Simultaneous Continuous Feature Selection and K Clustering by Multi Objective Genetic Algorithm

Dipankar Dutta
Department of
Computer Science and
Information Technology,
University Institute of Technology,
The University of Burdwan,
Golapbug (North), Burdwan,
West Bengal, India
PIN-713104
Email:dipankar_dutta@rediffmail.com

Paramartha Dutta
Department of Computer
and System Sciences,
Visva-Bharati University,
Santiniketan, West Bengal, India
PIN-731235
Email:paramartha.dutta@gmail.com

Jaya Sil
Department of
Computer Science and Technology,
Bengal Engineering
and Science University,
Shibpur, West Bengal, India
PIN-711103
Email:js@cs.becs.ac.in

***Abstract*—We can classify clustering into two categories. In K Clustering, we know the number of clusters or K. In other category of clustering, K in unknown. In this paper we have considered the first category only. We can broadly classify features within a data set into continuous and categorical. Here we have considered data set with continuous features only. Clustering can be done by all features or by relevant features only. Researches had commonly used some feature selection techniques to select relevant features for clustering and then did clustering by some clustering algorithm. Here we have used Multi Objective Genetic Algorithm (MOGA) for simultaneous feature selection and clustering. Here, K-means is hybridized with GA. We have used hybridized GA to combine global searching abilities of GA with local searching abilities of K-means. Considering context sensitivity, we have used a special crossover operator called "pairwise crossover" and "substitution". Elimination of redundant, irrelevant features increases clustering performance, reflected in MOGA_Feature_Selection (H, S) compared with MOGA (H, S). The main contribution of this paper is simultaneous dimensionality reduction and optimization of objectives using MOGA.**

I. Introduction

Excluding multimedia data set, we can classify features of data set as continuous and categorical. Continuous feature values are naturally ordered and have implicit distance and magnitude semantics. Categorical feature can only take certain values. Ages, height and weight of people are continuous whereas gender is categorical. Encoding scheme used in the work is not suitable for categorical data. So we have considered continuous features only for this work. Clustering is to identify natural groups (clusters) within a data set such that instances in the same group are more similar than instances in different groups. It is one of the important tasks of data mining i.e. extracting information from data. From the definition of clustering two objectives of clustering are clear - minimization of intra-cluster distances (Homogeneity, H) and maximization of inter-cluster distances (Separation, S). Data clustering has therefore been considered as an intrinsically multi-objective optimization problem (MOOP) [5], [18], [19] where more than one objectives are to be optimized simultaneously. It is achieved by measuring H and S using Euclidean distance per feature metric, suitable for continuous features. Clustering are usually classified as NP-hard [13] problem. Evolutionary algorithms are meta-heuristics widely believed to be able to provide satisfactory sub-optimal solutions to NP-hard problems in reasonable time [16], [29], [34]. Among these approaches, genetic algorithm (GA) has been known to offer significant advantages against conventional evolutionary algorithms by using simultaneously several search principles and heuristics. GAs are "search algorithms based on the dynamics of natural selection and natural genetics" [15]. Categories of GAs are - simple GA (SGA) and multi objective GA (MOGA). When an optimization problem involves only one objective, the task of finding the best solution is a single objective optimization problem. In clustering, it is necessary to optimize more than one objective simultaneously by MOGA. As the relative importance of different clustering criteria are unknown, it is better to optimize (Homogeneity, H) and (Separation, S) separately rather than combining them into a single measure to be optimized. Although GAs have been used in data clustering problems [7], [16], [29], [31], [34], [40], most of them optimized single objective, which is hardly equally applicable to all kinds of data sets. All features are not equally important from clustering viewpoint. Although some researchers have done clustering by MOGA or by other multi objective evolutionary algorithms [1], [9], [10], [11], [20], [24], [25], [33], [36], [37], [38] none of them indeed tried simultaneous feature selection. Recently we have done simultaneous feature selection and clustering by MOGA for categorical features [12] only. MOGA_Feature_Selection (H, S) automatically distinguish between relevant and irrelevant features for clustering without taking help of feature selection techniques described in [23], [26], [39], [41], which is the main contribution of this work.

Organization of this paper is as follows. Section II defines the problem. Section III describes various preprocessing steps.

