Abstract—Categorical data clustering has been gaining significant attention from researchers since the last few years, because most of the real life data sets are categorical in nature. In contrast to numerical domain, no natural ordering can be found among the elements of a categorical domain. Hence no inherent distance measure, like the Euclidean distance, would work to compute the distance between two categorical objects. Most of the clustering algorithms designed for categorical data are based on optimizing a single objective function. However, a single objective function is often not applicable for different kinds of categorical data sets. Motivated by this fact, in this article, the categorical data clustering problem has been modeled as a multiobjective optimization problem. A popular multiobjective genetic algorithm has been used in this regard to optimize two objectives simultaneously, thus generating a set of non-dominated solutions. The performance of the proposed algorithm has been compared with that of different well known categorical data clustering algorithms and demonstrated for a variety of synthetic and real life categorical data sets. Also a statistical significance test has been performed to establish the superiority of the proposed algorithm.

I. INTRODUCTION

Genetic algorithms [1], [2], [3] are randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. GAs perform search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. The algorithm starts by initializing a population of potential solutions encoded into strings called chromosomes. Each solution has some fitness value based on which the fittest parents that would be used for reproduction are found (survival of the fittest). The new generation is created by applying genetic operators like crossover (exchange of information among parents) and mutation (sudden small change in a parent) on selected parents. Thus the quality of population is improved as the number of generations increases. The process continues until some specific criterion is met or the solution converges to some optimized value.

Clustering [4], [5] is a useful unsupervised data mining technique which partitions the input space into K regions depending on some similarity/dissimilarity metric where the value of K may or may not be known a priori. K-means [4] is a traditional partitional clustering algorithm which starts with K random cluster centroids and the centroids are updated in successive iterations by computing the numerical averages of the feature vectors in each cluster. The objective of the K-means algorithm is to maximize the global compactness of the clusters. K-means clustering algorithm cannot be applied for clustering categorical data sets, where there is no natural ordering among the elements of an attribute domain. Thus no inherent distance measures, such as Euclidean distance, can be used to compute the distance between two feature vectors. Hence it is not feasible to compute the numerical average of a set of feature vectors. To handle such categorical data sets, a variation of K-means algorithm, namely K-medoids clustering has been proposed in [6]. In K-medoids algorithm, instead of computing the mean of feature vectors, a representative feature vector (cluster medoid) is selected for each cluster. A cluster medoid is defined as the most centrally located element in that cluster, i.e., it is the point from which the sum of the distances of the other points of the cluster is the minimum. K-medoids algorithm is also known as Partitioning Around Medoids (PAM) [6]. Another alternative of K-means clustering for categorical attributes is the K-modes clustering [7], [8], where instead of means, clusters are represented by modes (defined later).

A major disadvantage of K-means, K-medoids or K-modes clustering algorithms is that these algorithms often tend to converge to local optimum solutions. Moreover, they optimize a single objective function which may not work equally well for different kinds of data sets. Motivated by this fact, in this article, the categorical data clustering problem is modeled as one of multiobjective optimizations (MOO), where search is performed over a number of, often conflicting, objective functions. 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 further improved on any one objective without degrading another [9], [10], [11]. NSGA-II [12], a popular elitist multiobjective genetic algorithm, is used as the underlying optimization strategy. The two objective functions, the K-medoids error function [6] and the Silhouette cluster validity index [13], are optimized simultaneously. Integer valued strings representing the indices of the points are used to encode the cluster medoids.

A related work on multiobjective clustering around (MOCK-AM) has been proposed in [14] and [15], where the authors had used strings of length equal to the number of data points. In MOCK-AM, two objectives that were optimized are overall deviation (similar to K-medoids error function) and connectivity. Though MOCK-AM can be used to find non-convex shaped clusters, its performance degrades with more and more overlaps of clusters. Also this algorithm needs special initialization routines based on minimum spanning tree method. As it uses strings with length equal to

Anirban Mukhopadhyay and Ujjwal Maulik, Senior Member, IEEE
size of the data set, the algorithm needs more computational power.

