A Survey of Multi-Objective Evolutionary Algorithms for Data Mining: Part-II

Anirban Mukhopadhyay, *Senior Member, IEEE*, Ujjwal Maulik, *Senior Member, IEEE*, Sanghamitra Bandyopadhyay, *Senior Member, IEEE*, and Carlos A. Coello Coello, *Fellow, IEEE*

***Abstract*—This paper is the second part of the two-part paper on the survey of multi-objective evolutionary algorithms for data mining problems. In Part-I of the paper [1], multi-objective evolutionary algorithms used for feature selection and classification have been reviewed. In this Part-II, different multi-objective evolutionary algorithms used for clustering, association rule mining and other data mining tasks are surveyed. Moreover, a general discussion is provided along with scopes for future research in the domain of multi-objective evolutionary algorithms for data mining.**

***Index Terms*—Multi-objective evolutionary algorithms, clustering, association rule mining, biclustering, ensemble learning.**

I. Introduction

As mentioned in Part-I of the paper [1], multi-objective evolutionary algorithms (MOEAs) [2] have become increasingly popular in the domain of data mining during the last decade. In this two-part paper, we survey several MOEAs for different data mining tasks. In Part-I [1], we have introduced the basic concepts of multi-objective optimization and data mining and reviewed different MOEAs designed for addressing two important data mining tasks, namely feature selection and classification.

In this Part-II of the paper, MOEAs used for two other major data mining tasks, viz., clustering [3] and association rule mining [4] are surveyed. Both of these data mining tasks are unsupervised in nature and can be easily posed as multi-objective optimization problems. In recent years, several MOEAs have been proposed in the literature to accomplish these tasks. We review many of these approaches with focus on chromosome representation, objective functions, evolutionary operators, and methods for obtaining the final solution from the non-dominated set. Besides this, here we also review MOEAs employed for several other data mining tasks such as ensemble learning, biclustering, feature extraction, sub-group discovery and so on. Fig. 1 shows the different MOEAs-based data mining tasks reviewed in this part of the paper along with the corresponding references. A general discussion on the future scope of research in this area of multi-objective data mining is also provided.

Anirban Mukhopadhyay is with the Department of Computer Science & Engineering, University of Kalyani, India. $Email:\ anirban@klyuniv.ac.in$

Ujjwal Maulik is with the Department of Computer Science & Engineering, Jadavpur University, India. $Email:\ umaulik@cse.jdvu.ac.in$

Sanghamitra Bandyopadhyay is with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. $Email:\ sanghami@isical.ac.in$

Carlos A. Coello Coello is with CINVESTAV-IPN, Departamento de Computación (Evolutionary Computation Group), México, D.F., México. $Email:\ ccoello@cs.cinvestav.mx$

II. MOEAs for Clustering

Clustering techniques aim to find a suitable grouping of the input dataset so that some criteria are optimized. A straightforward way to pose clustering as an optimization problem is to optimize some cluster validity index [5] that reflects the goodness of the clustering solutions. All possible partitionings of the dataset and the corresponding values of the validity index define the complete search space. Under this context, genetic and other evolutionary algorithms have been widely used to reach the global optimum value of the chosen validity measure. Conventional evolutionary clustering techniques [6] use some validity measure as the fitness value. However, no single validity measure works equally well for different kinds of datasets. Thus, it is natural to simultaneously optimize multiple of such measures for capturing different characteristics of the data. Hence it is useful to utilize MOEAs for clustering. Multi-objective clustering techniques optimize more than one cluster validity index simultaneously, leading to high-quality results. The resultant set of near-Pareto-optimal solutions contains a number of non-dominated solutions, from which the user has to select the most appropriate one based on his/her own preferences. A number of multi-objective evolutionary clustering algorithms are available in the literature. They vary in different aspects, including the type of MOEA, the chromosome encoding, the objective functions optimized, the evolutionary operators adopted and the mechanism adopted to select the final solution from the non-dominated front.

A. Underlying MOEAs

There are mainly four MOEAs that have been used as the underlying optimization tool for multi-objective clustering. Pareto Envelope-Based Selection Algorithm-II (PESA-II) [59] has been used in the algorithms VIENNA (Voronoi Initialized Evolutionary Nearest-Neighbor Algorithm) [7], MOCK-AM [21] (Multi-Objective Clustering with automatic K determination Around Medoids), MOCK [10], and MECEA (Multi-objective Evolutionary Clustering Ensemble Algorithm) [13]. Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [60] has been employed in many multi-objective clustering approaches such as MOEA (Dynamic) [9], VRJGGA (Variable-length Real Jumping Genes Genetic Algorithms) [61], MOGA [12], MOGA(medoid) [22], MOES (Hybrid) (Multi-Objective Evolutionary Strategy (Hybrid)) [14], MOGA-SVM (Multi-objective GA with Support Vector Machine) [15], [16], EMCOC (Evolutionary Multi-Objective Clustering for Overlapping Clusters detection) [18], MOGA (mode) [24], DYN-

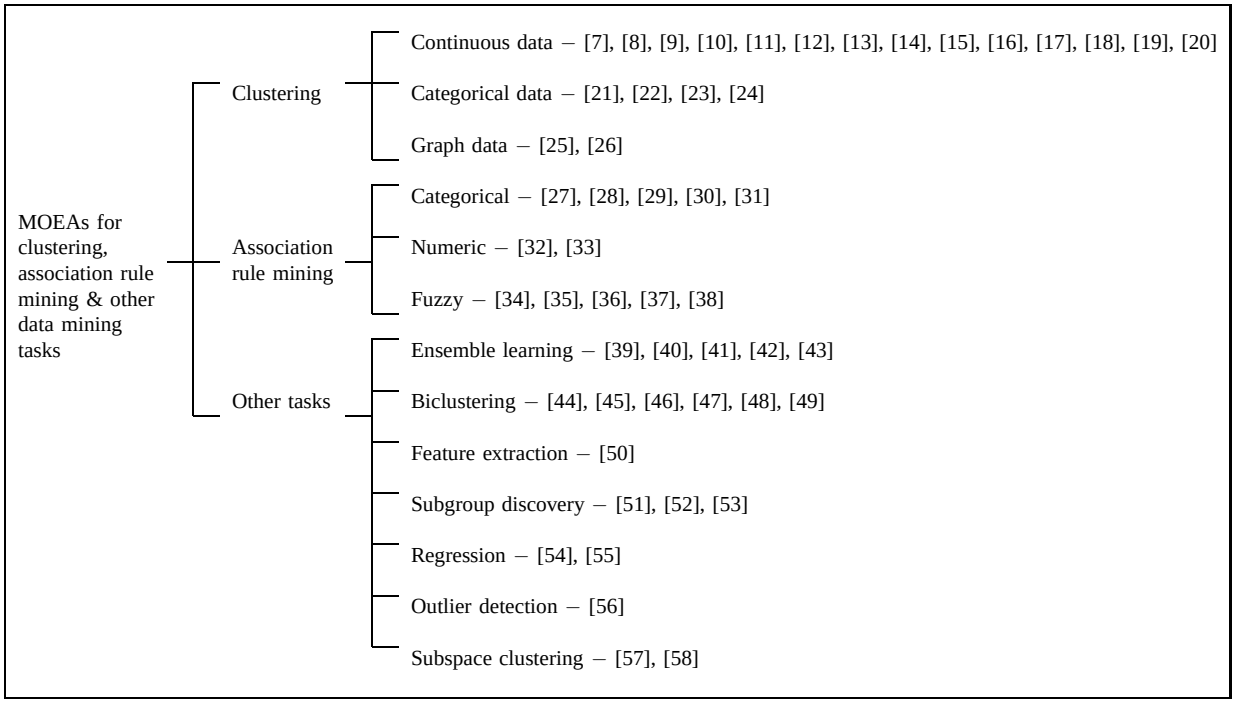

Fig. 1. MOEAs for clustering, association rule mining and other data mining tasks surveyed in this part of the article (with references)

MOGA (Dynamic MOGA) [26], MOVGA (Multi-objective Variable-length Genetic Algorithm) [19], and MOCA (Multi-Objective Clustering Algorithms) [20]. In [23] and [17], Strength Pareto Evolutionary Algorithm-2 (SPEA2) [62] has been used as the underlying optimization tool. Niched Pareto Genetic Algorithm (NPGA) [63] has been employed in MOKGA (Multi-objective K-Means Genetic Algorithm) [8].

B. Chromosome Representation

The choromosome representation approaches can broadly be classified into two major classes, viz., prototype-based approaches and point-based approaches. In the prototype-based approach, cluster representatives or prototypes, such as cluster centroids, medoids and modes are encoded in the chromosome. On the other hand, in the point-based approach, a complete clustering solution is encoded in the chromosome.

In a prototype (cluster center)-based approach, the chromosomes are made up of real numbers which represent the coordinates of the cluster centers. If a chromosome encodes the centers of K clusters in d-dimensional space, then its length l will be $d \times K$. In case of multi-objective clustering, this encoding scheme was first utilized by Mukhopadhyay, Maulik and Bandyopadhyay in [64] and the authors have used this encoding policy in a series of multi-objective clustering algorithms such as MOGA [12], SiMM-TS (Significant Multi-class Membership (Two-stage)) [65], MOGA-SVM [15], [16] and MOVGA [19]. Besides Mukhopadhyay et al., several other researchers have adopted this encoding policy in different multi-objective clustering algorithms such as VRJGGA [61], MOES(Hybrid) [14], and MOCA [20]. In some algorithms, instead of using cluster centers as cluster prototypes, cluster medoids have been encoded in the chromosome. A cluster medoid is the point of the cluster, from which the sum of the distances to the other points of the cluster is minimum. There are some approaches which encode the cluster medoids, or the indices of the points representing the cluster medoids in the chromosomes. Examples of such multi-objective evolutionary clustering algorithms include MOGA (medoid) [22], and EMCOC [18]. Another approach is to encode cluster modes in the chromosomes. Cluster modes are suitable for categorical attributes where the mean centroid of the cluster cannot be computed. Given a set of categorical points, their mode is defined as a vector of the attributes where each component value of the vector represents the most frequent value occurring in the corresponding attribute over all the points. MOGA (mode) [24] is a multi-objective clustering algorithm where cluster modes are encoded in the chromosomes. The advantage of prototype-based encoding is that here the length of the chromosomes is small and, therefore, it takes less time to apply the evolutionary operators such as crossover and mutation. Also, this encoding policy is good for capturing overlapping and fuzzy clusters. However, these algorithms have a tendency to capture round-shaped clusters only. Also, if the chromosomes encode different number of clusters, they have variable lengths which are to be handled while applying the evolutionary operators. Moreover, in this type encoding, the chromosomes may be very large if the number of attributes is large. Therefore, for higher dimensional datasets, this encoding strategy may not work very well.