Section IV describes proposed approach for solving the problem. In section V testing and comparison of two clustering algorithms are carried out on 5 popular benchmark data sets. Finally section VI summarizes the work with concluding remarks.

II. Definitions

A relation R denotes a relational schema, $R(A_1, A_2, ...A_n)$, where $A_1, A_2, ...A_n$, a set of features (attributes) and n is the total number of features (attributes) including class label.

$dom(A_i)$ is the domain of A_i, where $1 \leq i \leq n$.

X represents a data set, which is a set of tuples that is $X = \{t_1, t_2, ...t_m\}$, where m = total number of tuples or records.

Each tuple t is an n-dimensional feature vector, which is an ordered list of n values that is $t = [v_1^t, v_2^t, ...v_n^t]$, where $v_i^t \in dom(A_i)$, with $1 \leq i \leq n$. v_i^t is i^{th} value in tuple t, which corresponds to the feature A_i. Each tuple, t belongs to predefined class represented by v_n^t where $v_n^t \in dom(A_n)$. v_n^t or class labels are unknown in clustering, so $t = [v_1^t, v_2^t, ...v_{n-1}^t]$. Henceforth $n-1$ = total number of features excluding class label. After normalization (explained in section III-B) $t = [nv_1^t, nv_2^t, ..., nv_{n-1}^t]$, where $0 \leq nv_i^t \leq 1$.

The problem is to cluster every t_i of X ($1 \leq i \leq m$) in K number of non-overlapping groups $\{C_1, C_2, ...C_K\}$, where $C_1 \cup C_2 \cup ...C_K = X$, $C_i \neq \emptyset$ and $C_i \cap C_j = \emptyset$ for $i \neq j$ and $j \leq K$.

A solution is a set of cluster centers (CCs) that is $\{CC_1, CC_2, ...CC_K\}$. Each CC_i is $(n-1)$-dimensional feature vector, that is $CC_i = [c_1^i, c_2^i, ...c_{n-1}^i]$.

Equations 1, 2 and 3 calculate Euclidean distances per feature between one cluster center and one tuple, two cluster centers and two tuples respectively.

$$d(CC_i, t_j) = [[\sum_{l=1}^{n-1}(c_l^i - nv_l^j)^2]/(n-1)]^{1/2} \quad (1)$$

$$d(CC_i, CC_j) = [[\sum_{l=1}^{n-1}(c_l^i - c_l^j)^2]/(n-1)]^{1/2} \quad (2)$$

$$d(t_i, i_j) = [[\sum_{l=1}^{n-1}(nv_l^i - nv_l^j)^2]/(n-1)]^{1/2} \quad (3)$$

III. Preprocessing

Before applying any clustering algorithm data need to be preprocessed. Several preprocessing steps [17], [35] as described below.

A. Making class label as the last feature

Original data sets are changed to label the last feature as the class label. After this step class label becomes n^{th} feature. In clustering class labels are unknown, so we are considering first $(n-1)$ features of data sets.

B. Normalization of data

A_i^{min} (minimum value of i^{th} attribute) and A_i^{max} (maximum value of i^{th} attribute) are calculated, where $1 \leq i \leq n-1$. As defined earlier $t = [v_1^t, v_2^t, ...v_{n-1}^t]$, with $v_i^t \in dom(A_i)$, where $1 \leq i \leq (n-1)$. After normalization $t = [nv_1^t, nv_2^t, ..., nv_{n-1}^t]$ (where $0 \leq nv_i^t \leq 1$) using equation 4. It is a simple linear interpolation based conversion formula.

$$nv_i^t = (v_i^t - A_i^{min})/(A_i^{max} - A_i^{min}) \text{ for } i = 1, 2, ...(n-1). \quad (4)$$

IV. Proposed Approach

In this work, MOGA is doing clustering and feature selection simultaneously, so we denote this process as MOGA_Feature_Selection (H, S). In one of our recent work [9] we have done similar work i.e. clustering by MOGA but we have not addressed the problem of feature selection there. So we denote that work as MOGA (H, S).

In MOGA (H, S), all features except class labels are taking part to build chromosomes. But all features are not equally important. So MOGA_Feature_Selection (H, S) select only important/ relevant features for clustering. This method provides three-way benefits.

i) Clustering performance improves significantly.

ii) In can easily identify important features.

iii) Due to the reduction of dimensionality in MOGA_Feature_Selection (H, S), faster convergence than MOGA (H, S) is achieved.