The superiority of the proposed method over K-medoids, K-modes and single objective genetic clustering has been demonstrated on different synthetic and real life data sets. Also statistical significance tests have been carried out in order to confirm that the superior performance of the multiobjective clustering scheme is significant and has not occurred by chance.

The rest of the article is organized as follows: Section II describes some well known algorithms for categorical data clustering. In Section III, the distance metric used in this article to compute the distance between two categorical objects has been described. In section IV, a brief introduction to multiobjective optimization has been presented. The next section describes the proposed multiobjective genetic clustering method for categorical attributes. In section VI, a brief discussion on the choice of objective functions has been provided. Section VII presents the experimental results conducted on several synthetic and real life data sets. Finally, Section VIII concludes the article.

II. CATEGORICAL DATA CLUSTERING ALGORITHMS

This section describes some clustering algorithms used for categorical data.

A. K-medoids Clustering

Partitioning around medoids (PAM), also called K-medoids clustering [6], is a variation of K-means with the objective to minimize the within cluster variance $W(K)$.

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

Here $m_i$ is the medoid of cluster $C_i$ and $D(x, m_i)$ denotes the distance between the point $x$ and $m_i$. $K$ denotes the number of clusters. The resulting clustering of the data set $X$ is usually only a local minimum of $W(K)$. The idea of PAM is to select $K$ representative points, or medoids, in $X$ and assign the rest of the data points to the cluster identified by the nearest medoid. Initial set of $K$ medoids are selected randomly. Subsequently, all the 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 distance to all other points of the cluster. After that, all the points in $X$ are reassigned to their clusters in accordance with the new set of medoids. The algorithm iterates until $W(K)$ does not change any more.

B. K-modes Clustering

K-modes clustering [7], [16] algorithm works similarly to K-medoids with the only difference that here, instead of medoids, modes are used to represent the clusters. K-modes algorithm minimizes the following objective function.

$$TC(K) = \sum_{i=1}^{K} \sum_{x \in C_i} D(x, r_i)$$

Here $r_i$ denotes the mode of the cluster $C_i$. The mode of a set of points $P$ is defined as 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 $P$. The iteration steps are similar to K-medoids and only differ in the process of center (mode) updation.

III. DISTANCE METRIC

As discussed earlier, absence of any natural ordering among the elements of a categorical attribute domain prevents us to apply any inherent distance measure like Euclidean distance, to compute the distance between two categorical objects [17]. In this article following distance measure has been adopted for all the algorithms considered. Let $x_i = [x_{1i}, x_{2i}, \ldots, x_{pi}]$ and $x_j = [x_{1j}, x_{2j}, \ldots, x_{pj}]$ be two categorical objects described by $p$ categorical attributes.

The distance measure between $x_i$ and $x_j$, $D(x_i, x_j)$, can be defined by the total number of mismatches of the corresponding attribute categories of the two objects. Formally,

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

where

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

Note that $D(x_i, x_j)$ gives equal importance to all the categories of an attribute. However, in most of the categorical data sets, the distance between two data vectors depends on the nature of the data sets. Thus, if a distance matrix is precomputed for a given data set, the algorithms can adopt this for computing the distances.

IV. MULTIOBJECTIVE OPTIMIZATION

In many real world situations there may be several objectives that must be optimized simultaneously in order to solve a certain problem. The main difficulty in considering multiobjective optimization is that there is no accepted definition of optimum in this case, 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 often subjective and depends on the need of the designer or decision maker.

