluster centers (modes) are recomputed as follows. If the membership values are fixed, then the locations of the modes that minimize the objective function in (2) will be [7] \( z_{ij} = [z_{i1}, z_{i2}, \ldots, z_{ip}] \), where \( z_{ij} = a_{j}^j \in \text{DOM}(A_j) \), and

\[
\sum_{k, x_{ij} = a_{j}^j} u_{ik}^n \geq \sum_{k, x_{ij} = a_{j}^t} u_{ik}^n, \quad 1 \leq t \leq q_j, \quad r \neq t. \tag{7}\]

The algorithm terminates when there is no noticeable improvement in \( J_m \) value (2). Finally, each object is assigned to the cluster to which it has the highest membership.

The main disadvantages of the fuzzy K-modes clustering algorithms are that 1) it depends heavily on the initial choice of the modes, and 2) it often gets trapped into some local optimum.

III. MULTIOBJECTIVE OPTIMIZATION USING GENETIC ALGORITHMS

In many real-world situations, there may be several objectives that must be optimized simultaneously in order to solve a certain problem. This is in contrast to the problems tackled by
conventional GAs, which involve optimization of just a single criterion. The main difficulty in considering multiobjective optimization is that a single optimum solution does not exist, and therefore, it is difficult to compare one solution with another. In general, these problems admit multiple solutions, each of which is considered acceptable and equivalent when the relative importance of the objectives is unknown. The best solution is subjective and depends on the need of the designer or the decision maker.

As evolutionary algorithms are population-based methods, it is straightforward to extend them to handle multiple objectives. On the contrary, it is difficult to extend the traditional search and optimization methods such as gradient descent search, and other non-conventional ones, such as simulated annealing, to the multiobjective case, since they deal with a single solution.

The multiobjective optimization can be formally stated as follows [18]. Find the vector \( \bar{x}^* \) that satisfies the inequality constraints
\[
g_i(\bar{x}) \geq 0, \quad i = 1, 2, \ldots, m
\]
and the \( p \) equality constraints
\[
h_i(\bar{x}) = 0, \quad i = 1, 2, \ldots, p
\]
and optimizes the vector function
\[
\vec{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T.
\]

The constraints given in (8) and (9) define the feasible region \( F \) which contains all the admissible solutions. Any solution outside this region is inadmissible since it violates one or more constraints. The vector \( \bar{x}^* \) denotes an optimal solution in \( F \). In the context of multiobjective optimization, the difficulty lies in the definition of optimality, since it is only rarely that we will find a situation where a single vector \( \bar{x}^* \) represents the optimum solution with respect to all the objective functions.

The concept of Pareto optimality comes handy in the domain of multiobjective optimization. A formal definition of Pareto optimality from the viewpoint of minimization problem may be given as follows. A decision vector \( \bar{x}^* \) is called Pareto optimal if and only if there is no \( \bar{x} \) that dominates \( \bar{x}^* \), i.e., there is no \( \bar{x} \) such that
\[
\forall i \in \{1, 2, \ldots, k\}, \ f_i(\bar{x}) \leq f_i(\bar{x}^*) \quad \text{and} \quad \exists i \in \{1, 2, \ldots, k\}, \ f_i(\bar{x}) < f_i(\bar{x}^*).
\]

In words, \( \bar{x}^* \) is Pareto optimal if there exists no feasible vector \( \bar{x} \) that causes a reduction of some criterion without a simultaneous increase in at least another. In general, Pareto optimum usually admits a set of solutions called non-dominated solutions.

There are different approaches to solving multiobjective optimization problems [13], [18], e.g., aggregating, population based non-Pareto, and Pareto-based techniques. In aggregating techniques, the different objectives are generally combined into one using weighting or goal-based method. Vector evaluated genetic algorithm (VEGA) is a technique in the population-based non-Pareto approach in which different subpopulations are used for the different objectives. Multiple objective GA (MOGA), non-dominated sorting GA (NSGA), and niched Pareto GA (NPGA) constitute a number of techniques under the Pareto-based non-elitist approaches [13]. NSGA-II [21], SPEA [34], and SPEA2 [33] are some recently developed multiobjective elitist techniques. The present paper uses NSGA-II as the underlying multiobjective algorithm for developing the proposed fuzzy clustering method.

IV. MULTIOBJECTIVE FUZZY CLUSTERING FOR CATEGORICAL ATTRIBUTES

In this section, the method of using NSGA-II for evolving a set of near-Pareto optimal non-degenerate fuzzy partition matrices is described.

A. Chromosome Representation

Each chromosome is a sequence of attribute values representing the \( K \) cluster modes. If each categorical object has \( p \) attributes \( \{A_1, A_2, \ldots, A_p\} \), the length of a chromosome will be \( K \times p \), where the first \( p \) positions (or genes) represent the \( p \)-dimensions of the first cluster mode, the next \( p \) positions represent that of the second cluster mode, and so on. As an illustration let us consider the following example. Let \( p = 3 \) and \( K = 3 \). Then, the chromosome
\[
c_{11} \ c_{12} \ c_{13} \ c_{21} \ c_{22} \ c_{23} \ c_{31} \ c_{32} \ c_{33}
\]
represents the three cluster modes \( \{c_{11}, c_{12}, c_{13}\}, \{c_{21}, c_{22}, c_{23}\}, \text{and} \{c_{31}, c_{32}, c_{33}\} \), where \( c_{ij} \) denotes the \( j \)th attribute value of the \( i \)th cluster mode. Also, \( c_{ij} \in \text{DOM}(A_j), \ 1 \leq i \leq K, 1 \leq j \leq p \).

B. Population Initialization

The initial \( K \) cluster modes encoded in each chromosome are chosen as \( K \) random objects of the categorical dataset. This process is repeated for each of the \( P \) chromosomes in the population, where \( P \) is the population size.