As some features are forming chromosomes, equation 1, 2 and 3 can be modified as 5, 6 and 7 respectively.

$$d_{mod}(CC_i, t_j) = [[\sum_{l=1}^{n-1}(c_l^i - nv_l^j)^2]/nva]^{1/2} \quad (5)$$

$$d_{mod}(CC_i, CC_j) = [[\sum_{l=1}^{n-1}(c_l^i - c_l^j)^2]/nva]^{1/2} \quad (6)$$

$$d_{mod}(t_i, i_j) = [[\sum_{l=1}^{n-1}(nv_l^i - nv_l^j)^2]/nva]^{1/2} \quad (7)$$

where nva = number of valid features in corresponding chromosome that is where $c_l^i \neq \#$. If $c_l^i = \#$ then $(c_l^i - nv_l^j) = 0$ or $(c_l^i - c_l^j) = 0$ or $(nv_l^i - nv_l^j) = 0$.

Note that when all features are taking part in chromosome building, equations 5, 6 and 7 will become same as equations 1, 2 and 3 respectively.

Different steps of MOGA_Feature_Selection (H, S) are described below. It is using equations 5, 6 and 7.

A flowchart of MOGA (H, S) and MOGA_Feature_Selection (H, S) is provided in figure 1.

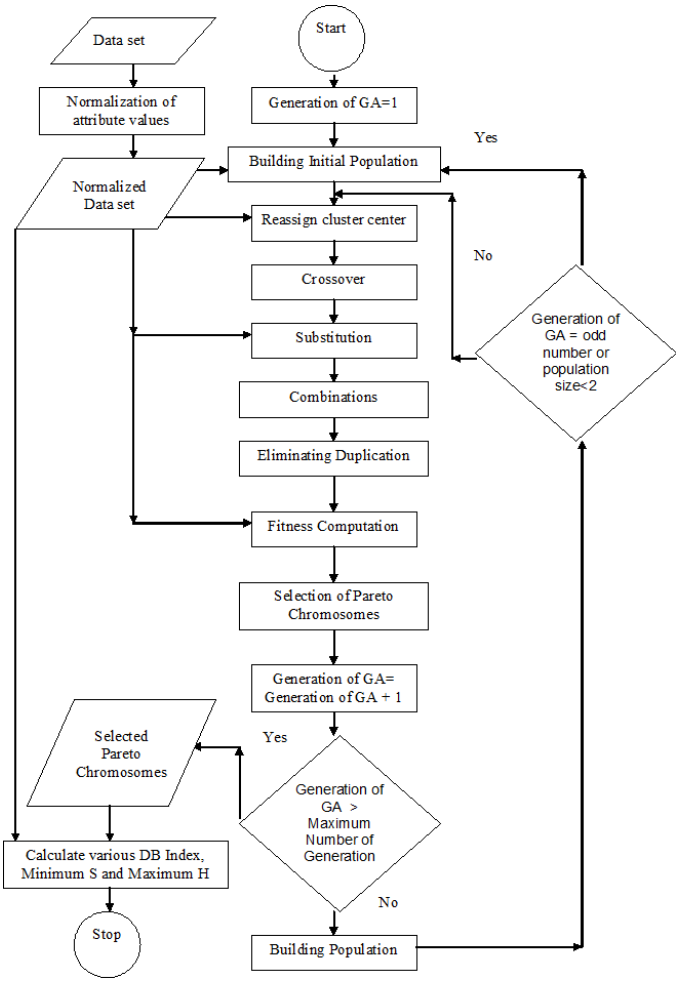

Fig. 1. Flowchart of MOGA (H, S) and MOGA_Feature_Selection (H, S)

A. Building Initial Population

Initial population size (IPS) is the nearest integer value of 10% of data set size. Random number of features build chromosomes. Although correlation between IP size and the number of instances in the data set are not known, we think IP size guides searching power of GA and therefore its size should increase with the size of the data set. In other words, this is better than fixed IPS for all data set often used in literature. Here Ch_j represents the j^{th} chromosome in the population, where $1 \leq j \leq IPS$. Each chromosome represents a prospective solution, which is a set of cluster centers (CCs) of K clusters. As tuples of data set build CCs, it induces faster convergence of MOGA, compared to building chromosomes by randomly choosing continuous feature values from the same feature domain. Detailed algorithm is available in [12] for categorical features.