The multiobjective optimization can be formally stated as following [10]: Find the vector $\overline{x} = [x_1, x_2, \ldots, x_m]^T$ of decision variables which will satisfy the $m$ inequality constraints:

$$g_i(\overline{x}) \geq 0, \quad i = 1, 2, \ldots, m,$$

the $p$ equality constraints

$$h_i(\overline{x}) = 0, \quad i = 1, 2, \ldots, p,$$

and optimizes the vector function

$$\overline{f}(\overline{x}) = [f_1(\overline{x}), f_2(\overline{x}), \ldots, f_k(\overline{x})]^T.$$
The constraints given in Eqns. 5 and 6 define the feasible region \( \mathcal{F} \) which contains all the admissible solutions. Any solution outside this region is inadmissible since it violates one or more constraints. The vector \( \pi^* \) denotes an optimal solution in \( \mathcal{F} \). In the context of multiobjective optimization, the difficulty lies in the definition of optimality, since it is only rare that we will find a situation where a single vector \( \pi^* \) represents the optimum solution 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 \( \pi^* \) is called Pareto optimal if and only if there is no \( \pi \) that dominates \( \pi^* \), i.e., there is no \( \pi \) such that
\[
\forall i \in \{1, 2, \ldots, k\}, \ f_i(\pi) \leq f_i(\pi^*)
\]
and
\[
\exists i \in \{1, 2, \ldots, k\}, \ f_i(\pi) < f_i(\pi^*).
\]
In other words, \( \pi^* \) is Pareto optimal if there exists no feasible vector \( \pi \) which causes a reduction on some criterion without a simultaneous increase in at least another.

There are different approaches to solving multiobjective optimization problems [9], [10], 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), niched Pareto GA (NPGA) constitute a number of techniques under the Pareto based non-elitist approaches [9]. NSGA-II [12], SPEA [18] and SPEA2 [19] are some recent developed multiobjective elitist techniques. The present article uses NSGA-II as underlying multiobjective framework for developing the proposed clustering algorithm.

V. MULTIOBJECTIVE GENETIC CLUSTERING AROUND MEDOIDS

In this section, the proposed NSGA-II multiobjective GA based clustering around medoids has been described in detail.

A. Chromosome Encoding and Population Initialization

Different encoding techniques are available in literature [14], [15], [20]. A straightforward approach for encoding is to assign each data point to one of the clusters. A variation of the this technique, called locus based adjacency representation, is proposed in [14], [15], where each data point is connected with another one. The corresponding clustering solution is found by identifying connected components of the resulting directed graph. Advantage of these two encoding techniques is that they are able to capture non-convex shaped clusters. However, they have disadvantage that the chromosomes have length equal to the number of points in the data set, thus the genetic operations like crossover and mutation take longer time. Also using these encoding methods, it is difficult to separate overlapped clusters.

In [20], real-valued encoding of cluster centers has been used where each chromosome has length of only \( K \times d \). Here \( K \) and \( d \) are the number of clusters and the number of features, respectively. This representation can be used to identify overlapped clusters, and also takes less time while performing the genetic operations.

In this article, we have used a variation of the encoding method used in [20] to cope with categorical data. Here, the length of each chromosome is equal to the number of clusters \( K \). Each gene of the chromosome has an allele value chosen randomly from the set \( \{1, 2, \ldots, n\} \), where \( n \) is the number of points. Hence a string is represented as a vector of indices of the points in the data set. Each point index in a chromosome implies that the corresponding point is a cluster medoid. A chromosome is valid if no point index occurs more than once in the chromosome. The population is initialized by generating \( P \) such random strings, where \( P \) is the population size and it is fixed.

B. Fitness Computation

In this process, first the clusters are formed from a chromosome by taking the points encoded in it as the medoid points and assigning other points in the data set to their nearest medoids. After forming the clusters, new medoids for each cluster is found by selecting the most centrally located point of each cluster and the chromosome is updated with the indices of those medoids.