C. Computation of Objective Functions

In this paper, the global compactness \( \pi \) [35] of the clusters and the fuzzy separation \( \text{Sep} \) [35] have been considered as the two objectives that need to be optimized simultaneously. For computing the measures, the modes encoded in a chromosome are first extracted. Let these be denoted as \( z_1, z_2, \ldots, z_K \). The membership values \( u_{ik}, i = 1, 2, \ldots, K \) and \( k = 1, 2, \ldots, n \) are computed as follows [7]
\[
u_{ik} = \frac{1}{\sum_{j=1}^{K} \left( \frac{D(z_i, x_k)}{D(z_j, x_k)} \right)^{m-1}}, \quad \text{for} \ 1 \leq i \leq K, \ 1 \leq k \leq n
\]
where \( D(z_i, x_k) \) and \( D(z_j, x_k) \) are as described earlier. \( m \) is the weighting coefficient. [Note that while computing \( u_{ik} \) using (11), if \( D(z_j, x_k) \) is equal to zero for some \( j \), then \( u_{ik} \) is set to zero for all \( i = 1, \ldots, K, i \neq j \), while \( u_{ik} \) is set equal to 1.] Subsequently, each mode encoded in a
The chromosome is updated to \( z_i = [z_{i1}, z_{i2}, \ldots, z_{ip}] \), where \( z_{ij} = a_j^r \in \text{DOM}(A_j) \) \[7\], and
\[
\sum_{k, x_{ij} = a_j^r} u_{ik}^{m} \geq \sum_{k, x_{ij} = a_j^r} u_{rk}^{m}, \quad 1 \leq t \leq q_j, \quad r \neq t. \quad (12)
\]

This means that the category of the attribute \( A_j \) of the cluster centroid \( z_i \) is set to the category value that attains the maximum value of the summation of \( u_{ij} \) (the degrees of membership to the \( i \)th cluster) over all categories. Accordingly, the cluster membership values are recomputed as per \( (11) \).

The variation \( \sigma_i \) and fuzzy cardinality \( n_i \) of the \( i \)th cluster \( i = 1, 2, \ldots, K \) are calculated using the following equations \[35\]:
\[
\sigma_i = \sum_{k=1}^{n} u_{ik}^{m} D(z_i, x_k), \quad 1 \leq i \leq K \quad (13)
\]
and
\[
n_i = \sum_{k=1}^{n} u_{ik}, \quad 1 \leq i \leq K. \quad (14)
\]

The global compactness \( \pi \) of the solution represented by the chromosome is then computed as \[35\]
\[
\pi = \sum_{i=1}^{K} \frac{\sigma_i}{n_i} = \sum_{i=1}^{K} \frac{\sum_{k=1}^{n} u_{ik}^{m} D(z_i, x_k)}{\sum_{k=1}^{n} u_{ik}}. \quad (15)
\]

To compute the other fitness function fuzzy separation \( \text{Sep} \), the mode \( z_i \) of the \( i \)th cluster is assumed to be the center of a fuzzy set \( \{z_j | 1 \leq j \leq K, j \neq i \} \). Hence, the membership degree of each \( z_j \) to \( z_i \), \( j \neq i \) is computed as \[35\]
\[
\mu_{ij} = \frac{1}{\sum_{k=1,k \neq j}^{K} \left( \frac{D(z_j, z_i)}{D(z_j, z_i)} \right)^{\frac{1}{\sigma^2}}} \quad i \neq j. \quad (16)
\]

Subsequently, the fuzzy separation is defined as \[35\]
\[
\text{Sep} = \sum_{i=1}^{K} \sum_{j=1,j \neq i}^{K} \mu_{ij}^{m} D(z_i, z_j). \quad (17)
\]

Although several cluster validity indices exist, a careful study reveals that most of these consider the cluster compactness and separation in some form \[2\], \[36\]. Hence, in this paper, we have chosen to optimize the global cluster variance \( \pi \) (reflective of cluster compactness) and the fuzzy separation \( \text{Sep} \) (reflective of cluster separation). The purpose of this paper is to establish the effectiveness of the basic principle of multiobjective fuzzy clustering for categorical data. However, an exhaustive study involving two or more other powerful fuzzy cluster validity indices will constitute an area of interesting future work.

D. Selection, Crossover, and Mutation

The popularly used genetic operations are selection, crossover, and mutation. The selection operation used here is the crowded binary tournament selection used in NSGA-II. After selection, the selected chromosomes are put in the mating pool. Conventional single-point crossover depending on crossover probability \( \mu_c \) has been performed to generate the new offspring solutions from the chromosomes selected in the mating pool. For performing the mutation, a mutation probability \( \mu_m \) has been used. If a chromosome is selected to be mutated, the gene position that will undergo mutation is selected randomly. After that, the categorical value of that position is replaced by another random value chosen from the corresponding categorical domain. The most characteristic part of NSGA-II is its elitism operation, where the non-dominated solutions among the parent and child populations are propagated to the next generation. For details on the different genetic processes, see \[13\]. The near-Pareto-optimal strings of the last generation provide the different solutions to the clustering problem.

E. Selecting a Solution From the Non-dominated Set

As discussed earlier, the multiobjective GA-based categorical data clustering algorithm produces near-Pareto-optimal non-dominated set of solutions in the final generation. Hence, it is necessary to choose a particular solution from among the set of non-dominated solutions \( N \). This problem has been addressed in several recent research works \[37\]–\[41\], where search is focussed to identify the solutions situated at the “knee” regions of the non-dominated front. In \[25\], the authors proposed a post-processing approach, where the most complete Pareto front approximation set is obtained first, and then, it is reduced to a single solution. The method is motivated by GAP statistic \[42\] and makes use of several domain-specific considerations.

The technique adopted in this paper is used to search for the complete approximated Pareto front and apply a post-processing to identify the solution that shares most information provided by all the non-dominated solutions. In this approach, all the non-dominated solutions have been given equal importance, and the idea is to extract the combined clustering information. In this regard, a majority voting technique followed by \( k \)-nearest neighbor \((k\text{-nn})\) classification has been adopted in order to select a single solution from the set of the non-dominated solutions.
First, the clustering label vectors are computed from the unique non-dominated solutions produced by the proposed multiobjective technique. This is done by assigning each of the data points to the cluster to which it has the highest membership. Subsequently, a majority voting technique has been applied to the label vectors, and the points that are labeled with same class by at least 50% solutions are identified. Before applying the majority voting, we ensure the consistency among the label vectors of the different solutions, i.e., cluster $i$ of the first solution should match the cluster $i$ of all other solutions. This is done as follows.