B. Reassign cluster centers

Every chromosome delivers a set of CCs that is $\{CC_1, CC_2, ...CC_K\}$, which is equivalent to step 1 of K-means algorithm where randomly chosen records builds CCs. However, in MOGA_Feature_Selection (H, S), all features are not building CCs. For valid features only, one iteration of K-means algorithm produces new CCs that is $\{CC_1^*, CC_2^*, ...CC_K^*\}$, which forms new chromosomes to replace chromosomes used as input of K-means algorithm. Detailed K-means algorithm is available in [27], [28]. Some recent works based on K-means algorithms are [3], [32]. Similar types of algorithms with GA are also a available in [10], [29].

C. Crossover

Context sensitivity is an important issue in a grouping task like clustering. Meaning of context insensitivity is "the schemata defined on the genes of the simple chromosomes do not convey useful information that could be exploited by the implicit sampling process carried out by a clustering GA" [13]. In their survey paper Hruschka et. al. [21] shows drawback of conventional single point crossover operators often described in the literature considering context sensitivity. We have used a special crossover operator called "Pairwise crossover" described by Fränti in [14] as "The clusters between the two solutions can be paired by searching the "nearest" cluster (in the solution B) for every cluster in the solution A. Crossover is then performed by taking one cluster centroid (by random choice) from each pair of clusters. In this way we try to avoid selecting similar cluster centroid from both parent solutions. The pairing is done in a greedy manner by taking for each cluster in A the nearest available cluster in B. A cluster that has been paired cannot be chosen again, thus the last cluster in A is paired with the only one left in B." This algorithm does not give the optimal pairing (2-assignment) but it is a reasonably good heuristic for the crossover purpose. Crossover probability (P_{co}), chosen as 0.9 that is 90% chromosomes undergo with this crossover.

Parent chromosomes must have the same set of valid and invalid features to calculate distance between CCs. So two extra steps are needed for crossover of MOGA_Feature_Selection (H, S). At first, one chromosome is selected randomly from the population to consider a set of valid and invalid features from population generated by reassign cluster centers method (section IV-B). As a second step, population for crossover is built by selecting chromosomes having the same set of valid and invalid features as the randomly selected chromosome. Crossover is done and it changes some chromosomes. Child chromosomes replace parent chromosomes in the original population formed by reassign cluster centers method (section IV-B).

Initially number of parent chromosomes = IPS. All Ch generated by K-means described in section IV-B are not $PChs$ of crossover. A subset of population having same set of valid and invalid features forms population for crossover. So some $PChs$ are participating in crossover and others are copied as it is for next operation. Similar type of algorithm for categorical features are discussed in [12].

D. Substitution

Substitution probability (P_{sb}) is 0.1. Chromosomes after crossover become parent chromosomes to this. Here $dom(A_i)$ is continuous. In conventional mutation, any random number in the range of 0 to 1 can replace nv_i, resulting many chromosomes. Considering context sensitivity, instead of replacing any nv_i of chromosome, substitution is replacing CCs by any tuples randomly. Approximately number of substitution (N_{sb}) = $\lfloor(N_{pch} \times K \times P_{sb})\rfloor$. Only valid features of CCs are substituted by feature values of any randomly chosen tuple. Similar type of algorithm for categorical features are discussed in [12].

E. Combination

At the end of every generation of MOGA some chromosomes are lying on the Pareto optimal front. These chromosomes have survived and are known as Pareto chromosomes. Chromosomes obtained from previous substitution method (algorithm discussed in section IV-D), say NC_{pre} and previous generation Pareto chromosomes (section IV-H) of MOGA, say PC_{pre} are considered to perform in the next generation. For the first generation of GA, PC_{pre} is zero because of the nonexistence of previous generation.

In general, for i^{th} generation of MOGA, if NC_{pre} is m and PC_{pre} at the end of $(i-1)^{th}$ generation of MOGA is n then after combination, number of chromosomes are $n+m$ or $PC_{pre} + NC_{pre}$.