Another popular encoding approach is point-based encoding, where the complete clustering of the data points are encoded instead of only the representatives/prototypes of the clusters. Under this scheme, there are two main approaches, viz., cluster label-based approach and locus-based adjacency representation. The cluster label-based approach is the most common form of point-based encoding. Here, the chromosome

lengths are equal to the number of points in the input dataset, and each position represents the cluster label of the corresponding points. If position i of the chromosome contains a value k, then the ith data point is assigned to cluster k. Obviously, the chromosomes can contain only integer values drawn from the set $\{1, 2, \ldots, K\}$, where K is the maximum number of clusters. The multi-objective clustering algorithms that use this encoding policy include VIENNA [7], MOKGA [8], and GraSC (Graph-based Sequence Clustering) [23], [66]. In MOCK [10], [21], Handl and Knowles used a variant of the cluster label-based encoding strategy. Here, each chromosome consists of n genes (n is the number of data points) and each gene can have integer values in $\{1, \ldots, n\}$. If the gene i is assigned a value j, it represents a link between the data points i and j, and, in the resulting clustering solution, these two points will belong to the same cluster. Thus, a graph is formed with the data points as the vertices and the links between two data points are the edges. Therefore, for decoding a chromosome, it is required that we identify all the connected components of the graph. This can be done in linear time [10]. The data points in the same connected component are then assigned to the same cluster. Hence, this representation encodes the clustering as well as the number of clusters (number of connected components). Many algorithms besides MOCK, such as MECEA [13], AI-NSGA-II [25], and DYN-MOGA [26] have adopted this encoding policy. Although point-based encoding techniques are not biased towards convex-shaped clusters, they suffer from the large length of chromosomes when the number of data points n is large. Thus, the algorithms using this encoding approach require more time to converge. However, unlike prototype-based encoding, here the chromosome length is independent of the encoded number of clusters.

C. Objective Functions

For the clustering problem, usually cluster validity indices [67] are used as the objective functions. Most of such multi-objective clustering algorithms have used two validity indices to be simultaneously optimized. In [7], [10], [68], the MOCK clustering algorithm minimizes two validity indices: overall cluster deviation ($Dev(C)$) and cluster connectedness ($Conn(C)$). Some other multi-objective clustering works have also used these two objectives [9], [13], [66]. In [12], [15], [16] the authors used two validity indices, J_m [69] and XB [70] which are minimized simultaneously to obtain compact and well-separated clusters. In [8], [71], [72], the two validity indices to be minimized are: Total Within-Cluster Variance ($TWCV$) and the number of clusters K. In [22], a multi-objective categorical data clustering algorithm is used to optimize: overall deviation $Dev(C)$ (with respect to medoids instead of centroids) and silhouette index [73]. In [11], [18], the intra-cluster entropy H and cluster separation $Sep(C)$ are used as the two objective functions. The index $\mathcal{I}$ [67] and XB are simultaneously optimized in [74]. In [23], [25], [66], the objectives adopted are: Min-Max Cut and the silhouette index [73]. In [17], the aim is to obtain compact and well-separated clusters and for that sake, the objectives to be minimized are: the validity indices overall deviation $Dev(C)$ and the Edge index $Edge(C)$. In [19], [24], [75], the objective functions are chosen to be the normalized J_m index ($\mathcal{J}$) and the fuzzy cluster separation $\mathcal{S}$ which are simultaneously minimized. It is to be noted that instead of cluster centroids, cluster modes have been used for computing the validity index values in [24], [75], since these algorithms have been applied on categorical data. In [76], out of several combinations, DB [77] and $Dunn$ [78] indices have also been chosen as the two objectives to be simultaneously optimized. The indices J_m and cluster separation have been used in [79]. There are also a few multi-objective clustering techniques which use more than two objective functions. For example, in [74], [80], three cluster validity measures, viz., XB index, $\mathcal{I}$ index and J_m index have been simultaneously optimized. In [20], three objective functions have been simultaneously optimized as well: average cluster variance, average between group sum of squares ($ABGSS$) and cluster connectedness. In [81], four objective functions are considered: overall cluster deviation, cluster separation, cluster dominance and the diameter of the biggest cluster. It is known that MOEAs usually do not perform very well when the number of objective functions increases to four or more [82]. However, in [81], the authors have not addressed this issue. It should be noted that the choice of a suitable set of objective functions is not a trivial problem, and the clustering output may heavily depend on this choice [83]. In view of this, recently, an interactive multi-objective clustering algorithm was proposed in [84]. In this approach the algorithm interacts with a human decision maker to learn the suitable set of objective functions along with evolving the clustering solution. However, a detailed study that compares the effects of different objective functions is still missing.

D. Evolutionary Operators

Evolutionary operators, such as crossover and mutation, depend on the adopted chromosome representation scheme. Many of the algorithms employing prototype-based representation have adopted single-point crossover. Examples of such multi-objective clustering algorithms include MOGA [12], MOGA-SVM [15], [16] and MOVGA [19]. In [9], two-point crossover has been used. Ripon et al. have employed jumping gene crossover in their multi-objective clustering algorithms [61], [18]. In [14], the authors have used a centroid-pool based crossover approach where the centroids encoded in the parent chromosomes are first combined to build a centroid-pool. Thereafter, an offspring solution is generated by randomly selecting a number of chromosomes from the centroid pool. The algorithms that employ a point-based encoding policy have used uniform crossover in most cases [7], [10], [13], [23], [25], [26]. Following the crossover operators, a variety of mutation operators are also employed. Mutation refers to small changes in the chromosomes and is used for maintaining the diversity of the population. In prototype-based encoding, the predominant mutation operator found is centroid perturbation [9], [12], [14], [15], [16], [19], [61]. The basic idea of this mutation operator is to shift a randomly selected centroid slightly from its current position. For medoid-based encoding and mode-based encoding, the mutation operators random medoid

replacement [22] and mode perturbation [24] have been used, respectively. In [20], a mutation operator is employed in which either random cluster centers of the chromosomes are perturbed or cluster centers are added/deleted to/from the chromosome with equal probability. For the cluster label-based encoding, the common approach for mutation is to replace the class label of the selected point by a random class label. This mutation operator has been adopted in [66], [85]. To tackle the problem of dealing with a large chromosome length, a special mutation operator, called directed neighborhood-biased mutation was proposed in [10]. In this mutation, each point i is linked to its L nearest neighbors $\{nn_{i1}, nn_{i2}, \ldots, nn_{iL}\}$, and thus the effective search space is reduced to L^n. Thus, changing the class label of point i induces the change to all its L nearest neighbors. The mutation probability is also decided adaptively. The same mutation operator has been used in many other algorithms [13], [17], [25], [26].

E. Obtaining a Final Solution

MOEAs-based clustering algorithms also differ in the method for obtaining the final solution from the non-dominated set of solutions yielded by the MOEA. These methods can be broadly classified into three categories, viz., independent objective-based approach, knee-based approach and cluster ensemble-based approach.

In the independent objective-based approach, an independent cluster validity index, other than those optimized during the clustering process, is used to select a single solution from the non-dominated front. Many of the currently available multi-objective clustering techniques have adopted this approach because of its simplicity. In [12], [64], the authors used the J_m and XB indices as the objective functions, whereas the final solution was selected using index $\mathcal{I}$. In a similar approach [19], fuzzy cluster compactness and separation were adopted as the two objectives whereas the $\mathcal{I}$ index was used as the selection criterion. In [86], the XB and $\mathcal{I}$ indices have been used as the objective functions whereas the silhouette index was used for selecting the final solution. In [71], the two objective functions are $TWCV$ and the number of clusters, whereas the authors used the DB index and the SD index [87], [88] for selecting the final solution from the Pareto front. In [8], the two objective functions used are $TWCV$ and the number of clusters, and various other validity indices, such as the Dunn index, the DB index and the silhouette index are adopted for selecting the final solution. The authors also presented a comparative study of their results. Demir et al., in their GraSC algorithm [66], optimized the silhouette index and the min-max cut index, and used the DB index for selecting the final solution. In [20], the authors optimized three objective functions, viz., average cluster variance, average between group sum of squares ($ABGSS$) and cluster connectedness, and they used the Rand index ($\mathcal{R}$) [5] for selecting the final solution from the Pareto front. Note that computation of $\mathcal{R}$ requires knowing about the true clustering of the dataset. Hence, this method is not applicable when the true clustering information is unknown. Although this approach for selecting the final solution is simple to implement and has low time requirement, the final result may be biased depending on the validity index chosen for selecting the final solution. Moreover, one may criticize this approach by questioning why this independent validity measure is not optimized directly.

The second approach is the knee-based approach, where the objective is to select the knee solution from the non-dominated front. A knee solution refers to an interesting solution for which the change of one objective value induces the maximum change in the other one. Handl and Knowles have used this knee-based approach in their MOCK algorithm [10], [21], [89]. This approach is motivated by the GAP statistic [90]. This is done by comparing the generated Pareto front with control fronts generated by applying MOCK on random control data. The solution that corresponds to the maximum distance between the generated Pareto front and the control fronts is selected as the final solution. However, there is no well-formed motivation behind choosing a knee solution as the final solution. It is not well explained why the user should be most interested on this solution. Another major problem is that it is a time consuming approach, because the algorithm has to be executed multiple times with random datasets to generate the control front. Therefore, a few variants of these technique have been proposed in [91], [92], [17] primarily for improving its scalability for larger datasets.