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ be the label vector of the first non-dominated solution, where each $x_i \in \{1, 2, \ldots, K\}$ is the cluster label of the point $x_i$. At first, $\mathcal{X}$ is relabeled so that the first point is labeled 1 and the subsequent points are labeled accordingly. To relabel $\mathcal{X}$, first a vector $\mathcal{L}$ of length $K$ is formed that stores the unique class labels from $\mathcal{X}$ in the order of their first appearance in $\mathcal{X}$. The vector $\mathcal{L}$ is computed as follows:

$$k = 1, \mathcal{L}_k = l_1, \text{lab} = \{\mathcal{L}_1\}$$

for $i = 2, \ldots, n$

if $l_i \notin \text{lab}$

$k = k + 1.$

$\mathcal{L}_k = l_i.$

$\text{lab} = \text{lab} \cup \{l_i\}.$

end if

end for

Then a mapping $\mathcal{M}: \mathcal{L} \rightarrow \{1, \ldots, K\}$ is defined as follows:

$$\forall i = 1, \ldots, K, \mathcal{M}[\mathcal{L}_i] = i. \quad (18)$$

Next a temporary vector $\mathcal{T}$ of length $n$ is obtained by applying the above mapping on $\mathcal{X}$ as follows:

$$\forall i = 1, 2, \ldots, n, \mathcal{T}_i = \mathcal{M}[l_i]. \quad (19)$$

After that, $\mathcal{X}$ is replaced by $\mathcal{T}$. This way $\mathcal{X}$ is relabeled. For example, let initially $\mathcal{X} = \{33111442\}$. After relabeling, it would be $\{11222334\}$.

Now, the label vector of each of the other non-dominated solutions is modified by comparing it with the label vector of the first solution as follows. Let $\mathbb{N}$ be the set of non-dominated solutions (label vectors) produced by the proposed multiobjective clustering technique and $\mathcal{X}$ be the relabeled cluster label vector of the first solution. Suppose $\mathcal{Y} \in \mathbb{N} \setminus \mathcal{X}$ (i.e., $\mathcal{Y}$ is a label vector in $\mathbb{N}$ other than $\mathcal{X}$) is another label vector which is to be relabeled in accordance with $\mathcal{X}$. This is done as follows: First, for each unique class label $l$ in $\mathcal{X}$, all the points $\mathcal{P}_l$ that are assigned the class label $l$ in $\mathcal{X}$ are found. Thereafter, observing the class labels of these points from $\mathcal{Y}$, we obtain the class label $b$ from $\mathcal{Y}$, which is assigned for the maximum number of points in $\mathcal{P}_l$. Then, a mapping $\mathcal{M}_b$ is defined as $\mathcal{M}_b:b \rightarrow l$. This process is repeated for each unique class label $l \in \{1, \ldots, K\}$ in $\mathcal{X}$. After getting all the mappings $\mathcal{M}_b$ for all unique class labels $b \in \{1, \ldots, K\}$ in $\mathcal{Y}$, these are applied on $\mathcal{Y}$ to relabel $\mathcal{Y}$ in accordance with $\mathcal{X}$. All the non-dominated solutions $\mathcal{Y} \in \mathbb{N} \setminus \mathcal{X}$ are relabeled in accordance with $\mathcal{X}$ as discussed above.

Note that the mapping $\mathcal{M}_b$ should be one to one to ensure that after relabeling, $\mathcal{Y}$ contains all the $K$ class labels. This constraint may be violated while finding $b$, especially in cases of ties. This situation is handled as follows. If a one-to-one mapping cannot be obtained, we try to match all possible relabelings, i.e., $K!$ number of relabelings of $\mathcal{Y}$ and find the best match with $\mathcal{X}$. The best matched relabeling of $\mathcal{Y}$ is kept.

Consider the following example. Let $\mathcal{X} = \{11222334\}$ and two other label vectors be $\mathcal{Y} = \{22444113\}$ and $\mathcal{Z} = \{42333221\}$. If $\mathcal{Y}$ and $\mathcal{Z}$ are relabeled to make them consistent with $\mathcal{X}$, then relabeled $\mathcal{Y}$ becomes $\{11222334\}$, and relabeled $\mathcal{Z}$ becomes $\{13222334\}$.

After relabeling all the label vectors, majority voting is applied for each point. The points that are voted by at least 50% of the solutions to have a particular class label are now taken as the training set for the remaining points. The remaining points are assigned a class label according to $k$-nn classifier. That is, for each unassigned points, $k$-nearest neighbors are computed, and the point is assigned a class label that is obtained by the majority voting of the $k$-nearest neighbors. The value for $k$ is selected as 5.

Application of majority voting followed by $k$-nn classification produces a new cluster label vector $\%$ that shares the clustering information of all the non-dominated solutions. Thereafter, the percentage of matching with $\%$ is computed for each of the label vectors corresponding to each non-dominated solution. The label vector of the non-dominated solution that matches best with $\%$ is chosen from the set of the non-dominated solutions.

V. CONTESTANT METHODS

This section describes the contestant clustering algorithms that are used for the purpose of performance comparison.

A. K-medoids

Partitioning around medoids (PAM), which is also called the K-medoids clustering [3], is a variation of the K-means with the objective to minimize the within cluster variance $W(K)$

$$W(K) = \sum_{i=1}^{K} \sum_{x \in C_i} D(m_i, x). \quad (20)$$

Here, $m_i$ is the medoid of cluster $C_i$, and $D(\cdot)$ denotes a dissimilarity measure. A cluster medoid is defined as the most centrally located point within the cluster, i.e., it is the point from which the sum of distances to the other points of the cluster is minimum. Thus, a cluster medoid always belongs to the set of input data points $X$. The resulting clustering of the dataset $X$ is usually only a local minimum of $W(K)$. The idea of PAM is to select $K$ representative points, or medoids, in $X$ and to assign the rest of the data points to the cluster identified by the closest medoid. Initial medoids are chosen randomly. Then, all points in $X$ are assigned to the nearest medoid. In each iteration, a new medoid is determined for each cluster by finding the data point with minimum total dissimilarity to
all other points of the cluster. Subsequently, all the points in \( X \) are reassigned to their clusters in accordance with the new set of medoids. The algorithm iterates until \( W(K) \) no longer changes.