Two objective functions used in this article are the K-medoids error function [6] and a cluster validity index called Silhouette index [13]. The first objective function \( W(K) \) is given in Eqn. 1. The second objective function is computed as following: Suppose \( a \) represents the average distance of a point \( x_i \) from the other points of the cluster to which the point is assigned, and \( b \) represents the minimum of the average distances of the point from the points of the other clusters. Now the Silhouette width \( s(x_i) \) of the point is defined as:
\[
s(x_i) = \frac{b - a}{\max\{a, b\}}.
\]  
(8)

Silhouette index \( S(K) \) is the average silhouette width of all the data points and it reflects the compactness and separation of clusters.
\[
S(K) = \frac{1}{n} \sum_{i=1}^{n} s(x_i).
\]  
(9)

The value of Silhouette index varies from -1 to 1 and higher value indicates better clustering result.

The first objective function \( W(K) \) is to be minimized to get compact clusters, whereas the second objective function \( S(K) \) is to be maximized to get compact and well separated clusters. As the problem is modeled as minimization of objectives, we take the second objective as \( 1 - S(K) \). Thus the two fitness functions \( f_1 \) and \( f_2 \) are defined as follows:
\[
f_1 = \begin{cases} W(K) & \text{if the string is valid} \\ M & \text{otherwise}, \end{cases}
\]  
(10)
and
\[ f_2 = \begin{cases} 
1 - S(K) & \text{if the string is valid} \\
M & \text{otherwise,}
\end{cases} \tag{11} \]
where \( M \) is an arbitrary large number. Due to large value of \( M \), invalid strings automatically get out of the competition as the generations move.

C. Genetic Operators

The selection operation used here is the crowded binary tournament selection used in NSGA-II. Subsequently, conventional single point crossover depending on crossover probability \( \mu_c \) has been performed for generating the new offspring solutions from the chromosomes selected in the mating pool. The mutation operation has been defined as following: From the string to be mutated, a random element is chosen and it is replaced by a different index of point in the range \( \{1, \ldots, n\} \), such that no element is duplicated in the string. The operation is done with mutation probability \( \mu_m \). 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, the reader may refer to [12]. The algorithm has been run for a fixed number of generations. The near-Pareto-optimal strings of the last generation provide the different solutions to the clustering problem.

VI. CHOICE OF OBJECTIVES

In multiobjective clustering, choice of objective functions plays an important role in order to evolve proper clustering solutions. The chosen objectives should be such that they should be contradictory in nature. In this article, the two cluster validity measures, viz., the \( K \)-medoids error function (\( W(K) \)) and the silhouette index (\( S(K) \)) are used to design the two objectives. The two fitness functions \( f_1 \) (Eqn. 10) and \( f_2 \) (Eqn. 11) are designed in such a way that minimization of \( f_1 \) and \( f_2 \) corresponds to minimization of \( W(K) \) and maximization of \( S(K) \), respectively. Note that \( W(K) \) and \( S(K) \) represent two different aspects of clustering. Minimization of \( W(K) \) ensures identification of compact clusters, whereas, maximization of \( S(K) \) means identification of well separated and compact clusters. Hence, the two objective functions \( f_1 \) and \( f_2 \) are contradictory in nature and both of them have to be minimized to obtain compact and well separated clusters.

VII. EXPERIMENTAL RESULTS

The performance of the proposed algorithm has been evaluated on two synthetic data sets (Cat100_10.4 and Cat250_15.5) and three real life data sets (Congressional Votes, Zoo and Soybean). The proposed multiobjective clustering scheme has been compared with different algorithms, viz., \( K \)-medoids, \( K \)-modes, a single objective GA based clustering that minimizes \( f_1 \) only, (SGA(\( f_1 \))), a single objective GA based clustering that minimizes \( f_2 \) only, (SGA(\( f_2 \))) and a single objective GA based clustering that minimizes \( f_1 \times f_2 \) (SGA(\( f_1, f_2 \))). Each algorithm has been run for 20 times. As the multiobjective technique results in a set of non-dominated solutions, the solution that provides best Minkowski score (described later) has been considered in each run.