The third approach is the cluster ensemble-based approach where it is assumed that all the non-dominated solutions contain some information about the clustering structure of the dataset. Therefore, the motivation is to combine this information to obtain a single clustering solution. In [80], some well-known cluster ensemble techniques, such as the Cluster-based Similarity Partitioning Algorithm (CSPA), the HyperGraph Partitioning Algorithm (HGPA) and the Meta-CLustering Algorithm (MCLA) [93] have been used to combine the non-dominated front solutions to obtain the final clustering and their performance is compared by the authors. In a similar approach [13], MCLA has been used for ensembling purposes. In [15], [16], [24], Mukhopadhyay et al. proposed a novel approach for combining the nondominated solutions. Here, the points that are put in the same class by most of the non-dominated solutions are first identified. These points are considered to be highly confident and then, some classifier such as SVM or k-nn is trained using these points. Thereafter, the remaining points are classified by the trained classifier. This way, the class labels for all the points are generated. It has been shown that ensemble-based techniques work better than the independent objective-based techniques [12] for both satellite image segmentation [15] and microarray data clustering [16]. Although these methods are promising and motivating, the ensemble method takes reasonable time and the final solution depends on the choice of the ensemble technique. Also, sometimes it is necessary to map one non-dominated solution to another [24] to ensure that cluster label i means the same cluster in all the solutions. Therefore, the final solution also depends on the mapping technique utilized.

F. Relative Comparison and Applications

We have summarized the processes of well-known MOEAs-based clustering algorithms in Table I. A total of nineteen

different algorithms are considered here. The algorithms are categorized based on the data types where those have been applied, i.e., continuous data, categorical data and graph data. In each category, we have reported the underlying MOEAs, the encoding strategies, the objective functions, the evolutionary operators and the final solution selection methods used by the different clustering methods. The algorithms have been arranged in ascending order of their time of publication to illustrate how they have evolved over time. Out of the nineteen algorithms, ten of them used different versions of prototype-based encoding and the rest used point-based encoding strategies. NSGA-II has been found again to be the most commonly used approach. However, other MOEAs have also been adopted, including PESA-II, NPGA and SPEA2.

MOEAs-based clustering algorithms have found several applications in real-life domains such as image segmentation, bioinformatics, web mining and social networks. Usually, the problem of image segmentation can be posed as the problem of clustering the pixels of the images in the intensity space. If the image has multiple bands, then they serve as the different attributes of the dataset. In [17], a few benchmark color images have been segmented. Maulik et. al. [12], [15], [64] have applied multi-objective fuzzy clustering for segmentation of remote sensing imagery of multi-spectral satellite images. Besides this, the application of multi-objective evolutionary clustering can also be found in the segmentation of MRI medical imagery [19], [80]. Multi-objective clustering has also been applied in texture image segmentation [13]. Another important application area of multi-objective evolutionary clustering algorithms is bioinformatics, where microarray gene expression data sets are clustered to group co-expressed genes. There have been various studies in this area [16], [65], [86], [94], [95]. Multi-objective clustering has also found its application in finding gene markers [96], [97] from expression data. Recently, multi-objective clustering has also been used in clustering protein-protein interaction networks [98]. Multi-objective clustering algorithms have also been applied in web data mining. For example, in [23], a web-recommender system has been built using multi-objective clustering by extracting web usage patterns. An extension of this work is presented in [66] where different multi-objective clustering approaches have been compared for determining a suitable approach for clustering web user sessions, which consist of sequences of web pages visited by the users. In recent times, clustering social networks has gained popularity and a number of recent studies have applied multi-objective clustering techniques to detect strong communities within social networks [25], [26].

III. MOEAs for Association Rule Mining

An association rule can be considered as a general case of a classification rule. The consequent of a classification rule consists of the class attribute only, whereas, in association rules, the consequent may consist of a set attributes. Therefore, the number of association rules for a given dataset is much greater than that of classification rules. Most of the classical association rule mining (ARM) algorithms, such as the *Apriori* algorithm [4], first generate all frequent itemsets (i.e., itemsets having a support greater than the minimum support threshold) and thereafter, from the frequent itemsets, the association rules that surpass the minimum confidence threshold. Generating all the frequent itemsets is itself a time consuming task when the number of items is large, because it needs at least a number k of scans of the dataset for k items. Therefore, it would be beneficial if one could generate the association rules in a direct way, skipping the frequent itemset generation step. For this purpose, evolutionary algorithms have been used widely for generating association rules by maximizing the support/confidence of the rules [99]. However, the goodness of an association rule cannot only be represented by its support or confidence. There are many other metrics available to measure the goodness of an association rule [100]. Therefore, the problem of ARM can be posed as a multi-objective optimization problem where the goal is to find association rules while optimizing several such goodness criteria simultaneously. In the past decade, several MOEAs have been proposed for ARM. These techniques can broadly be classified into three categories, namely categorical association rules, numeric association rules and fuzzy association rules. Here, we discuss several multi-objective evolutionary ARM algorithms from these three categories.

A. Categorical Association Rules

Categorical association rules are generated from a binary or categorical dataset. In a binary dataset, a rule like $ABC \Rightarrow DE$ can be interpreted as follows: if items A, B, and C are purchased, then items D and E are also purchased. Thus, these rules do not say anything about the number of items that are to be purchased; they simply imply the presence or absence of items. For categorical data, if some item has multiple categorical values, then each attribute-value pair is treated as a separate item. This way the dataset is converted into a binary dataset.

1) Underlying MOEAs: Different standard and non-standard MOEAs have been used in various works on categorical ARM. We call a MOEA as non-standard if it does not follow any of the standard MOEA approaches directly, but uses instead some combination of operators. In [27], a multi-objective genetic algorithm (MOGA) is used. In [28], the authors used a multi-objective co-evolutionary algorithm for this purpose. In [29] and [31], some non-standard MOEAs are used for the rule mining problem. NSGA-II has been used in [30] for ARM.

2) Chromosome Representation: There are mainly two chromosome representation techniques for categorical ARM, similar to the ones available for classification rule mining [1]. In the first approach (Pittsburgh approach), a set of possible association rules are encoded in each chromosome. This approach is more suitable for classification rule mining, where the objective is to identify a *good set of rules*. However, in ARM, the objective is to find a *set of good rules*. Therefore, for this case, the Michigan approach, in which each chromosome represents exactly one rule, is more suitable [27]. Most of the MOEAs-based categorical ARM techniques use this chromosome representation. In an early work [27], the authors adopted the Michigan approach as follows: each

TABLE I
COMPARISON OF DIFFERENT MOEAS FOR CLUSTERING

Algorithm	Underlying MOO tool	Data Type	Encoding	Objective functions	Evolutionary operators	Final solution from non-dominated front
Handl and Knowles [7], 2004 (VIENNA)	PESA-II	Continuous	Integer (Label-based)	$Dev(C)$, $Conn(C)$	No crossover, neighborhood-biased mutation	Independent objective-based (F-measure)
Liu et. al. [8], 2005 (MOKGA)	NPGA	Continuous	Integer (Label-based)	$TWCV$, number of clusters K	One-point crossover, probability-based replacement mutation	Independent objective-based ($Dunn$, DB, Silhouette, C, SD, S_Dbw indices)
Chen and Wang [9], 2005 (MOEA(Dynamic))	NSGA-II	Continuous	Real-valued (Centroid-based)	$Dev(C)$, $Conn(C)$	Two-point crossover, centroid perturbation mutation (Gaussian mutation)	Independent objective-based (F-measure)
Handl and Knowles [10], 2007 (MOCK)	PESA-II	Continuous	Integer (Adjacency graph-based)	$Dev(C)$, $Conn(C)$	Uniform crossover, neighborhood-biased mutation	Knee-based (with null model)
Ripon et. al. [11], 2006 (VRJGGA)	NSGA-II	Continuous	Real-valued (Centroid-based)	Entropy H, separation $Sep(C)$	Jumping gene crossover, centroid perturbation mutation (polynomial mutation)	Independent objective-based (Deviation and $Dunn$ index)
Bandyopadhyay et. al. [12], 2007 (MOGA)	NSGA-II	Continuous	Real-valued (Centroid-based)	J_m, XB	One-point crossover, centroid perturbation mutation (uniform distribution)	Independent objective-based ($\mathcal{I}$ index)
Qian et. al. [13], 2008 (MECEA)	PESA-II	Continuous	Integer (Adjacency graph-based)	$Dev(C)$, $Conn(C)$	Uniform crossover, neighborhood-biased mutation	Ensemble-based (Graph-based - MCLA)
Won et. al. [14], 2008 (MOES(Hybrid))	NSGA-II	Continuous	Real-valued (Centroid-based) (variable-length)	$TWCV$, number of clusters K	Centroid pool crossover, centroid perturbation mutation (log normal distribution)	None
Mukhopadhyay et. al. [15], [16], 2009 (MOGA-SVM)	NSGA-II	Continuous	Real-valued (Centroid-based)	J_m, XB	One-point crossover, centroid perturbation mutation (uniform distribution)	Ensemble-based (Majority vote and SVM classifier)
Shirakawa and Nagao [17] 2009 (MOCK variant)	SPEA2	Continuous	Integer (Adjacency graph-based)	$Dev(C)$, $Edge(C)$	Uniform crossover, neighborhood-biased mutation	Knee-based (without null model)
Ripon and Siddique [18], 2009 (EMCOC)	NSGA-II	Continuous	Binary (Medoid-based)	Entropy H, separation $Sep(C)$	Jumping gene crossover, no mutation	Independent objective-based (Entropy and Seprataion)
Mukhopadhyay and Maulik [19], 2011 (MOVGA)	NSGA-II	Continuous	Real-valued (Centroid-based) (variable-length)	Normalized J_m fuzzy separation $\mathcal{S}$	One-point crossover centroid perturbation mutation (uniform distribution)	Independent objective-based ($\mathcal{I}$ index)
Kirkland et. al. [20], 2011 (MOCA)	NSGA-II	Continuous	Real-valued (Centroid-based) (variable-length)	Average deviation, $ABGSS$, $Conn(C)$	Exchange corresponding prototypes crossover, centroid pool mutation (add/delete/modify centroid)	Independent objective-based (Rand index)
Handl and Knowles [21], 2005 (MOCK-am)	PESA-II	Categorical/ distance matrix	Integer (Adjacency graph-based)	$Dev(C)$, $Conn(C)$	Uniform crossover, neighborhood-biased mutation	Knee-based (with null model)
Mukhopadhyay and Maulik [22], 2007, MOGA(medoid)	NSGA-II	Categorical	Integer (Medoid-based)	$Dev(C)$, silhouette	One-point crossover, medoid replacement (point index)	Independent objective-based (Minkowski score)
Demir et. al. [23], 2007 (GraSC)	SPEA2	Categorical/ distance matrix	Integer (Label-based)	Min-Max cut, silhouette	Modified uniform crossover, random replacement mutation	Non-domination status
Mukhopadhyay et. al. [24], 2009 (MOGA(mode))	NSGA-II	Categorical	Categorical (Mode-based)	Normalized J_m fuzzy separation $\mathcal{S}$	One-point crossover, mode replacement mutation (categorical value replacement)	Ensemble-based (Majority vote and k-nn classifier)
Kim et. al. [25], 2010 (AI-NSGA-II)	NSGA-II	Graph	Integer (Adjacency graph-based)	Entropy H, separation $Sep(C)$	Uniform crossover, neighborhood-biased mutation	Non-domination status
Folino and Pizzuti [26], 2010 (DYN-MOGA)	NSGA-II	Graph	Integer (Adjacency graph-based)	$CS(C)$, NMI	Uniform crossover, neighborhood-biased mutation	Independent objective-based (Modularity)