**B. K-modes**

K-modes clustering [6] is the crisp version of the fuzzy K-modes algorithm. The K-modes algorithm works similar to the K-medoids with the only difference that here, instead of medoids, modes are used to represent a cluster. The K-modes algorithm minimizes the following objective function:

\[
TC(K) = \sum_{i=1}^{K} \sum_{x \in C_i} D(m_i, x). \tag{21}
\]

Here, \( m_i \) denotes the mode of the cluster \( C_i \). The mode of a set of points \( P \) is a point (not necessarily belongs to \( P \)) whose \( j \)th attribute value is computed as the most frequent value of the \( j \)th attribute over all the points in \( P \). If there are more than one most frequent value, one of them is chosen arbitrarily. The iteration steps are similar to K-medoids and only differ in center (mode) updating process.

**C. Hierarchical Agglomerative Clustering**

Agglomerative clustering techniques [1] begin with singleton clusters and combine two least distant clusters at every iteration. Thus, in each iteration, two clusters are merged, and hence, the number of clusters is reduced by one. This proceeds iteratively in a hierarchy, providing a possible partitioning of the data at every level. When the target number of clusters \( K \) is achieved, the algorithms terminate. Single, average, and complete linkage agglomerative algorithms differ only in the linkage metric used, i.e., they differ in computing the distance between two clusters. For the single linkage algorithm, the distance between two clusters \( C_i \) and \( C_j \) is computed as the smallest distance between all possible pairs of data points \( x \) and \( y \), where \( x \in C_i \), and \( y \in C_j \). For the average and the complete linkage algorithms, the linkage metrics are taken as the average and largest distances, respectively.

**D. Clustering Categorical Data Based on Distance Vectors**

Clustering categorical data based on distance vectors (CCDV) [8] is a recently proposed clustering algorithm for categorical attributes. CCDV sequentially extracts the clusters from a given dataset based on the Hamming distance (HD) vectors, with automatic evolution of number of clusters. In each iteration, the algorithm identifies only one cluster, which is then deleted from the dataset at the next iteration. This procedure continues until there are no more significant clusters in the remaining data. For the identification and extraction of a cluster, the cluster center is first located by using a Pearson chi-squared-type statistic on the basis of HD vectors. The output of the algorithm does not depend on the order of the input data points.

**E. Single Objective GA-based Fuzzy Clustering Algorithms**

Three single objective GA (SGA) based clustering algorithms with different objective functions have been considered here. All the algorithms have the same chromosome representation as that of the multiobjective one and similar genetic operators. The first SGA-based algorithm minimizes the objective function \( \pi \), and thus, we call it SGA(\( \pi \)). The second SGA-based clustering maximizes \( \text{Sep} \), and hence, it is named SGA(\( \text{Sep} \)). The last one minimizes the objective function \( \pi/\text{Sep} \), and it is named SGA(\( \pi, \text{Sep} \)).

**F. Multiobjective GA-based Crisp Clustering Algorithm**

To establish the utility of incorporating fuzziness, a multiobjective crisp clustering algorithm (MOGA\(_{\text{crisp}}\)) for categorical data has been utilized. This algorithm uses similar encoding technique and similar genetic operators as the proposed multiobjective fuzzy clustering method. The only difference is that it optimizes the crisp versions of the objective functions described in (15) and (17), respectively. The objective functions for MOGA\(_{\text{crisp}}\) are as follows:

\[
\pi_{\text{crisp}} = \sum_{i=1}^{K} \sigma_i n_i = \sum_{i=1}^{K} \sum_{x_j \in C_i} D(z_i, x_j) \tag{22}
\]

\[
\text{Sep}_{\text{crisp}} = \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} D(z_i, z_j). \tag{23}
\]

Here \( R(z_i, x_k) \) is defined as follows:

\[
R(z_i, x_k) = \begin{cases} 
1, & \text{if } x_k \in C_i \\
0, & \text{otherwise}
\end{cases} \tag{24}
\]

Here \( C_i \) denotes the \( i \)th cluster and all other symbols have the same meaning as before. The data points are assigned to particular clusters as per nearest distance criterion. The final solution is selected from the generated non-dominated front following the procedure described in Section IV-E.

**VI. RESULTS AND DISCUSSION**

The performance of the proposed algorithm has been evaluated on four synthetic datasets (Cat250_15_5, Cat100_10_4, Cat500_20_10, and Cat280_10_6) and four real-life datasets (Congressional Votes, Zoo, Soybean, and Breast cancer).

**A. Dissimilarity Measures**

As stated earlier, there is no inherent distance/dissimilarity measure, such as Euclidean distance, that can be directly applied to compute the dissimilarity between two categorical objects. This is because there is no natural order among the categorical values of any particular attribute domain. Hence, it is difficult to measure the dissimilarity between two categorical objects.

In this paper following dissimilarity measure has been used for all the algorithms considered. Let \( x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}] \), and \( x_j = [x_{j1}, x_{j2}, \ldots, x_{jp}] \) be two categorical objects described by \( p \) categorical attributes. The dissimilarity measure between \( x_i \) and \( x_j, D(x_i, x_j) \), can be defined by the total
Dmis matches of the corresponding attribute categories of the two objects. Formally

\[ D(x_i, x_j) = \sum_{k=1}^{p} \delta(x_{ik}, x_{jk}) \]  

(25)

where

\[ \delta(x_{ik}, x_{jk}) = \begin{cases} 
0, & \text{if } x_{ik} = x_{jk} \\
1, & \text{if } x_{ik} \neq x_{jk}. 
\end{cases} \]  

(26)

Note that \( D(x_i, x_j) \) gives equal importance to all the categories of an attribute. However, in most of the categorical datasets, the distance between two data vectors depends on the nature of the datasets. Thus, if a dissimilarity matrix is predefined for a given dataset, the algorithms can adopt this to compute the dissimilarities.

B. Visualization

In this paper, for visualization of the datasets, the well-known visual assessment of (cluster) tendency (VAT) representation [43] is used. To visualize a clustering solution, first the points are reordered according to the class labels given by the solution. Thereafter, the distance matrix is computed on this reordered data matrix. In the graphical plot of the distance matrix, the boxes lying on the main diagonal represent the clustering structure.

Also, the plots of the Pareto frontier produced by the proposed algorithm have been used for visualization of the results.

C. Synthetic Datasets

Cat250_15_5: This synthetic dataset has a one-layer clustering structure [see Fig. 1(a)] with 15 attributes and 250 points. It has five clusters of the same size (50 points in each cluster). Each cluster has random categorical values selected from \( \{0, 1, 2, 3, 4, 5\} \) in a distinct continuous set of 12 attributes, while the rest attributes are set to 0.