A. Synthetic Data Sets

1) Cat100_10.4: This is a synthetic data set with 100 points and 10 attributes. The data set has 4 clusters of same sizes (25 points each). For each cluster, 2 random attributes of the points of that cluster are zero valued and the remaining attributes have values in the range \( \{1, 2, 3, 4, 5\} \). The data set has been shown in Fig. 1(a). The last column of the figure gives the class labels where different colors denote different classes.

2) Cat220_10.5: This synthetic data set has 10 attributes and 220 points. It has 5 clusters and clusters 1 through 5 contain 38, 58, 51, 32, 41 points, respectively. Each cluster has random categorical values selected from \( \{1, 2, \ldots, 10\} \) in a distinct set of a random number of attributes, while the rest of the attributes are set to 0. The data set has been shown in Fig. 1(b).

B. Real Life Data Sets

1) Congressional Votes: This data set is the United States Congressional voting records in 1984. Total number of records is 435. Each row corresponds to one Congress man’s votes on 16 different issues (e.g., education spending, crime etc.). All attributes are boolean with the vote on each issue. The classification label of Republican or Democrat is provided with each data record. The data set contains records for 168 Republicans and 267 Democrats.

2) Zoo: The Zoo data consists of 101 instances of animals in a zoo with 17 features. The name of the animal constitutes the first attribute. This attribute is neglected. There are 15 boolean attributes corresponding to the presence of hair, feathers, eggs, milk, backbone, fins, tail; and whether 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 data set consists of 7 different classes of animals.

3) Soybean: The Soybean data set contains 47 data points on diseases in soybeans. Each data point has 35 categorical attributes and is classified as one of the four diseases, i.e., number of clusters in the data set is 4.

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

C. Input Parameters

The GA-based clustering techniques (both for single objective and multiobjective) are executed for 100 generations with a fixed population size = 50. The crossover and mutation probabilities, chosen based on several experiments, are 0.8 and 0.01, respectively. The number of iterations for \( K \)-medoids and \( K \)-modes algorithms are taken as 100 unless they converge before that.
Fig. 1. True clustering of synthetic data sets: (a) Cat100 10 4 data set, (b) Cat220 10 5 data set

D. Performance Metric

Here, the performances of the clustering algorithms are evaluated in terms of the Minkowski score. A clustering solution for a set of \( n \) elements can be represented by an \( n \times n \) matrix \( C \), where \( C_{i,j} = 1 \) if point \( i \) and \( j \) are in the same cluster according to the solution, and \( C_{i,j} = 0 \) otherwise. The Minkowski score (MS) of a clustering result \( C \) with reference to \( T \), the matrix corresponding to the true clustering, is defined as

\[
MS(T, C) = \frac{\| T - C \|}{\| T \|} \quad (12)
\]

where \( \| T \| = \sqrt{\sum_i \sum_j T_{i,j}} \).

The Minkowski score is the normalized distance between the two matrices. Lower Minkowski score implies better clustering solution, and a perfect solution will have a score zero.

E. Results

The results in terms of Minkowski scores produced by several algorithms on the synthetic and real life data sets are shown graphically as boxplots in Fig. 2 and Fig. 3, respectively. From the figures it is evident that for all the data sets, the solutions generated by the multiobjective method are better than that produced by the other algorithms. The median values of Minkowski scores given by the proposed technique is superior than that for all other algorithms including the single objective GA based methods.

![Boxplots of Minkowski scores for different algorithms on synthetic data sets: (a) Cat100 10 4 data set, (b) Cat220 10 5 data set](image)

![Boxplots of Minkowski scores for different algorithms on synthetic data sets: (a) Cat100 10 4 data set, (b) Cat220 10 5 data set](image)

The Tables I and II report the average values for Minkowski scores obtained by different algorithms over 20 runs on synthetic and real life data sets, respectively.
The multiobjective genetic clustering method is found to consistently outperform the K-medoids, K-modes and its single objective counterparts (i.e., SGA($f_1$), SGA($f_2$) and SGA($f_1, f_2$)). These results indicate the fact that global optimum of objective $f_1$ (minimized by SGA($f_1$) and K-medoids) or that of objective $f_2$ (minimized by SGA($f_2$)) do not correspond to the global optimum of Minkowski score, which is achieved by the proposed multiobjective algorithm.