chromosome had length $2k$, where k was the number of items. The chromosomes were binary strings where each attribute was given two bits. If these two bits are 00 or 11, then the attribute appears in the antecedent or consequent parts of the rule, respectively; otherwise, the attribute is absent from the rule. In a similar approach [28], the presence of an attribute in the antecedent and consequent part are represented by bits 10 and 01, whereas other bit combinations represent the absence of the attribute from the rule.

The above encoding schemes [27], [28] can only be adopted for binary datasets, i.e., when an item is either present or absent in a transaction. If someone wants to use this encoding for more general categorical data, where an item may be present in a transaction with certain value (a categorical state), the dataset will first need to be transformed into a binary one by considering each attribute-value pair as an item. In view of this, an alternative encoding strategy is presented in [30], which can be used for a categorical dataset directly. Here, each attribute has two parts. The first part represents the position of the attribute in the rule, and the second part represents the categorical value it takes. The first part contains two bits and the attribute appears in the antecedent and the consequent of the rule if the bits are 10 and 11, respectively; otherwise, it is absent from the rule. The second part represents categorical values taken by attributes in binary form. However, the authors did not explain how a binary value in the second part represents a categorical state if the number of states for an attribute is not an exact power of 2.

The main disadvantage of using a binary encoding scheme is that it gives rise to a large chromosome length when the number of attributes is large, since at least two bits are needed for each attribute. An integer encoding may come handy in this

respect. Such an integer encoding scheme has been proposed in ARMMGA (Association Rule Mining using Multi-objective Genetic Algorithm) [31], where the chromosomes encode the index of the attributes. A chromosome encoding a k-rule, k being the total number of items in the antecedent and the consequent, has $k + 1$ genes. The first gene position indicates the separating position of the chromosome where the antecedent and the consequent attributes are separated. For example, if A_i represents the ith item, then the chromosome $\{3 \mid 2\ 5\ 4\ 1\ 3\}$ represents the rule $A_2A_5A_4 \Rightarrow A_1A_3$. This representation significantly reduces the length of the chromosome, but not effectively the search space, because now for each position, a large number of alternative indices are to be searched. Moreover, this representation scheme gives rise to a variable chromosome length, thus requiring a specialized crossover operator. Also, there remains a possibility of finding duplicate indices in a chromosome after crossover/mutation, which must be taken care of during the evolutionary process.

3) Objective Functions: Although support and confidence are two popular objectives which are to be maximized, there are several other metrics to measure the interestingness of association rules. These metrics, which have been used by different algorithms for optimization in a multi-objective framework, include coverage, lift, comprehensibility, cosine, prevalence, recall, Laplace, conviction, surprise, Jaccard, J-measure, etc. [100]. In [27], the rule mining problem has been modeled as a three-objective optimization problem where confidence, comprehensibility, and interestingness have been optimized simultaneously. They defined the comprehensibility of a rule as $log(1 + |C|)/log(1 + |A \cup C|)$, where $|C|$ and $|A \cup C|$ denote the number of attributes in the consequent part and total rule, respectively. They considered that the lower value of comprehensibility, i.e., less number of attributes in the consequent of the rule, leads to better understandability of the rule. The interestingness measure, on the other hand, is defined as a product of three probabilities, viz., the probability of generating the rule given the antecedent (ratio of the support of the rule to the support of the antecedent), the probability of generating the rule given the consequent (ratio of the support of the rule to the support of the consequent), and the probability of generating the rule given both antecedent and consequent (ratio of the support of the rule to the total number of transactions). A rule becomes more interesting if it has a high interestingness value. In [28], two objective functions, statistical correlation and comprehensibility, have been simultaneously optimized in a co-evolutionary framework. The statistical correlation measure indicates a better association of the rule. In [29], five objective functions, viz., support, confidence, J-measure, interest and surprise [100] have been simultaneously optimized. They found five different groups of correlated measures. To make the objective functions contradictory and uncorrelated, they selected these five measures from five different groups. In [30], six different measures (support, confidence, interest, comprehensibility, cosine and attribute frequency) have been considered. Three of these measures have been taken at a time and optimized simultaneously. Measures such as support, confidence, interest and comprehensibility tend to be better if the rule-length is smaller. To counter this bias, they also maximized attribute frequency, which is the ratio of the rule-length to the total number of items. The reason behind taking three objective functions at a time is that NSGA-II, the underlying MOEA, is known to perform well when the number of objective functions is at most three. Secondly, due to correlation of the measures, it is unnecessary to use correlated measures for optimization. In [31], the classical measures (support and confidence of the rules) are simultaneously optimized. Thus, it is apparent from the above discussion that different sets of rule-interestingness measures have been chosen by various authors as their objective functions. However, a systematic comparison among the chosen objective functions is still missing in the literature.

4) Evolutionary Operators: When binary encoding has been adopted, standard crossover and mutation operators have been used. For example, in [27], multi-point crossover and bit-flip mutation have been used. In [30] bit-flip mutation has been adopted, however, the authors did not specifically mention which crossover operator is used. In [28], the authors proposed Pareto neighborhood crossover, a combination operator and an annexing operator. However, the way in which these operators work is not explained. There is no mention about the motivation for defining these operators, as well as no experimental results have been provided showing their improved effectiveness with respect to the standard operators.

In the encoding strategies, where along with the attributes, their values are also encoded, other types of evolutionary operators are needed. In [29], although the authors did not explain the encoding strategy explicitly, from the description of the evolutionary operators, it appears that they used an approach in which the categorical values of the attributes participating in the rule are encoded. Here, the authors used value exchange and insertion crossover operators. If two parents have some common attributes in the antecedent part, then a value exchange crossover is performed by exchanging the categorical values of one of the common attributes. When the parents do not have any common attribute, then one random attribute selected from one parent is inserted into the other with a probability that is inversely proportional to the length of the later chromosome. Four mutation operators were applied with equal probabilities. A value mutation randomly replaces a chosen categorical value with another random value from the same domain. An attribute mutation randomly replaces an attribute with another one. An insertion mutation inserts a new attribute-value pair, and a deletion mutation deletes a randomly chosen attribute-value pair. In [31], where integer encoding of the attributes is used, an order-1 crossover strategy is adopted. In this strategy, first, one segment is chosen equally from two chromosomes and, respectively, are copied from the first and second parents to the first and second offspring. Next, starting from the right side of the segment, the values of the genes that do not exist in the selected segment of the first parent, are copied to the first offspring. The same procedure is repeated for the second offspring as well. The mutation operator replaces a chosen item from the chromosome with a random item not present in the chromosome.

5) Obtaining a Final Solution: All the works for categorical rule mining using MOEAs that have been discussed in this

paper, use a Michigan type of encoding, where each chromosome encodes one association rule. Hence, the final generation produces a set of non-dominated solutions each of which are given to the user as the association rules generated from the input dataset. Thus, in this case, there is no specific need of selecting a single solution from the non-dominated front.

B. Numeric Association Rules

For datasets having continuous attribute domains, the ARM algorithms designed for categorical attributes do not work well. This is because such algorithms need categorization of the continuous attributes. Hence the results of the ARM algorithms depend a lot on the categorization technique adopted. To overcome this limitation, many numeric/quantitative ARM algorithms have been proposed and some of them adopted a multi-objective optimization approach.

A quantitative association rule is represented as [32], [33]:

$$(l_1 \leq A_1 \leq h_1) \ \wedge \ (l_2 \leq A_2 \leq h_2) \Rightarrow (l_3 \leq A_3 \leq h_3).$$

Here A_i represents the ith attribute. l_i and h_i represent the lower and upper bound of the attribute values. Thus, $[l_i, h_i]$ defines an interval of values for the attribute A_i. Here, we discuss two different works on quantitative ARM.

1) Underlying MOEAs: In this section, we review two multi-objective numerical/quantitative rule mining algorithms. The first is a multi-objective differential evolution based numeric association rule mining algorithm (MODENAR) [32]. In this case, a multi-objective differential evolution (MODE) algorithm is used as the underlying optimization framework. In another work, an NSGA-II-based quantitative association rule mining algorithm (NSGA-II-QAR) is proposed [33].