Cat100_10_4: This is a synthetic dataset with 100 points and 10 attributes [see Fig. 1(b)]. The dataset has four clusters of same sizes (25 points each). For each cluster, two random attributes of the points of that cluster are zero valued and the remaining attributes have values in the range \( \{0, 1, 2, 3, 4, 5\} \).

Cat500_20_10: This synthetic dataset is generated by using the data generator available at http://www.datgen.com. This generator provides various options to specify such as the number of attributes, attribute domains and the number of tuples. The number of classes in the dataset is specified by the use of conjunctive rules of the form \((\text{Attr}_1 = a, \text{Attr}_2 = b, \ldots) \Rightarrow \text{classc}_1, \text{etc}\). The dataset contains 500 points and 20 attributes [see Fig. 1(c)]. The points are clustered into 10 clusters.

Cat280_10_6: This is another synthetic dataset obtained using the data generator. The dataset contains 280 points, 10 attributes, and six clusters [see Fig. 1(d)].

D. Real-Life Datasets

Congressional Votes: This dataset is the U.S. Congressional voting records in 1984 [see Fig. 2(a)]. Total number of
records is 435. Each row corresponds to one Congressman’s votes on 16 different issues (e.g., education spending, crime, etc.). All the attributes are boolean with Yes (i.e., 1) and No (i.e., 0) values. A classification label of Republican or Democrat is provided with each data record. The dataset contains records for 168 Republicans and 267 Democrats.

**Zoo:** The Zoo data consists of 101 instances of animals in a zoo with 17 features [see Fig. 2(b)]. The name of the animal constitutes the first attribute. This attribute is ignored. There are 15 boolean attributes corresponding to the presence of hair, feathers, eggs, milk, backbone, fins, tail and whether they are airborne, aquatic, predator, toothed, breathes, venomous, domestic, and catsize. The character attribute corresponds to the number of legs lying in the set \{0, 2, 4, 5, 6, 8\}. The dataset consists of seven different classes of animals.

**Soybean:** The Soybean dataset contains 47 data points on diseases in soybeans [see Fig. 2(c)]. Each data point has 35 categorical attributes and is classified as one of the four diseases, i.e., number of clusters in the dataset is four.

**Breast Cancer:** This dataset has total 699 records and nine attributes, each of which is described by 10 categorical values [see Fig. 2(d)]. The 16 rows which contain missing values are deleted from the dataset, and the remaining 683 records are used. The dataset is classified into two classes: benign and malignant.

The real-life datasets are obtained from the UCI Machine Learning Repository (www.ics.uci.edu/~mlearn/MLRepository.html).

### E. Performance Measure

The performance of the algorithms was measured using adjusted rand index (ARI) [44], [45]. Suppose \( T \) is the true clustering of a dataset and that \( C \) is a clustering result given by some clustering algorithm. Let \( a, b, c, \) and \( d \), respectively, denote the number of pairs of points belonging to the same cluster in both \( T \) and \( C \), the number of pairs belonging to the same cluster in \( T \) but to different clusters in \( C \), the number of pairs belonging to different clusters in \( T \) but to the same cluster in \( C \), and the number of pairs belonging to different clusters in both \( T \) and \( C \). ARI\((T, C)\) is then defined as follows:

\[
ARI(T, C) = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)}. \tag{27}
\]

The value of ARI\((T, C)\) lies between 0 and 1, and the higher value indicates that \( C \) is more similar to \( T \). Also, ARI\((T, T) = 1\).

### F. Comparison Procedure

The proposed multiobjective clustering technique and its crisp version search in parallel a number of solutions, and finally a single solution is chosen from the set of non-dominated solutions as discussed before. The single objective GA-based algorithms also search in parallel, and the best chromosome of the final generation is treated as the desired solution. On the contrary, the iterated algorithms, such as fuzzy K-modes, K-modes, K-medoids, and CCDV algorithms, try to
improve a single solution iteratively. They depend a lot on the initial configuration and often get stuck at the local optima. In order to compare these algorithms with GA-based methods, the following procedure is adopted. Iterated algorithms are run \( N \) times where each run consists of \( I \) re-iterations as follows:

\[
\begin{align*}
\text{for } i &= 1 \text{ to } N \\
\text{for } j &= 1 \text{ to } I \\
ARI[j] &= ARI \text{ score obtained by running the algorithm with new random seed.} \\
\text{end for} \\
ARI^I[i] &= \max\{ARI[1], \ldots, ARI[I]\}. \\
\text{end for} \\
\text{AvgARI} &= \text{avg}\{ARI^I[1], \ldots, ARI^I[N]\}.
\end{align*}
\]

In Tables I and III, we have reported the average \( ARI \) scores (\text{AvgARI}) for each algorithm.

The GA-based algorithms have been run \( N \) times, with number of generations as \( I \). The average of the best \( ARI \) scores for the GA-based algorithms are computed from the \( ARI \) scores of the \( N \) runs.

\section*{G. Input Parameters}

The GA-based algorithms are run for 100 generations with population size 50. The crossover and mutation probabilities are fixed at 0.8 and 0.1 (\textit{chromosome length}), respectively. These values are chosen after several experiments. The parameters \( N \) and \( I \) are taken as 50 and 100. Each re-iteration of the fuzzy K-modes, K-modes, and K-medoids algorithms have been executed 500 times, unless they converge earlier. This means that each of these three iterative algorithms has been executed for \( 50 \times 100 \) times, and each such execution is allowed for a maximum of 500 iterations. This is done for a fair comparison of these algorithms with the GA-based techniques which explore a total of \( 50 \times 100 \) combinations (since number of generations and population size of the GA-based techniques are 100 and 50, respectively). The fuzzy exponent \( m \) has been chosen to be 2.