F. Eliminating duplication

Survived chromosomes from previous generation may be generated again in the present generation of MOGA, resulting multiple copies of same chromosome. So in this step system is selecting unique chromosomes from the combined chromosome set.

G. Fitness Computation

Modified intra-cluster distance (Homogeneity) (H_{mod}) and modified inter-cluster distances (Separation) (S_{mod}) are two measures for calculating fitness values of MOGA_Feature_Selection (H, S) clustering algorithm. Maximization of 1/H_{mod} and S_{mod} are the objectives of MOGA. It converts optimization problem into max-max framework.

In MOGA (H, S) equation 1 calculates the Euclidean distance per feature between one cluster center and one tuple.

If $d_{lowest}(CC_i, t_j)$ is the lowest distance for any value of i (where $1 \leq i \leq K$), then t_j is assigned to C_i. Equation 8 defines H_i (average intra-cluster distance or homogeneity of i^{th} cluster).

$$H_i = [\sum_{j=1}^{m} d_{lowest}(CC_i, t_j)]/m_i \tag{8}$$

where m_i is the number of tuples belonging to i^{th} cluster and t_j is assigned to i^{th} cluster based on lowest distance.

Summation of H_i is H, defined in equation 9 as

$$H = \sum_{i=1}^{K} H_i \tag{9}$$

Say, t_j and t_p are two distinct tuples from a data set where t_j is assigned to C_x and t_p is assigned to C_y. S is defined in equation 10 using $d_{lowest}(CC_i, t_j)$ as

$$S = \sum_{j,p=1}^{m} d(t_j, t_p) \tag{10}$$

where $j \neq p$ and $x \neq y$.

In MOGA_Feature_Selection (H, S) equations 8, 9 and 10 are modified into equations 11, 12 and 13 respectively.

$$H_{mod_i} = [\sum_{j=1}^{m} d_{mod_{lowest}}(CC_i, t_j)]/m_i \tag{11}$$

$$H_{mod} = \sum_{i=1}^{K} H_{mod_i} \tag{12}$$

$$S_{mod} = \sum_{j,p=1}^{m} d_{mod}(t_j, t_p) \tag{13}$$

As we are using Euclidean distance per feature and all normalized features, $1/H_{mod}$ and S_{mod} values of different chromosomes having different sets of valid features become comparable with one another.

Note that when all features are taking part in chromosome building equations 11, 12 and 13 become same as equations 8, 9 and 10 respectively.

In MOGA_Feature_Selection (H, S) system calculates $1/H_{mod}$ and S_{mod} for every chromosome in the population. In this case equations 5, 6 and 7 have been used in place of equations 1, 2 and 3 respectively of MOGA (H, S).

H. Selection of Pareto chromosomes

All chromosomes lying on the front of max($1/H_{mod}$, S_{mod}) are Pareto chromosomes. In other words maximizing $1/H_{mod}$ and S_{mod} are two objectives of MOGA. As this process is applied after combination, elitism [5], [6] is adhered to.

I. Building intermediate population

From 2^{nd} generation onwards for every even generation, selected Pareto chromosomes of previous generation build population. This helps to build a population of chromosomes close to the Pareto optimal front, resulting convergence of MOGA. For every odd generation of MOGA or if for any generation previous generation of MOGA does not produce Pareto chromosomes greater than 2 then population are created using algorithm discussed in section IV-A. This introduces new chromosomes in population and induces diversity in the population. Population size is also varies from one generation to another.

V. Testing and Comparison

In literature different validity indices are available to measure the goodness of clusters. C Index [22], Dunn Index [8] and Davies-Bouldin (DB) Index [4] are some of them. Among these we have used Davies-Bouldin (DB) Index [4] here. Equation 14 is calculating DB Index.

$$DB\ Index = 1/K \sum_{i=1, i\neq j}^{K} max((H_{mod_i} + H_{mod_j})/d_{mod}(CC_i, CC_j)) \quad (14)$$

where K is the number of clusters, H_{mod_i} is the average distance of all patterns in cluster i to their cluster center CC_i, H_{mod_j} is the average distance of all patterns in cluster j to their cluster center CC_j and $d_{mod}(CC_i, CC_j)$ is the distance between cluster centers CC_i and CC_j. For MOGA (H, S) equation 14 is using equations 1 and 2 whereas for MOGA_Feature_Selection (H, S) equation 14 is using equations 5 and 6.