2) Chromosome Representation: The chromosomes representing numeric or quantitative association rules need to encode the lower and upper bounds of the intervals of the attributes participating in a rule. In [32], where the MODENAR algorithm has been proposed, the following encoding technique has been adopted for the chromosomes. They used chromosomes where each attribute has three components. The first component indicates whether the attribute is present or absent in the rule, and if present, in which part (antecedent or consequent) in the rule it is. The second and third components indicate the lower and upper bounds of the ranges of the attribute. The first component can have integer values 0, 1 or 2, which indicate the presence of the attribute in the antecedent of the rule, the presence of the attribute in the consequent of the rule, and the absence of the attribute from the rule, respectively. The second and third components can take real values from the corresponding attribute ranges. It is to be noted that as MODENAR uses differential evolution as an optimizer and works on real-valued chromosomes, the authors used a round-off operator to handle the integer part of the chromosome. A similar encoding scheme is adopted in NSGA-II-QAR. The only difference is that in this case, the first part of the chromosome, instead of using the values 0, 1, 2, adopts the values 0, 1 and -1, respectively, to denote the same meaning. In both cases, the algorithms used a Michigan encoding strategy, i.e., each chromosome encodes one rule.

3) Objective Functions: MODENAR optimizes four criteria of the rules [32]: support, confidence, comprehensibility and amplitude of the intervals that make up the itemset and the rule. Comprehensibility is used to bias the search process toward shorter rules, under the assumption that shorter rules provide more non-redundant information. They also proposed that the amplitude of the intervals must be smaller for interesting rules, but the rationale for this is not explained. In NSGA-II-QAR [33], three objective functions are simultaneously optimized: lift, comprehensibility and performance. Performance is defined by the product of confidence and support. Lift is defined as the ratio of support of the rule to the product of the supports of the antecedent and the consequent of the rule [101]. A high value for the lift measure indicates that the rule is interesting, since its support is high with respect to the supports of its antecedent and its confidence. The comprehensibility is defined simply as the reciprocal of the number of attributes in the rule. In [33], an experimental comparison between NSGA-II-QAR and MODENAR is provided.

4) Evolutionary Operators: MODENAR [32] used the standard version of the crossover and mutation operators adopted by the version of differential evolution called DE/rand/1. Additionally, a rounding operator is used to round-off the first part of the attribute which requires an integer (0, 1, 2) for computing the objective function values. In NSGA-II-QAR [33], a multi-point crossover is utilized. The two parts of the chromosome undergoes two different mutations. In the first part, where the chromosome can have a value of -1, 0 or 1, a random value is selected from the set {-1, 0, 1} and it replaces the existing value. The other part of the chromosome encodes the lower and upper bound of the chromosome. A mutation in applied to this part by increasing or decreasing these values randomly. In both of these studies [32], [33], during mutation/crossover, it may happen that the lower bound becomes larger than the upper bound, or they go outside the bounds. For this, some repairing operators are also adopted to make the chromosome a valid one.

5) Obtaining a Final Solution: Both MODENAR and NSGA-II-QAR used a Michigan approach of rule mining by encoding one rule in one chromosome. Thus, the final non-dominated set gives a set of numeric rules. Thus, there is no need for any particular solution from the final non-dominated set. All the solutions will serve as the final selected rule set.

C. Fuzzy Association Rules

One of the major problems of mining numeric association rules is that these algorithms deal with sharp boundaries between consecutive intervals. Thus, they cannot represent smooth changes from one interval to another, which can be easily handled by fuzzy association rules. A number of MOEA-based fuzzy ARM techniques have been developed in the past decade. Here, we describe several of these algorithms and discuss different approaches that incorporate them.

The general form of a fuzzy association rule is as [34]:

$$If \ X = \{x_1, x_2, \ldots, x_p\} \ is \ A = \{f_1, f_2, \ldots, f_p\}$$

$$Then \ Y = \{y_1, y_2, \ldots, y_q\} \ is \ B = \{g_1, g_2, \ldots, g_q\}.$$

Here X and Y represent two sets of attributes and $X \cap Y = \phi$. A and B represent the fuzzy sets (linguistic values) of the corresponding attributes in X and Y, respectively. Therefore, if a rule is encoded in a chromosome, both the attributes and their linguistic values should be encoded in it. A number of studies have been done on the application of MOEAs for fuzzy association rule mining. Here we review some of them.

1) Underlying MOEAs: Different MOEAs have been employed in various works on fuzzy ARM. In [34], [35], [36], Kaya et al. used a variant of SPEA for fuzzy rule mining. In [37], a multi-objective GA (MOGA) is used for this purpose. In another work on fuzzy association rule mining, NSGA-II has been employed [38]. However, in none of these studies, relative comparison among different MOEAs for fuzzy rule mining has been addressed.

2) Chromosome Representation: There are two categories of chromosome representations for fuzzy ARM. In the first approach, a chromosome represents a set of fuzzy clusters corresponding to each attribute. The objective is to find a suitable set of fuzzy clusters that partition the range of values in each attribute domain. This approach is adopted in a series of works done by Kaya et al. in [34], [35], [36]. In these works, each chromosome represents the base values of the variable number of membership functions representing the fuzzy sets for each quantitative attribute. Standard triangular membership functions are used to represent the fuzzy sets. Real-valued representation of the chromosomes is used for this purpose. Here a chromosome does not represent association rules. It represents a suitable fuzzy clustering of the attribute domains. The evolved fuzzy membership functions are then used as the linguistic values of the corresponding attributes. Fuzzy association rules are mined using standard algorithms based on minimum support and minimum confidence criteria. A similar encoding approach is adopted in [37].

The second approach directly encodes fuzzy association rules in the chromosomes. This is a kind of Michigan approach where each chromosome encodes a possible rule. In [38], such an encoding is adopted to mine temporal fuzzy association rules. Here, the authors used a mixed representation of chromosomes combining integer and real values. The chromosome encodes the lower and upper bounds of the temporal interval in the rules as integers. The indices of the items participating in the rule are also encoded as integers. Finally, the real-valued parameters of the triangular membership functions corresponding to each item are encoded in the chromosome. Thus, this representation induces variable-length chromosomes needing special evolutionary operators.

3) Objective Functions: In the works of Kaya et al. [34], [35], [36], the authors optimize two criteria, viz., number of large itemsets and time spent to obtain the large itemsets. Thus, here the objective is to evolve a possible fuzzy clustering of the numeric attributes that maximizes the number of large itemsets while minimizing the time required to obtain all large itemsets given the clustering. After optimizing the clustering, the authors then used the membership functions as the linguistic values for the fuzzy association rules extracted based on minimum support and minimum confidence criteria.

In [37], where a similar encoding strategy is adopted as in [34], two objective functions are optimized simultaneously. The first objective function is stability of the encoded membership functions, which has two components, viz., overlap factor and coverage factor. The stability is optimized to avoid generation of too redundant and too separated fuzzy sets for an item. The second objective is to maximize the total number of large 1-itemsets for given minimum support values. Although this work is a consequence of the works of Kaya et al. with modifications in the objective functions and evolutionary operators (described later), the authors did not compare their results with those of Kaya et al. So, it is difficult to judge any improvement of the performance over the previous approaches.

In [38], the authors used a direct approach to temporal fuzzy association rule mining by adopting the Michigan form of chromosomes. Thus, here the objective functions are related to the optimization of the encoded rules. In this work, four objective functions, namely temporal support, temporal confidence, fuzzy support and membership function widths, are optimized. Whereas the first three objective functions are obvious, the last objective function is used to prevent a membership function from covering the whole range of attribute values. Without this objective function, the solutions could evolve to cover the complete range of attribute values, since this gives higher support values as it includes more number of items.

4) Evolutionary Operators: In the works of Kaya et al. [34], [35], the authors have used standard multi-point crossover operations. In [36], the authors used arithmetic crossover. Also, they employed standard real-value mutation. In [37], the authors used Max-min arithmetical crossover and one-point mutation. This crossover operator generates four offspring at a time out of which the two best offspring are chosen. However, the authors did not describe the crossover process in detail, and did not discuss its advantage over a standard crossover operator. The mutation operator is used to slightly change the center of the fuzzy set being mutated. It is to be noted that when mutation takes place at the center of a fuzzy membership function, it may disrupt the order of the resulting fuzzy membership functions. Hence, these fuzzy membership functions need rearrangement according to their center values after the mutation. In [38], for a Michigan type of encoding, a modified uniform crossover operator is adopted. For mutating the genes representing the lower and upper bounds of the time interval, the values are generated within the endpoint range (epr) where the midpoint is the value of the current gene (g), such that the mutated value is a member of the set $\{-epr/2, \ldots, g, \ldots, epr/2\}$. This is done to reduce the effect of random sampling of the dataset.

5) Obtaining a Final Solution: As in [34], [35], [36], [37], a chromosome encodes a possible fuzzy clustering of the attribute values, it is necessary to select a suitable solution from the final non-dominated set, based on which of the final association rules are extracted. However, in [34], [35], [37], this issue has been overlooked. In [36], the authors presented an approach based on the lower bound of the objective function values to identify interesting solutions. The authors first determined a lower bound for an objective such that the values under the located lower bound are infeasible solutions for us. The lower bounds are the parameters that

are varied by the optimizer to obtain multiple non-dominated sets of solutions. Subsequently, the solution, which maximizes the other objective in feasible space is chosen. However, as the author mentioned, the lower bound must be chosen carefully and it is not a trivial task. In [38], the authors used a Michigan type of encoding of temporal fuzzy association rules. Therefore, all the rules encoded in the final non-dominated set are considered as extracted rules. There is no specific need of choosing any particular solution from the non-dominated set.

D. Relative Comparison and Applications

In Table II, we provide a comparative overview of different approaches for MOEA-based association rule mining. The approaches are categorized in three types as discussed, viz., categorical rule mining, numeric rule mining and fuzzy rule mining. Different methods are compared with respect to the underlying MOO tool, encoding strategy, objective functions, evolutionary operators and method for obtaining the final solution from the non-dominated set. It is evident from the table that most of the methods have used a Michigan encoding and thus all the non-dominated solutions are treated as final solutions without needing a particular solution from the set. Although a number of different methods have been discussed here, very few comparative studies of these methods are available in the literature. Only in [33], two numeric rule mining approaches using a MOEA, namely MODENAR and NSGA-II-QAR, have been compared in terms of different rule-interestingness metrics. However, in all the other works, the authors have concentrated on comparing the performance of their approaches with respect to existing single-objective evolutionary and other non-evolutionary methods.