\section*{H. Results for Synthetic Datasets}

Clustering results in terms of the average values of the \( ARI \) scores over 50 runs (\text{AvgARI}) on the four synthetic datasets using different algorithms are reported in Table I. The maximum values of \text{AvgARI} are shown in bold letters. From the table, it can be observed that the proposed multiobjective genetic clustering algorithm gives the best \text{AvgARI} scores for all the datasets. It is also evident from the table that for all the synthetic datasets, the fuzzy version of a clustering method performs better than its crisp counterpart. For example, the \text{AvgARI} scores for Fuzzy K-modes and K-modes algorithms are 0.5926 and 0.4998, respectively, for Cat280_10_6 data. This is also the case for multiobjective clustering. MOGA(\( \pi \)) and MOGA\(_{\text{crisp}}\) provide \text{AvgARI} scores 0.5851 and 0.5442, respectively, for this dataset. For all other datasets also the fuzzy algorithms provide better results than the corresponding crisp versions. This establishes the utility of incorporating fuzziness for clustering categorical datasets.

Table II reports another interesting observation. Here, the best \( ARI \) scores for single objective and multiobjective GA-based fuzzy algorithms have been shown for the Cat250_15_5 dataset. The final objective function values are also reported. As expected, SGA(\( \pi \)) produces the minimum \( \pi \) value (11.29), whereas SGA(\( \text{Sep} \)) gives the maximum \( \text{Sep} \) value (16.39). The proposed MOGA(\( \pi \), \( \text{Sep} \)) method provides a \( \pi \) value (11.34) greater than that provided by SGA(\( \pi \)), whereas a \( \text{Sep} \) value (15.38) smaller than that provided by SGA(\( \text{Sep} \)). However, in terms of the \( ARI \) scores, the proposed technique provides the best result (\( ARI = 1 \)). This signifies the importance of optimizing both \( \pi \) and \( \text{Sep} \) simultaneously instead of optimizing them separately, and this finding is very similar to that in [25].

Fig. 3 plots the Pareto fronts produced by one of the runs of the proposed multiobjective algorithm along with the best solutions provided by the other algorithms for the synthetic datasets. The figure also marks the selected solution from the non-dominated Pareto-optimal set. It appears that these selected solutions tend to fall at the knee regions of the Pareto fronts. Similar plots have been used for illustrations in [25] for showing the Pareto front generated by the multiobjective algorithm along with the solutions generated by other crisp clustering methods for continuous data. Here we have plotted the solutions for both fuzzy and crisp clustering methods used for clustering categorical data. As expected, each of the fuzzy K-modes, K-modes, K-medoids, and SGA(\( \pi \)) algorithms tends to minimize objective \( \pi \) and thus gives smaller values for \( \text{Sep} \) (larger values for \( 1/\text{Sep} \)). On the other hand, SGA(\( \text{Sep} \)) maximizes the objective \( \text{Sep} \) and, hence, gives larger values of the objective \( \pi \). The algorithms CCDV, SGA(\( \pi \), \( \text{Sep} \)), and MOGA\(_{\text{crisp}}\) are found to come nearest to the selected solution in the Pareto front.

<table>
<thead>
<tr>
<th>Result</th>
<th>Category</th>
<th>( \pi )</th>
<th>( \text{Sep} )</th>
<th>\text{Best ARIB Score}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat250</td>
<td>0.8077</td>
<td>0.5331</td>
<td>0.4243</td>
<td>0.4894</td>
</tr>
<tr>
<td>Cat100</td>
<td>0.7453</td>
<td>0.4855</td>
<td>0.2954</td>
<td>0.4537</td>
</tr>
<tr>
<td>Cat500</td>
<td>1.0000</td>
<td>0.5884</td>
<td>0.4276</td>
<td>0.5264</td>
</tr>
<tr>
<td>Cat280</td>
<td>1.0000</td>
<td>0.5983</td>
<td>0.4562</td>
<td>0.5442</td>
</tr>
</tbody>
</table>

Table II: Objective Function Values and the Best ARIB Scores for Cat250_15_5 Dataset
I. Results for Real-Life Datasets

Table III reports the AvgARIB scores over 50 runs of the different clustering algorithms on the real-life datasets. It is evident from the table that for all the datasets, the proposed multiobjective clustering technique produces the best AvgARIB scores. Here, it can also be noted that the fuzzy clustering procedures outperform the corresponding crisp versions for all the real-life datasets indicating the utility of incorporating fuzziness.

Fig. 4 shows the Pareto fronts produced by one of the runs of the proposed multiobjective technique along with the best solutions provided by other algorithms for the real-life datasets. Here, the selected solutions from the Pareto front are also mostly in the knee regions of the Pareto fronts. In addition, in the case of real datasets, the best competitive algorithms are CCDV, SGA(π, Sep), and MOGA\textsubscript{crisp}, which come closest to the selected non-dominated solution.

For both synthetic and real-life and categorical datasets, it has been found that the fuzzy clustering methods perform better than their crisp counterparts. This is due to the fact that since the fuzzy membership functions allow a data point to belong to multiple clusters simultaneously with different degrees of membership, the incorporation of fuzziness makes the algorithm capable of handling the overlapping partitions better. For this reason, the fuzzy algorithms outperform their corresponding crisp versions. Also note that both the fuzzy and crisp versions of the multiobjective categorical data clustering

Table III

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Votes</th>
<th>Zoo</th>
<th>Soybean</th>
<th>Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy K-modes</td>
<td>0.4983</td>
<td>0.6873</td>
<td>0.8412</td>
<td>0.7155</td>
</tr>
<tr>
<td>K-modes</td>
<td>0.4792</td>
<td>0.6789</td>
<td>0.5881</td>
<td>0.6115</td>
</tr>
<tr>
<td>K-medoids</td>
<td>0.4821</td>
<td>0.6224</td>
<td>0.8408</td>
<td>0.7124</td>
</tr>
<tr>
<td>Average linkage</td>
<td>0.5551</td>
<td>0.8927</td>
<td>1.0000</td>
<td>0.0127</td>
</tr>
<tr>
<td>CCDV</td>
<td>0.4964</td>
<td>0.8143</td>
<td>1.0000</td>
<td>0.7415</td>
</tr>
<tr>
<td>SGA(π)</td>
<td>0.4812</td>
<td>0.8032</td>
<td>0.8861</td>
<td>0.7268</td>
</tr>
<tr>
<td>SGA(Sep)</td>
<td>0.4986</td>
<td>0.8011</td>
<td>0.8877</td>
<td>0.2052</td>
</tr>
<tr>
<td>SGA(π, Sep)</td>
<td>0.5012</td>
<td>0.8348</td>
<td>0.9535</td>
<td>0.7032</td>
</tr>
<tr>
<td>MOGA\textsubscript{crisp}</td>
<td>0.5593</td>
<td>0.8954</td>
<td>1.0000</td>
<td>0.7621</td>
</tr>
<tr>
<td>MOGA(π, Sep)</td>
<td>0.5707</td>
<td>0.9175</td>
<td>1.0000</td>
<td>0.8016</td>
</tr>
</tbody>
</table>
methods use the similar encoding policy and final solution selection based on majority voting followed by k-nn classification. Hence, the better performance of MOGA(*π*, *Sep*) compared with MOGA−crisp indicates that improvement in clustering results is solely due to the introduction of fuzziness in the objective function, as well as in the clustering stage, and not due to the final solution selection strategy. This signifies the utility of incorporating fuzziness in the clustering techniques.