### TABLE I

**AVERAGE MINKOWSKI SCORES FOR SYNTHETIC DATA SETS**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cat100</th>
<th>Cat220</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-medoids</td>
<td>0.9223</td>
<td>0.8018</td>
</tr>
<tr>
<td>K-modes</td>
<td>1.0677</td>
<td>1.0840</td>
</tr>
<tr>
<td>SGA($f_1$)</td>
<td>0.8899</td>
<td>0.7961</td>
</tr>
<tr>
<td>SGA($f_2$)</td>
<td>0.7058</td>
<td>0.7401</td>
</tr>
<tr>
<td>SGA($f_1,f_2$)</td>
<td>0.6924</td>
<td>0.6525</td>
</tr>
<tr>
<td>Multiobjective</td>
<td>0.6435</td>
<td>0.5788</td>
</tr>
</tbody>
</table>

### TABLE II

**AVERAGE MINKOWSKI SCORES FOR REAL LIFE DATA SETS**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Votes</th>
<th>Zoo</th>
<th>Soybean</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-medoids</td>
<td>0.6815</td>
<td>0.6601</td>
<td>0.4155</td>
</tr>
<tr>
<td>K-modes</td>
<td>0.6686</td>
<td>0.8515</td>
<td>0.7928</td>
</tr>
<tr>
<td>SGA($f_1$)</td>
<td>0.6787</td>
<td>0.6411</td>
<td>0.4462</td>
</tr>
<tr>
<td>SGA($f_2$)</td>
<td>0.6864</td>
<td>0.5120</td>
<td>0.7563</td>
</tr>
<tr>
<td>SGA($f_1,f_2$)</td>
<td>0.6550</td>
<td>0.5096</td>
<td>0.4147</td>
</tr>
<tr>
<td>Multiobjective</td>
<td>0.5344</td>
<td>0.3807</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

In this article, a statistical significance test called one way ANOVA (ANalysis Of VAriance) [21] has been conducted at the 5% significance level, to compare the average Minkowski scores produced by different algorithms. Six groups, corresponding to six algorithms (K-medoids, K-modes, SGA($f_1$), SGA($f_2$), SGA($f_1,f_2$) and MOGA), have been created for each data set. Each group consists of Minkowski scores produced by 20 consecutive runs of the corresponding algorithm.

As a null hypothesis ($H_0$), it is assumed that there are no significant differences among the average Minkowski scores produced by all the algorithms.

$$H_0 : \mu_1 = \mu_2 = \ldots = \mu_5.$$  \hspace{1cm} (13)

The alternative hypothesis ($H_1$) is that there are significant differences in average Minkowski scores for at least two methods.

$$H_1 : \exists i,j : i \neq j \Rightarrow \mu_i \neq \mu_j,$$  \hspace{1cm} (14)

where $\mu_i$ denotes the average Minkowski score of the $i^{th}$ group.

Tables III to VII report the ANOVA results for the five data sets considered here, respectively. The tables give diagnostics concerning between groups variance and within groups variance. The second data column of each table provides the degrees of freedom (df) connected with both the between groups variance (number of groups - 1) and the within groups variance (number of observations - number of groups). The first and third data columns present the sum of squares (SS)
and the mean square (MS) of variances of both between groups and within groups. The statistic to check the null hypothesis (F statistic) is the ratio of the mean squares between groups to the mean squares within groups. The high value for the F statistic allows us to reject the null hypothesis.

For example, ANOVA result for Congressional Votes data set (Table V) gives the value of the F statistic of 184.96, which is greater than its critical value (F-critical = 2.29) and probability (P-value) that this result occurred by chance is very less (5.63E-53). This is a strong evidence against the null hypothesis, indicating that the better average Minkowski score produced by the multiobjective genetic clustering technique has not occurred by chance. Similar results are noticed for all other data sets.