Although MOEA-based ARM algorithms have gained popularity in recent years, their use in real-life applications is still fairly limited. The authors have mainly preferred to demonstrate their methods on some UCI repository datasets. It would interesting, however, to see applications of these techniques in domains such as mining gene expression and other biological data, financial databases and text mining.

IV. MOEAs for Other Data Mining Tasks

Most of MOEA-based data mining techniques have considered the four areas (feature selection, classification, clustering and association rule mining) as discussed before (in Part-I [1] and this part of the paper). However, besides these, MOEAs have also been applied for many other data mining tasks. These tasks include ensemble learning, biclustering/co-clustering, etc. In this section, we discuss some of the MOEA-based approaches that have been applied in these areas.

A. MOEAs for Ensemble Learning

Ensemble learning refers to the task of combining the predictions of individual classifiers in some way to obtain more robust predictions. The inherent strength of MOEAs to produce a set of trade-off classifiers in the form of a non-dominated set has made them popular in designing ensembles of classifiers. The general idea is to use MOEAs to yield a set of diverse classifiers encoded in the chromosomes of the final non-dominated front, and then make the final prediction by combining the predictions of these individual classifiers through a majority vote. Integration of diverse classifiers through ensemble learning may prevent overfitting and may provide better classification accuracy and improved robustness compared to the predictions based on a single classifier [102].

The general framework for ensemble classification design is to produce a diverse set of classifiers by optimizing certain contradictory criteria. A popular approach in this regard is to optimize an artificial neural network-based classifier or MLP with respect to the complexity of the classifier and its predictive accuracy. The complexity of an MLP refers to the number of hidden layer units and weights of the connections. This approach has been adopted in [39], [40], [41], [42], [43]. In [39], a Pareto-frontier differential evolution (PDE) algorithm [103] is used to develop the Memetic Pareto Artificial Neural Network (MPANN) method. In MPANN, the authors performed a comparison between two multi-objective formulations to the formation of neuro-ensembles. In the first formulation, the training set is split into two non-overlapping stratified subsets. The objectives are to minimize the training error on each subset. In the second formulation, they add random noise to the training set to form a second objective. They also compared there algorithm with a negative correlation learning (NCL) algorithm for training an ensemble of ANNs using Backpropagation [104]. In [40], the problem of regularization of neural network classifiers is addressed and as a bi-product, a neural network ensemble is generated. They compare the use of NSGA-II and a dynamic weighted aggregation method in generating the ensemble by optimizing two objectives, viz., training mean squared error and number of network connections. A similar approach for the generation of an ensemble of MLPs is found in [41] with different objective functions to be optimized. Here the authors minimized Type-I and Type-II errors simultaneously, which refer to the number of false positives and number of false negatives, respectively. The algorithm, called MG-Prop, is designed based on the single front genetic algorithm (SFGA) proposed in [105]. The authors showed that this ensemble works well for class-imbalanced data. In [42], an algorithm called DIVACE (DIVerse and Accurate Ensemble Learning Algorithm) is proposed. DIVACE uses ideas from NCL [104] and MPANN [39], and formulates the ensemble learning problem as a multi-objective problem explicitly within an evolutionary setup. The aim of the algorithm is to find good a trade-off between diversity and accuracy to produce an ensemble of neural network classifiers. The diversity is modeled as a correlation penalty [104]. The authors showed that DIVACE performs better than the MPANN algorithm. In a recent work, a Multi-objective genetic algorithm based Artificial Neural Network Ensemble (MANNE) method is proposed in [43] for intrusion detection. The authors optimized neural network classifiers using NSGA-II with two objective functions, namely detection rate and false positive rate to generate the ensemble. The method was compared with a decision tree and its ensembles using bagging and boosting methods.

Another popular approach for building MOEAs-based clas-

TABLE II
COMPARISON OF DIFFERENT MOEAS FOR ASSOCIATION RULE MINING

Algorithm	Underlying MOO tool	Type	Encoding	Objective functions	Evolutionary operators	Final solution from non-dominated front
Ghosh and Nath [27], 2004	MOGA	Categorical	Binary (Michigan)	Confidence, comprehensibility, interestingness	Multi-point crossover, bit-flip mutation	None
Hu and Yang-LI [28], 2007	Pareto-based coevolutionary	Categorical	Binary (Michigan)	Statistical correlation, comprehensibility	Pareto neighborhood crossover, combination, annexing	None
Khabzaoui et. al. [29], 2008	Non-standard	Categorical	Not mentioned	Support, confidence, J-measure, interest, surprise	Value exchange crossover, insertion crossover, value/attribute mutation, insertion/deletion mutation	None
Anand et. al. [30], 2009	NSGA-II	Categorical	Binary (Michigan)	Combination (3 at a time) of support, confidence, interest, comprehensibility, cosine, attribute frequency	Crossover not mentioned, bit-flip mutation	None
Qodmanan et. al. [31], 2011 (ARMMGA)	Non-standard	Categorical	Integer (Michigan)	Support, confidence	Order-1 crossover, random replacement mutation	None
Alatas et. al. [32], 2008 (MODENAR)	MODE	Numeric	Mixed (Integer + real) (Michigan)	Support, confidence, comprehensibility amplitude of interval	DE/Rand/1	None
Martin et. al. [33], 2011 (NSGA-II-QAR)	NSGA-II	Numeric	Real-valued (Michigan)	Lift, comprehensibility, performance (support $\times$ confidence)	Multi-point crossover, random increase/decrease mutation, random replacement mutation	None
Kaya and Alhajj [34], 2003 Kaya and Alhajj [35], 2004	SPEA variant	Fuzzy	Real-valued (membership functions)	Number of large itemsets, time taken to find all large itemsets	Multi-point crossover, standard real-value mutation	Not mentioned
Alhajj and Kaya [36], 2008	SPEA variant	Fuzzy	Real-valued (membership functions)	Number of large itemsets, time taken to find all large itemsets	Multi-point crossover, standard real-value mutation	Lower-bound based
Chen et. al. [37], 2008	MOGA	Fuzzy	Real-valued (membership functions)	Number of large 1-itemsets, suitability of membership functions	Max-min arithmetic crossover, one-point mutation	None
Matthews et. al. [38], 2011	NSGA-II	Fuzzy (temporal)	Mixed (integer + real) (Michigan)	Temporal support, temporal confidence, fuzzy support, membership function width	Modified uniform crossover, random change mutation	None

sifier ensembles is to encode a feature subset and other parameters in a chromosome and use some classifier as a wrapper to compute the objective functions (usually classification accuracy and feature subset size). The idea is to evolve a set of non-dominated classifiers with respect to the trade-off between accuracy and feature subset size. Each of them works on a specific subspace of the dataset and can be used to form an ensemble of classifiers [106], [107], [108]. In [106], Oliveira et al. used MLP as the wrapper and the classification accuracy and feature subset size as the two objective functions to be optimized. In [107], the authors considered both the supervised and the unsupervised cases. For the supervised case, they have used MLP as the wrapper and as objective functions the same defined in [106]. In the unsupervised case, they have used the K-means clustering algorithm and used DB index and number of features as the objective functions. Experimental studies established performance improvement compared to classical bagging and boosting techniques. In [108], on the other hand, three classifiers have been used as wrappers, namely decision tree, SVM and MLP. Two objective functions used are average accuracy of these three classifiers and consensus accuracy of them. The authors demonstrated that the proposed method outperforms single objective GA-based methods designed with one of these classifiers as wrapper.

B. MOEAs for Biclustering

A variant of the clustering, called biclustering or co-clustering [109], aims to capture local structures within a dataset. A clustering algorithm groups similar objects where the similarity is computed based on all attributes. On the contrary, the goal of a biclustering algorithm is to find a group of objects that are not necessarily similar over all the attributes, but are similar based on a subset of attributes. Hence, biclustering can be thought of as the simultaneous clustering of objects and attributes. Biclustering algorithms have several applications in different real-life domains such as text mining [110], recommender systems [111] and collaborative filtering [112]. However, almost all the MOEAs for biclustering are applied for mining biclusters from microarray gene expression data [109]. Here, we review some of these algorithms.

As the biclustering problem requires several objectives to be optimized such as mean squared residue (MSR) (a coherence measure) [113], volume, row variance, etc., this problem can be posed as a multi-objective optimization problem in a straightforward manner. In recent years, a number of studies have been done in solving biclustering problems using MOEAs. In [44], a multi-objective GA-based biclustering technique is proposed. The authors use a binary string of length $\mathcal{G} + \mathcal{C}$, where $\mathcal{G}$ and $\mathcal{C}$ denote the number of genes and number of conditions/samples/time points, respectively. If a bit position is '1', then the corresponding gene or condition is selected in the bicluster and if a bit position is '0', the corresponding gene or condition is not selected in the bicluster. The algorithm optimizes the MSR and volume of the biclusters simultaneously, in order to obtain coherent and large biclusters. The algorithm uses NSGA-II as the underlying multi-objective optimization tool. Cheng and Church's biclustering algorithm [113] has been used as a local search strategy.

In [45], a different encoding policy is adopted. The algorithm is termed as Multi-Objective GA-based Biclustering (MOGAB). Here, each string has two parts: one for clustering the genes, and another for clustering the conditions. If M and N denote the maximum number of gene clusters and the maximum number of condition clusters,

respectively, then the length of each string is $M + N$. The first M positions represent the M cluster centers for the genes, and the remaining N positions represent the N cluster centers for the conditions. Thus a string, looks like follows: $\{gc_1 \ gc_2 \ \ldots \ gc_M \ cc_1 \ cc_2 \ \ldots \ cc_N\}$, where each gc_i, $i = 1 \ldots M$, represents the index of a gene that acts as a cluster center of a set of genes, and each cc_j, $j = 1 \ldots N$, represents the index of a condition that acts as a cluster center of a set of conditions. A string that encodes M gene clusters and N condition clusters, represents a set of $M \times N$ biclusters, taking each pair of gene and condition clusters. Each pair $< gc_i, cc_j >$, $i = 1 \ldots M, \ j = 1 \ldots N$, represents a bicluster that consists of all genes of the gene cluster centered at gene gc_i, and all conditions of the condition cluster centered at condition cc_j. During the fitness computation, the gene and condition clusters encoded in the chromosome are updated through K-means at each iteration. Two objectives, viz., $\frac{MSR(I,J)}{\delta}$ and $\frac{1}{1+VAR(I,J)}$ are optimized simultaneously. This approach also adopts NSGA-II for optimization. Single-point crossover and random replacement mutation have been used as the evolutionary operators. MOGAB also uses NSGA-II as its underlying optimization tool. In [46], a fuzzy version of MOGAB is proposed. Fuzzy versions of MSR and row variance have been simultaneously optimized.