**J. Execution Time**

The execution time for the fuzzy clustering algorithms are usually more than that of corresponding crisp versions due to the computation of the fuzzy membership function and its updating. In this section, we have compared the time consumption among the fuzzy clustering algorithms. All the algorithms have been implemented in MATLAB and executed on an Intel Core 2 Duo 2.0-GHz Machine with a Windows XP operating system. On average, the proposed MOGA(*π*, *Sep*) clustering executes for 990.43 s for the Cat250_15_5 dataset, whereas fuzzy K-modes, SGA(*π*), SGA(*Sep*), and SGA(*π*, *Sep*) take 610.32, 680.73, 630.44, and 890.81 s, respectively, for this dataset. The execution times have been computed on the basis of the parameter setting discussed in Section VI-G. As expected, the execution time of the proposed multiobjective fuzzy clustering technique is larger than the other single objective fuzzy clustering methods because of some additional operations necessitated by its multiobjective nature. However, as is evident from the results, the clustering performance of MOGA(*π*, *Sep*) is the best among all the methods compared for all the datasets considered in this paper. It is also found during experimentation that even if the other algorithms used for comparison (both fuzzy and crisp) are allowed to run for the time taken by MOGA(*π*, *Sep*), they are not able to improve their clustering results any further.

The execution time of MOGA(*π*, *Sep*) for the other datasets are as follows: Cat100_10: 376.35 s, Cat500_20: 2045.58 s, Cat280_10: 1030.33 s, Zoo: 530.47 s, Soybean: 120.49 s, and Cancer: 1080.56 s. The timing requirements of the proposed technique can be reduced by using a stopping criterion based on some test of convergence for multiobjective evolutionary process.

**VII. STATISTICAL SIGNIFICANCE OF THE CLUSTERING RESULTS**

Tables I and III report the AvgARIB scores produced by different algorithms over 50 consecutive runs for the synthetic and real-life datasets, respectively. It is evident from the table that the AvgARIB scores produced by the proposed multiobjective clustering technique are better than that produced by

![Fig. 4. Pareto-optimal fronts produced by proposed technique for real-life datasets along with best results provided by other algorithms. (a) Votes dataset. (b) Zoo dataset. (c) Soybean dataset. (d) Cancer dataset.](image-url)

Authorized licensed use limited to: INDIAN INSTITUTE OF INFORMATION TECHNOLOGY. Downloaded on November 28, 2009 at 23:17 from IEEE Xplore. Restrictions apply.
all other algorithms, except for some of the datasets, where average linkage, CCDV, and SGA(\pi, Sep) provide scores similar to that of the proposed technique. To establish that this better performance of the proposed algorithm is statistically significant, some statistical significance test is required. In this paper, the statistical significance of the clustering solutions has been tested through t-test [46] at the 5% significance level. Ten groups, corresponding to the ten algorithms (1) Fuzzy K-modes, 2) K-modes, 3) K-medoids, 4) Average linkage, 5) CCDV, 6) SGA(\pi), 7) SGA(Sep), 8) SGA(\pi, Sep) 9) MOGA(\pi, Sep), and 10) MOGACrisp have been created for each dataset. Each group consists of the ARIB scores produced by 50 consecutive runs of the corresponding algorithm.

Table IV reports the P-values produced by t-test for comparison of two groups [group corresponding to MOGA(\pi, Sep) and a group corresponding to some other algorithm] at a time. As a null hypothesis, it is assumed that there are no significant differences between the AvgARIB scores of the two groups, whereas the alternative hypothesis is that there is a significant difference in the mean values of the two groups. All the P-values reported in the table are less than 0.05 (5% significance level). For example, the t-test between the algorithms MOGA(\pi, Sep) and fuzzy K-modes for Votes dataset provides a P-value of 4.33E-08, which is much less than the significance level 0.05. This is strong evidence against the null hypothesis, indicating that the better AvgARIB scores produced by the proposed method is statistically significant and has not occurred by chance. Similar results are obtained for all other datasets and for all other algorithms compared with MOGA(\pi, Sep), establishing the significant superiority of the proposed multiobjective fuzzy clustering algorithm.

### VIII. Conclusion

In this paper, a multiobjective genetic algorithm-based fuzzy clustering algorithm for clustering categorical datasets has been proposed. The proposed method optimizes two objectives, namely, the fuzzy compactness and the fuzzy separation of the clusters, simultaneously. The algorithm is designed using NSGA-II, which is a popular multiobjective GA. Also, a novel technique for selecting a particular solution from the non-dominated set of solutions produced by the proposed multiobjective technique has been proposed. The performance of the proposed technique, based on ARI, has been compared with that of the several other well-known categorical data clustering algorithms. Four synthetic and four real-life categorical datasets were used for performing the experiments. The superiority of the proposed multiobjective technique has been demonstrated, and use of multiple objectives rather than single objective has been justified. Moreover, statistical significance test based on t-statistic has been carried out in order to judge the statistical significance of the clustering solutions.

As a scope for future research, use of multiobjective algorithms other than NSGA-II, such as AMOSA [20], will be studied. Simultaneous optimization of other fuzzy validity indices in categorical domain, maybe more than two, can also be tried. Furthermore, use of data-specific dissimilarity measures needs a closer look. Also, while determining a single solution from the Pareto front, classification tools other than k-nn, such as support vector machines (SVMs), artificial neural networks (ANNs), etc. can be tried. Moreover, the use of variable string length GAs [16] to encode a variable number of clusters should be studied in order to automatically discover the number of clusters along with the clustering results.