5 popular data sets from UCI Machine Learning Repository [30] compares the performance of three clustering algorithms in table I. For every data set for every clustering algorithm average values of 10 individual runs are tabulated.

MOGA (H, S) is better than K-means and MOGA_Feature_Selection (H, S) outperforms MOGA (H, S) almost in all aspects for 5 data sets. It clearly shows the effectiveness of simultaneous attribute selection and optimization using MOGA.

VI. Conclusions

In this work, we have implemented a version of a novel real coded hybrid elitist MOGA for K-clustering (MOGA_Feature_Selection (H, S)). It is known that elitist model of GAs provides the optimal string as the number of iterations increases to infinity [2]. Due to the use of special population building process, hybridization of K-means with GA, special crossover and substitution operator GAs are producing good clustering solution in smaller numbers of iteration (100). MOGA_Feature_Selection (H, S) is giving best results among 2 algorithms (MOGA (H, S) and MOGA_Feature_Selection (H, S)). It shows the effectiveness of simultaneous feature selection.

We have achieved one important data mining task of clustering by finding cluster centers. We have considered only continuous features in this work. Categorical features need other type of encoding, as Euclidean distance per feature measures is not suitable for those types of features. Dealing with missing features values and unseen data are other problem areas. In the present form algorithm is not dealing with missing values in data. It may be interesting to adapt MOGA based K-clustering with unknown number of clusters. Authors are working in these directions.

References

[1] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, May. 2007, pp. 1506-1511, doi:10.1109/TGRS.2007.892604.

[2] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic Algorithm with elitist model and its convergence," *Int. J. Patt. Recogn. Artif. Intell.*, vol. 10, no. 06, Sep. 1996, pp. 731–747, doi: 10.1142/S0218001496000438.

[3] M. C. Chiang, C. W. Tsai, and C. S. Yang, "A time-efficient pattern reduction algorithm for k-means clustering," *Information Sciences*, vol. 181, no. 4, Feb. 2011, pp. 716–731, doi: 10.1016/j.ins.2010.10.008.

[4] D. L. Davies, and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, Apr. 1979, pp. 224-227, doi: 10.1109/TPAMI.1979.4766909.

[5] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: John Wiley & Sons, 2001.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, Apr. 2002, pp. 182-197, doi:10.1109/4235.996017.

[7] K. Dhiraj, and S. K. Rath, "Comparison of SGA and RGA based clustering algorithm for pattern recognition," *Int. J. of Recent Trends in Engg.*, vol. 1, no. 1, May. 2009, pp. 269–273.

[8] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *J. Cybernetics*, vol. 3, no. 3, 1973, pp. 32–57, doi: 10.1080/01969727308546046.

[9] D. Dutta, P. Dutta, and J. Sil, "Clustering by multi objective genetic algorithm," in *Proc. of 1st Int. Conf. on Recent Advances in Information Technology (RAIT 2012)*, IEEE, Dhanbad, India, Mar. 2012, pp. 548–553, doi:10.1109/RAIT.2012.6194619.

[10] D. Dutta, P. Dutta, and J. Sil, "Clustering Data Set with Categorical Feature Using Multi Objective Genetic Algorithm," in *Proc. Int. Conf. on Data Science and Engineering (ICDSE 2012)*, IEEE, Cochin, Kerala, India, Jul. 2012, pp. 103–108, doi: 10.1109/ICDSE.2012.6281897.

[11] D. Dutta, P. Dutta, and J. Sil, Data Clustering with Mixed Features by Multi Objective Genetic Algorithm, in: Proc. of the 12th Int. Conf. on Hybrid Intelligent Systems (HIS 2012), Dec. 2012, IEEE, 2012, pp. 336–341.

[12] D. Dutta, P. Dutta, and J. Sil, Simultaneous Feature Selection and Clustering for Categorical Features Using Multi Objective Genetic Algorithm, in: Proc. of the 12th Int. Conf. on Hybrid Intelligent Systems (HIS 2012), Dec. 2012, IEEE, 2012, pp. 191–196.