In [47], the authors proposed the Sequential Multi-Objective Biclustering (SMOB) algorithm. They adopted binary encoding in this case. Three objective functions, viz., mean squared residue, volume and row variance were optimized. In [48], a hybrid multi-objective biclustering algorithm that combines NSGA-II and a Estimation of Distribution Algorithm (EDA) [114] for searching biclusters was proposed. The volume and MSR of the biclusters are simultaneously optimized. In [49], an NSGA-II based multi-objective biclustering algorithm was proposed. This approach uses integer encoding. Here, the integers represent the indices of the rows and the columns of the dataset. The objectives optimized are the similarity within the biclusters and the volume of the biclusters.

Although different biclustering approaches are proposed using MOEAs, there has been no effort to compare them systematically. MOEA-based biclustering algorithms have been compared with respect to standard single-objective evolutionary biclustering approaches as well as with respect to other non-evolutionary algorithms based on several criteria. However, comparative studies among different MOEA-based approaches are practically non-existent. These algorithms differ in their encoding strategies, objective functions, evolutionary operators and underlying MOEAs. Therefore, some studies to compare their performance would be beneficial for the users to select the most suitable method for their applications.

C. Other MOEAs-based Data Mining Approaches

There are a few additional areas of data mining where MOEAs have been applied, but they are not well-studied still now. One of such areas is feature extraction and construction. Feature extraction or construction refers to the task of creation of new features from functions of the original features. Feature selection can be considered as a special case of feature extraction. In [50], the problem of feature extraction for recognizing isolated handwritten symbols is posed as a multi-objective optimization problem, and a multi-objective genetic algorithm is proposed to solve the problem. The proposed algorithm has been shown to outperform human experts. However, not much progress has been noticed in this area, using MOEAs.

Subgroup discovery is another data mining problem where the aim is to mine fuzzy rules for subgroup discovery. These fuzzy rules help to represent the knowledge about patterns of interest which is explanatory and understandable to the expert. A few MOEAs-based approaches have been proposed for this purpose over the last few years [51], [52], [53]. The objective is to optimize different rule-interestingness criteria as in ARM.

MOEAs have also been used for regression. In [54], a MOEAs-based approach for obtaining linguistic fuzzy rule-based regression models from imprecise data is proposed. Here, each chromosome encodes one rule (Michigan approach), which competes with others in terms of maximum coverage and fitting. The knowledge base is formed through cooperation of individuals in the population. In a similar work [55], the authors proposed a multi-objective genetic fuzzy system (GFS) to learn the granularities of fuzzy partitions, for tuning the membership functions (MFs), and for learning the fuzzy rules for a regression problem. The proposed method uses dynamic constraints. This enables three-parameter membership function tuning for improved accuracy and guarantees the transparency of fuzzy partitions at the same time.

Another application of MOEAs has been found in outlier detection. In [56], a multi-objective genetic algorithm is proposed for outlier detection. The MOEA in this case is mainly employed as an effective search method in unsupervised learning for finding outlying subspaces from training data. Besides this, MOEAs have also been used in soft subspace clustering [57], [58]. However, MOEAs have been applied in these areas only very recently and much more work is still needed.

V. Future Directions

Although MOEAs are being applied in data mining tasks over the past decade and the literature is already quite rich, still some important future research issues remain open. Here, we discuss some relevant and important research topics to be addressed in the future. First of all, most of the studies have focused on comparing the proposed MOEA-based data mining techniques with existing non-evolutionary or traditional single-objective evolutionary techniques. However, as discussed before, practically none of the studies have compared the performance of different MOEA-based algorithms for different data mining tasks in a systematic way. Thus, for a novice user, it is difficult to judge which algorithm he/she should use for a particular task in hand. Possible reasons for unavailability of these studies may be the lack of publicly available softwares/codes, difficulty in reproducing the results, use of a variety encoding strategies, objective functions, evolutionary operators and final solution selection. Thus, a systematic comparison to guide new users to choose a suitable method for his/her application would be very valuable as it is still missing in the specialized literature.

In the majority of the studies on MOEA-based data mining, the performance of the algorithms has been reported based on the quality of the obtained result in terms of some metrics. However, to address large scale data mining problems using MOEAs, such as clustering large images or selecting genes from gene expression data containing several thousands of genes, along with the quality measure, the efficiency of the algorithm is also an important concern. Evolutionary algorithms have long been criticized for consuming large amounts of computational time as compared to other heuristics. Moreover, MOEAs typically require more computational time than single-objective evolutionary algorithms. Almost none of the MOEAs-based data mining studies reviewed here has considered to provide a systematic time complexity analysis. Therefore, it is difficult to compare different MOEAs in terms of time usage. Computational efficiency of MOEAs used in data mining is, indeed another promising research area. For example, one could incorporate local search strategies in MOEAs to improve the convergence rate. Many of the MOEAs-based data mining techniques currently available have already adopted this strategy, specially in clustering [12] and biclustering [44]. Another possibility is the efficient parallelization of MOEAs using multiple processors. A few studies in this regard have been done for ARM [115] and clustering [116], but more studies are needed to explore other application areas. Another way to reduce the search time for the MOEAs used in data mining is to use some appropriate stop criterion for them, instead of a fixed number of generations (as traditionally done). Some approaches are currently available for defining stop criteria for MOEAs (see for example [117]), but none of them have been adopted in data mining yet.

Most of the data mining problems have many objectives to be optimized. For example, a rule mining problem has objectives such as support, confidence, rule length, comprehensibility, interestingness, lift, etc., whereas a clustering algorithm may optimize a number of cluster validity measures simultaneously. However, few MOEA-based data mining problems have been posed with more than three objective functions [81]. Traditional MOEAs such as NSGA-II, SPEA2 and PAES are known to have difficulties to solve problems with 4 or more objectives, and other approaches are required to deal with them (see for example [82]). The use of such approaches in data mining is, however, still missing in the specialized literature.

Another research direction that deserves attention is interactive data mining using MOEAs. In interactive data mining, during the execution of a data mining algorithm, it interacts with a human decision maker to learn in a gradual way. This might be very useful to incorporate such interactions in MOEA-based data mining algorithms when some expert user is available for the problem at hand. Such an approach has been proposed in [84], where the authors have developed an interactive MOEA-based clustering approach called interactive multi-objective clustering (IMOC). In IMOC, the algorithm interacts with a human decision maker during its execution in order to learn the suitable set of cluster validity indices for the input dataset. Thus, different sets of validity measures may be chosen for different datasets. The method has been shown to perform well in clustering of gene expression data. Similar interactive MOEAs may be developed for other data mining tasks such as feature selection, classification and rule mining.

VI. Conclusions

In this two-part article, we have surveyed several MOEAs used for four primary data mining tasks namely feature selection and classification (in Part-I [1]), and clustering and association rule mining (in Part-II). The main focus has been on the chromosome representation, objective functions, evolutionary operators, and final solution selection from the non-dominated front. Moreover, a comparative overview among different methods in each category along with some real-life applications are provided. Additionally, in this Part-II of the paper, several other MOEAs-based data mining tasks, such as ensemble learning, biclustering, feature extraction, sub-group discovery etc. have been reviewed. Finally we have discussed a number of future research areas that deserve attention from the researchers working on the development of MOEAs-based data mining algorithms.

References

[1] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. Coello Coello, "A survey of multi-objective evolutionary algorithms for data mining: Part-I," *IEEE Trans. Evol. Comput.*, this issue.

[2] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. England: John Wiley and Sons, 2001.

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, 1999.

[4] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 1993, pp. 207–216.

[5] U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay, *Multiobjective Genetic Algorithms for Clustering - Applications in Data Mining and Bioinformatics*. Springer, 2011.

[6] U. Maulik and S. Bandyopadhyay, "Genetic algorithm based clustering technique," *Pattern Recognition*, vol. 33, pp. 1455–1465, 2000.

[7] J. Handl and J. D. Knowles, "Evolutionary multiobjective clustering," in *Proc. 8th Int. Conf. Parallel Problem Solving in Nature (PPSN-04)*, 2004, pp. 1081–1091.

[8] Y. Liu, T. zyer, R. Alhajj, and K. Barker, "Integrating multi-objective genetic algorithm and validity analysis for locating and ranking alternative clustering," *Informatica*, vol. 29, pp. 33–40, 2005.

[9] E. Chen and F. Wang, "Dynamic clustering using multi-objective evolutionary algorithm," in *Proceedings of the 2005 international conference on Computational Intelligence and Security - Volume Part I*, ser. CIS'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 73–80.

[10] J. Handl and J. D. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, 2007.

[11] 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. Int. Jt. Conf. Neural Networks*, 2006, pp. 3609–3616.

[12] 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, pp. 1506–1511, 2007.

[13] X. Qian, X. Zhang, L. Jiao, and W. Ma, "Unsupervised texture image segmentation using multiobjective evolutionary clustering ensemble algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2008)*, 2008, pp. 3561–3567.

[14] J.-M. Won, S. Ullah, and F. Karray, "Data clustering using multi-objective hybrid evolutionary algorithm," in *Proc. Int. Conf. Control, Automation and Systems*, 2008, pp. 2298–2303.