### REFERENCES


<table>
<thead>
<tr>
<th>Datasets</th>
<th>Fuzzy K-modes</th>
<th>K-modes</th>
<th>K-medoids</th>
<th>Avg link</th>
<th>CCDV</th>
<th>SGA(\pi)</th>
<th>SGA(Sep)</th>
<th>SGA(\pi, Sep)</th>
<th>MOGACrisp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat250_15_5</td>
<td>2.13E-07</td>
<td>2.13E-10</td>
<td>7.44E-07</td>
<td>same</td>
<td>same</td>
<td>5.45E-10</td>
<td>1.67E-20</td>
<td>same</td>
<td>same</td>
</tr>
<tr>
<td>Cat100_10_4</td>
<td>3.02E-10</td>
<td>4.07E-17</td>
<td>5.39E-11</td>
<td>1.46E-12</td>
<td>6.28E-07</td>
<td>5.11E-15</td>
<td>3.82E-12</td>
<td>4.88E-07</td>
<td>3.98E-06</td>
</tr>
<tr>
<td>Cat500_20_10</td>
<td>2.73E-08</td>
<td>1.02E-10</td>
<td>7.88E-10</td>
<td>5.07E-13</td>
<td>1.22E-24</td>
<td>2.12E-11</td>
<td>1.43E-08</td>
<td>2.93E-11</td>
<td>6.92E-08</td>
</tr>
<tr>
<td>Cat280_10_6</td>
<td>4.06E-10</td>
<td>9.32E-12</td>
<td>4.95E-12</td>
<td>2.69E-31</td>
<td>3.44E-09</td>
<td>4.63E-12</td>
<td>3.66E-14</td>
<td>1.82E-09</td>
<td>5.21E-09</td>
</tr>
<tr>
<td>Votes</td>
<td>4.33E-08</td>
<td>3.85E-11</td>
<td>2.56E-09</td>
<td>7.23E-07</td>
<td>4.84E-08</td>
<td>2.33E-10</td>
<td>7.83E-08</td>
<td>4.72E-08</td>
<td>3.73E-08</td>
</tr>
<tr>
<td>Zoo</td>
<td>4.57E-19</td>
<td>6.45E-19</td>
<td>7.48E-20</td>
<td>5.11E-09</td>
<td>3.54E-07</td>
<td>4.12E-13</td>
<td>8.44E-14</td>
<td>2.18E-07</td>
<td>4.66E-09</td>
</tr>
<tr>
<td>Soybean</td>
<td>5.62E-06</td>
<td>2.04E-18</td>
<td>8.55E-06</td>
<td>same</td>
<td>same</td>
<td>5.66E-05</td>
<td>3.18E-05</td>
<td>5.08E-05</td>
<td>same</td>
</tr>
<tr>
<td>Cancer</td>
<td>1.83E-11</td>
<td>6.17E-12</td>
<td>2.48E-10</td>
<td>2.33E-40</td>
<td>7.55E-09</td>
<td>6.03E-08</td>
<td>5.22E-26</td>
<td>2.66E-08</td>
<td>8.56E-08</td>
</tr>
</tbody>
</table>


Ujjwal Maulik (M’99–SM’05) received the B.S. degrees in physics and computer science from University of Calcutta, Kolkata, India, in 1986 and 1989, respectively, and the M.S. and Ph.D. degrees in computer science from Jadavpur University, Kolkata, India, in 1991 and 1997, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India. He was the Head of the School of Computer Science and Technology with Kalyani Government Engineering College, Kalyani, India, during 1996–1999. He was with the Center for Adaptive Systems Application, Los Alamos, NM, in 1997; the University of New South Wales, Sydney, Australia, in 1999; the University of Texas at Arlington, in 2001; the University of Maryland Baltimore County in 2004; the Fraunhofer Institute AIS, St. Augustin, Germany, in 2005; Tsinghua University, Beijing, China, in 2007; and the University of Rome, Rome, Italy, in 2008. He has also visited many Institutes/Universities around the world to deliver invited lectures and conduct collaborative research. He is a co-author of two books and about 130 research publications. His research interests include artificial intelligence and combinatorial optimization, soft computing, pattern recognition, data mining, bioinformatics, VLSI, and distributed system.

Dr. Maulik was the recipient of the Government of India BOYSCAST fellowship in 2001. He has been the Program Chair, Tutorial Chair, and a Member of Program Committees of many international conferences and workshops. He is a Fellow of IETE, India.

Sanghamitra Bandyopadhyay (M’99–SM’05) received the B.S. degrees in physics and computer science from University of Calcutta, Kolkata, India, in 1988 and 1991, respectively, the M.S. degree in computer science from Indian Institute of Technology (IIT), Kharagpur, India, in 1993, and the Ph.D. degree in computer science from the Indian Statistical Institute, Kolkata, India, in 1998. Currently, she is an Associate Professor with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. She was with the Los Alamos National Laboratory, Los Alamos, NM, in 1997; the University of New South Wales, Sydney, Australia, in 1999; the University of Texas at Arlington in 2001; the University of Maryland at Baltimore in 2004; the Fraunhofer Institute, St. Augustin, Germany, in 2005; and Tsinghua University, Beijing, China, in 2006 and 2007. She has co-authored more than 150 technical articles in international journals, book chapters, and conference/workshop proceedings. She has delivered many invited talks and tutorials around the world. She has also edited special issues of journals in the areas of soft computing, data mining, and bioinformatics. Her research interests include computational biology and bioinformatics, soft and evolutionary computation, pattern recognition, and data mining.

Dr. Bandyopadhyay is the first recipient of the Dr. Shanker Dayal Sharma Gold Medal as well as the Institute Silver Medal for being adjudged the best all-round postgraduate performer at IIT; in 1994. She also received the Young Scientist Awards from the Indian National Science Academy and the Indian Science Congress Association in 2000. In 2002, she received the Young Scientist Award from the Indian National Academy of Engineers. She also received the 2006–2007 Swarnajayanti Fellowship in Engineering Sciences from the Government of India. She was the Program Co-Chair of the First International Conference on Pattern Recognition and Machine Intelligence held in Kolkata, India, during December 18–22, 2005. She authored a book entitled Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence (Springer) and two edited books entitled Advanced Methods for Knowledge Discovery from Complex Data, (Springer, 2005), and Analysis of Biological Data: A Soft Computing Approach (World Scientific, in 2007).