[13] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. New York, NY: John Wiley & Sons, 1998.

[14] P. Fränti, J. Kivijärvi, T. Kaukoranta, and O. Nevalainen, "Genetic algorithms for large-scale clustering problems," *The Computer Journal*. vol. 40, no. 9, Sep. 1997, pp. 547-554, doi: 10.1093/comjnl/40.9.547.

[15] D. E. Goldberg, *Genetic Algorithms for Search, Optimization, and Machine Learning*. 1st ed., Reading, MA: Addison-Wesley Longman, 1989.

[16] L. O. Hall, I. B. Özyurt, and J. C. Bezdek, "Clustering with a genetically optimized approach," *IEEE Trans. Evol. Comput.* vol. 3, no. 2, Jul. 1999, pp. 103–112, doi:10.1109/4235.771164.

[17] J. Han, and M. Kamber, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann, 2005.

[18] J. Handl, and J. Knowles, "Evolutionary multiobjective clustering," in *Proc. 8th Int. Conf. Parallel Problem Solving From Nature (PPSN VIII)*, Birmingham, UK, 2004, pp. 1081–1091. X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, and H. Schwefel, Eds., in *Lecture Notes in Computer Science*, chapter 109, vol. 3242, pp. 1081–1091, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, doi:10.1007/978-3-540-30217-9_109.

[19] J. Handl, and J. Knowles, "Exploiting the Trade-off - The Benefits of Multiple Objectives in Data Clustering," in *Proc. of 3rd Int. Conf. on Evolutionary Multi-Criterion Optimization (EMO 2005)*, Guanajuato, Mexico, 2005, pp. 547–560. C. A. Coello Coello, A. H. Aguirre, and E. ZitzlerEds., in *Lecture Notes in Computer Science*, chapter 38, vol. 3410, pp. 547–560, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, doi:10.1007/978-3-540-31880-4_38.

[20] J. Handl, and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Computat.*, vol. 11, no. 1, Feb. 2007, pp. 56-76, doi:10.1109/TEVC.2006.877146.

TABLE I
COMPARISON OF PERFORMANCE OF CLUSTERING ALGORITHMS

Meaning of columns used in this table

A1:H of K-means, A2:S of K-means, A3:DB Index of K-means

1:Optimized min. H of population, 2:Optimized max. S of population, 3:DB Index of chromosome giving optimized min. H of population, 4:DB Index of chromosome giving optimized max. S of population, 5:Average DB Index of population, 6:Best DB Index of population

Methods	K-means			MOGA (H, S)						MOGA_Feature_Selection (H, S)					
Data set	A1	A2	A3	1	2	3	4	5	6	1	2	3	4	5	6
Glass	0.58	4242.88	1.19	0.38	4404.03	1.08	1.72	1.53	0.88	0.15	5856.44	0.91	1.23	2.87	0.47
Heart	0.58	9489.67	1.85	0.62	12267.33	1.68	1.94	1.84	1.50	0.29	16079.86	0.59	0.95	0.81	0.50
Iris	0.30	2991.45	0.80	0.26	3087.85	0.73	0.79	0.78	0.60	0.15	3748.00	0.84	0.39	1.66	0.39
Sonar	0.37	3252.50	2.12	0.20	3354.27	0.76	2.54	2.21	0.74	0.22	3607.76	0.83	1.88	1.96	0.87
Wine	0.41	3307.81	1.32	0.31	3315.29	1.03	1.45	1.17	0.96	0.25	3609.35	0.84	0.99	0.96	0.77

[21] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Trans. Syst. Man Cybern. C*, vol. 39, no. 2, Mar. 2009, pp. 133–155, doi:10.1109/TSMCC.2008.2007252.

[22] L. Hubert, and J. Schultz, "Quadratic assignment as a general data-analysis strategy," *British J. of Mathematical and Statistical Psychology*, vol. 29, no. 2, Nov. 1976, pp. 190–241, doi: 10.1111/j.2044-8317.1976.tb00714.x.

[23] Y. Kim, W. N. Street, and F. Menczer, "Feature selection in unsupervised learning via evolutionary search," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Dis. Data Mining*, Boston, MA, USA, Aug. 2000, pp. 365-369.