### TABLE III

**ANOVA RESULT FOR CAT100,10,4 DATA SET**

<table>
<thead>
<tr>
<th>Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
<th>F crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betwn Groups</td>
<td>2.74</td>
<td>5</td>
<td>0.549</td>
<td>242.63</td>
<td>5.08E-59</td>
<td>2.29</td>
</tr>
<tr>
<td>Within Groups</td>
<td>0.26</td>
<td>114</td>
<td>0.002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.00</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE IV**

**ANOVA RESULT FOR CAT220,10,5 DATA SET**

<table>
<thead>
<tr>
<th>Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
<th>F crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betwn Groups</td>
<td>3.03</td>
<td>5</td>
<td>0.605</td>
<td>273.60</td>
<td>9.51E-62</td>
<td>2.29</td>
</tr>
<tr>
<td>Within Groups</td>
<td>0.25</td>
<td>114</td>
<td>0.002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.28</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE V**

**ANOVA RESULT FOR CONGRESSIONAL VOTES DATA SET**

<table>
<thead>
<tr>
<th>Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
<th>F crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betwn Groups</td>
<td>0.34</td>
<td>5</td>
<td>0.068</td>
<td>184.96</td>
<td>5.63E-53</td>
<td>2.29</td>
</tr>
<tr>
<td>Within Groups</td>
<td>0.04</td>
<td>114</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>0.38</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE VI**

**ANOVA RESULT FOR ZOO DATA SET**

<table>
<thead>
<tr>
<th>Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
<th>F crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betwn Groups</td>
<td>2.04</td>
<td>5</td>
<td>0.408</td>
<td>471.06</td>
<td>2.29E-74</td>
<td>2.29</td>
</tr>
<tr>
<td>Within Groups</td>
<td>0.13</td>
<td>114</td>
<td>0.001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>2.17</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE VII**

**ANOVA RESULT FOR SOYBEAN DATA SET**

<table>
<thead>
<tr>
<th>Variation</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>P-value</th>
<th>F crit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Betwn Groups</td>
<td>8.28</td>
<td>5</td>
<td>1.655</td>
<td>447.22</td>
<td>3.84E-73</td>
<td>2.29</td>
</tr>
<tr>
<td>Within Groups</td>
<td>0.42</td>
<td>114</td>
<td>0.004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>8.70</td>
<td>119</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**F. Execution Time**

The proposed algorithm has been implemented using Matlab and executed on Pentium-IV 3 GHz machine with 256 MB RAM having Windows XP operating system. In Table VIII, the average execution times (in seconds) of the proposed multiobjective clustering algorithm over 20 runs have been reported for different data sets. The parameter setting for the proposed method is taken as discussed in Section VII-C. It appears from the table that the execution time increases with the increase in the number of points and the number of features in the data sets.

### VIII. CONCLUSIONS

Most of the clustering algorithms designed for categorical data optimize a single objective function, which may not be equally applicable for different kinds of data sets. Motivated by this, in this article, a multiobjective GA based clustering algorithm for clustering categorical data around medoids, has been proposed. The indices of the data points are encoded as the cluster medoids for this purpose. Two objective functions, namely K-medoids error function and Silhouette validity index have been simultaneously optimized using the multiobjective clustering method. The algorithm is designed on the framework of a popular multiobjective GA, NSGA-II. The performance of the proposed method has been compared with that of its single objective versions, K-medoids and K-modes algorithms. The superiority of the proposed scheme has been demonstrated on a number of synthetic and real life data sets. Also statistical significance tests based on ANOVA have been conducted to judge the statistical significance of the clustering solutions produced by different algorithms. As a future work, simultaneous optimization of other objective functions, may be more than two, is to be studied. Also, the use of multiobjective GAs other than NSGA-II, deserves some attention.

### REFERENCES


"Finding Groups in Data: An Introduction to Cluster Analysis.