[24] E. E. Korkmaz, J. Du, R. Alhajj, and K. Barker, "Combining advantages of new chromosome representation scheme and multi-objective genetic algorithms for better clustering," *Intell. Data Anal.*, vol. 10, no. 2, Mar. 2006, pp. 163-182.

[25] M. H. C. Law, A. P. Topchy, and A. K. Jain, "Multiobjective data clustering," in *Proc. IEEE Comput. Soc. Conf. on Comput. Vision and Pattern Recognit (CVPR 2004)*, vol. 2, IEEE press, Washington, DC, Jun.-Jul. 2004, pp. 424–430, doi:10.1109/CVPR.2004.1315194.

[26] H. Liu, and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 1-12, Apr. 2005. doi:10.1109/TKDE.2005.56.

[27] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theor.*, vol. IT-28, no. 2, Mar. 1982, pp. 129–137, doi: 10.1109/TIT.1982.1056489.(Original version: Technical Report, Bell Labs, 1957).

[28] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, Statistical Laboratory of the University of California, Berkeley, 1967, pages 281–297.

[29] U. Maulik, and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognit.*, vol. 3, no. 9, Sep. 2000, pp. 1455–1465.

[30] C. J. Mertz, P. M. Murphy, UCI repository of machine learning databases. Univ. California, Irvine, Dept. of Information and Computer Sciences, 1998. Available from: ¡http://www.ics.uci.edu/ mlearn/MLRepository.html¿.

[31] P. Merz, and A. Zell, "Clustering gene expression profiles with memetic algorithms," in *Proc. of the 7th Int. Conf. on Parallel Prob. Solving from Nature (PPSN VII)*, Lecture Notes in Computer Science, chapter 78, vol. 2439. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 811-820, doi:10.1007/3-540-45712-7_78.

[32] B. K. Mishra, N. R. Nayak, A. K. Rath, and S. Swain, "Far Efficient K-Means Clustering Algorithm," in *Proc. of the Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI 2012)*, ACM, Chennai, India, Aug. 2012, pp. 106–110, doi: 10.1145/2345396.2345414.

[33] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Multiobjective genetic fuzzy clustering of categorical attributes," in *Proc. 10th Int. Conf. on Inf. Technol. (ICIT 2007)*, IEEE, Orissa, India, Dec. 2007, pp. 74-79, doi: 10.1109/ICIT.2007.13.

[34] H. Pan, J. Zhu, and D. Han, "Genetic algorithms applied to multi-class clustering for gene expression data," *Genomics, Proteomics, Bioinformatics*, vol. 1, no. 4, Nov. 2003, pp. 279–287.

[35] D. Pyle, Data Preparation for Data Mining, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[36] K. S. N. Ripon, C. H. Tsang, and S. Kwong, "Multi-objective data clustering using variable-length real jumping genes genetic algorithm and local search method," in *Proc. of the Int'l Joint Conf. on Neural Networks (IJCNN 2006)*, IEEE, Vancouver, BC, Canada, Jul. 2006, pp. 3609–3616, doi: 10.1109/IJCNN.2006.247372.

[37] K. S. N. Ripon, C. H. Tsang, S. Kwong, and M. Ip, "Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm," in *Proc. of the 18th Int'l Conf. on Pattern Recognition (ICPR 2006)*, IEEE, Hong Kong, China, Aug. 2006, pp. 1200–1203, doi:10.1109/ICPR.2006.827.

[38] K. S. N. Ripon, and M. N. H. Siddique, "Evolutionary multi-objective clustering for overlapping clusters detection," in *Proc. of the 11th Conf. on Congress on Evolutionary Computation (CEC 2009)*, IEEE, Piscataway, NJ, May. 2009, pp. 976–982, doi:10.1109/CEC.2009.4983051.

[39] H. Samet, *Foundations of Multidimensional and Metric Data Structures*. Amsterdam, UA:Elsevier, 2006.

[40] P. Scheunders, "A genetic C-means clustering algorithm applied to color image quantization," *Pattern Recognition*, vol. 30, no. 6, Jun. 1997, pp. 859-866, doi: 10.1016/S0031-3203(96)00131-8.

[41] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM Review*, vol. 35, no. 4, Dec. 1993, pp. 551-566, doi:10.1137/1035134.