[15] A. Mukhopadhyay and U. Maulik, "Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with SVM classifier," *IEEE Trans on Geosc. and Remote Sensing*, vol. 47, no. 4, pp. 1132–1138, 2009.

[16] U. Maulik, A. Mukhopadhyay, and S. Bandyopadhyay, "Combining Pareto-optimal clusters using supervised learning for identifying co-expressed genes," *BMC Bioinformatics*, vol. 10, no. 27, 2009.

[17] S. Shirakawa and T. Nagao, "Evolutionary image segmentation based on multiobjective clustering," in *Proc. IEEE Cong. Evol. Comput. (CEC-09)*, 2009, pp. 2466–2473.

[18] K. S. N. Ripon and M. N. H. Siddique, "Evolutionary multi-objective clustering for overlapping clusters detection," in *Proc. IEEE Cong. Evol. Comput. (CEC-09)*, 2009, pp. 976–982.

[19] A. Mukhopadhyay and U. Maulik, "A multiobjective approach to MR brain image segmentation," *Applied Soft Computing*, vol. 11, pp. 872–880, January 2011.

[20] O. Kirkland, V. Rayward-Smith, and B. de la Iglesia, "A Novel Multi-Objective Genetic Algorithm for Clustering," in *Intelligent Data Engineering and Automated Learning - IDEAL 2011*, H. Yin, W. Wang, and V. Rayward-Smith, Eds. Berlin: Springer. Lecture Notes in Computer Science, Vol. 6936, 2011, pp. 317–326.

[21] J. Handl and J. D. Knowles, "Multiobjective clustering around medoids," in *Proc. IEEE Cong. Evol. Comput. (CEC-05)*, 2005, pp. 632–639.

[22] A. Mukhopadhyay and U. Maulik, "Multiobjective approach to categorical data clustering," *Proc. IEEE Cong. Evol. Comput. (CEC-07)*, pp. 1296–1303, 2007.

[23] G. N. Demir, A. S. Uyar, and S. G. Ögüdücü, "Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems," in *Proc. 9th Ann. Conf. Genetic and Evol. Comput. (GECCO-07)*, 2007, pp. 1943–1950.

[24] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Multi-objective genetic algorithm based fuzzy clustering of categorical attributes," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 991–1005, 2009.

[25] K. Kim, R. B. McKay, and B.-R. Moon, "Multiobjective evolutionary algorithms for dynamic social network clustering," in *Proc. 12th Ann. Conf. Genetic and Evol. Comput. (GECCO-10)*, 2010, pp. 1179–1186.

[26] F. Folino and C. Pizzuti, "A multiobjective and evolutionary clustering method for dynamic networks," in *International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2010)*. IEEE Computer Society, 2010, pp. 256–263.

[27] A. Ghosh and B. Nath, "Multi-objective rule mining using genetic algorithms," *Inf. Sci.*, vol. 163, no. 1-3, pp. 123–133, June 2004.

[28] J. Hu and X. Yang-Li, "Association Rules Mining Using Multi-objective Coevolutionary Algorithm," in *Proc. Int. Conf. Computational Intelligence and Security Workshops (CISW'07)*, 2007, pp. 405–408.

[29] M. Khabzaoui, C. Dhaenens, and E.-G. Talbi, "Combining evolutionary algorithms and exact approaches for multi-objective knowledge discovery," *RAIRO - Operations Research*, vol. 42, no. 1, pp. 69–83, 2008.

[30] R. Anand, A. Vaid, and P. K. Singh, "Association rule mining using multi-objective evolutionary algorithms: Strengths and challenges," in *NaBIC*, 2009, pp. 385–390.

[31] H. R. Qodmanan, M. Nasiri, and B. Minaei-Bidgoli, "Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 288–298, Jan 2011.

[32] B. Alatas, E. Akin, and A. Karci, "MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 646–656, Jan 2008.

[33] D. Martin, A. Rosete, J. Alcala-Fdez, and F. Herrera, "A multi-objective evolutionary algorithm for mining quantitative association rules," in *Proc. 11th Int. Conf. Intell. Sys. Des. and Appl. (ISDA-11)*, 2011, pp. 1397–1402.

[34] M. Kaya and R. Alhajj, "Facilitating fuzzy association rules mining by using multi-objective genetic algorithms for automated clustering," in *Proc. Third IEEE Int. Conf. Data Mining (ICDM-03)*. Washington, DC, USA: IEEE Computer Society, 2003.

[35] ——, "Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining," in *Proceedings of the Fourth IEEE International Conference on Data Mining*, ser. ICDM '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 431–434.

[36] R. Alhajj and M. Kaya, "Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining," *J. Intell. Inf. Syst.*, vol. 31, no. 3, pp. 243–264, 2008.

[37] C.-H. Chen, T.-P. Hong, V. S. Tseng, and L.-C. Chen, "A multi-objective genetic-fuzzy mining algorithm." in *Proc. IEEE Int. Conf. Granular Computing (GRC 2008)*, 2008, pp. 115–120.

[38] S. G. Matthews, M. A. Gongora, and A. A. Hopgood, "Evolving temporal fuzzy association rules from quantitative data with a multi-objective evolutionary algorithm," in *Proc. Int. Conf. Hybrid Artificial Intelligent systems - Volume Part I*, ser. HAIS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 198–205.

[39] H. A. Abbass, "Pareto Neuro-Evolution: Constructing Ensemble of Neural Networks Using Multi-objective Optimization," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 3. Canberra, Australia: IEEE Press, December 2003, pp. 2074–2080.

[40] Y. Jin, T. Okabe, and B. Sendhoff, "Neural network regularization and ensembling using multi-objective evolutionary algorithms," in *Proc. IEEE Cong. Evolutionary Computation (CEC04)*, 2004, pp. 1–8.

[41] P. A. Castillo, M. G. Arenas, J. J. Merelo, V. M. Rivas, and G. Romero, "Multiobjective optimization of ensembles of multilayer perceptrons for pattern classification," in *Proc. 9th Int. Conf. Parallel Prob. Solving from Nat. (PPSN-06)*, 2006, pp. 453–462.

[42] A. Chandra and X. Yao, "Ensemble learning using multi-objective evolutionary algorithms," *J. Math. Model. Algorithms*, vol. 5, no. 4, pp. 417–445, 2006.

[43] G. Kumar and K. Kumar, "The use of multi-objective genetic algorithm based approach to create ensemble of ann for intrusion detection," *Int. J. Intelligence Science*, vol. 2, no. 4, pp. 115–127, 2012.

[44] S. Mitra and H. Banka, "Multiobjective evolutionary biclustering of gene expression data," *Patt. Recogn.*, vol. 39, pp. 2464–2477, 2006.

[45] U. Maulik, A. Mukhopadhyay, and S. Bandyopadhyay, "Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm," *IEEE Trans. Inform. Tech. in Biomedicine*, vol. 13, no. 6, pp. 969–976, 2009.

[46] U. Maulik, A. Mukhopadhyay, S. Bandyopadhyay, M. Q. Zhang, and X. Zhang, "Multiobjective fuzzy biclustering in microarray data: Method and a new performance measure," in *Proc. IEEE World Cong. Computational Intelligence (WCCI 2008)/IEEE Cong. Evolutionary Computation (CEC 2008)*, Hong Kong, 2008, pp. 383–388.

[47] F. Divina and J. S. Aguilar-Ruiz, "A multi-objective approach to discover biclusters in microarray data," in *Proc. 9th Ann. Conf. Genetic and Evol. Comput. (GECCO-07)*, 2007, pp. 385–392.

[48] L. Fei and L. Juan, "Biclustering of gene expression data with a new hybrid multi-objective evolutionary algorithm of NSGA-II and EDA," in *Proc. Int. Conf. Bioinfo. and Biomed. Engg. (ICBBE-08)*, 2008.

[49] K. Seridi, L. Jourdan, and E.-G. Talbi, "Multi-objective evolutionary algorithm for biclustering in microarrays data," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, june 2011, pp. 2593 –2599.

[50] P. V. W. Radtke, T. Wong, and R. Sabourin, "A multi-objective memetic algorithm for intelligent feature extraction," in *Proceedings of the Third international conference on Evolutionary Multi-Criterion Optimization*, ser. EMO'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 767–781.

[51] F. Berlanga, M. Jesus, P. Gonzlez, F. Herrera, and M. Mesonero, "Multiobjective evolutionary induction of subgroup discovery fuzzy rules: A case study in marketing," in *Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, ser. Lecture Notes in Computer Science, P. Perner, Ed. Springer Berlin Heidelberg, 2006, vol. 4065, pp. 337–349.

[52] M. Del Jesus, P. Gonzalez, and F. Herrera, "Multiobjective genetic algorithm for extracting subgroup discovery fuzzy rules," in *Proc. IEEE Symp. Comput. Intell. Multicriteria Decision Making*, 2007, pp. 50–57.

[53] V. Pachón, J. Mata, J. L. Domínguez, and M. J. M. na, "A multi-objective evolutionary approach for subgroup discovery," in *Proceedings of the 6th international conference on Hybrid artificial intelligent systems - Volume Part II*, ser. HAIS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 271–278.

[54] L. Sanchez, J. Otero, and I. Couso, "Obtaining linguistic fuzzy rule-based regression models from imprecise data with multiobjective genetic algorithms," *Soft Computing*, vol. 13, no. 5, pp. 467–479, 2009.

[55] P. Pulkkinen and H. Koivisto, "A dynamically constrained multiobjective genetic fuzzy system for regression problems," *Fuzzy Systems, IEEE Transactions on*, vol. 18, no. 1, pp. 161–177, Feb. 2010.

[56] J. Zhang, Q. Gao, and H. Wang, "SPOT: A system for detecting projected outliers from high-dimensional data streams," in *Proc. IEEE 24th Int. Conf. Data Engg. (ICDE-08)*, 2008, pp. 1628–1631.

[57] L. Zhu, L. Cao, and J. Yang, "Multiobjective evolutionary algorithm-based soft subspace clustering," in *2012 IEEE Congress on Evolutionary Computation (CEC'2012)*. Brisbane, Australia: IEEE Press, June 10-15 2012, pp. 2732–2739.

[58] H. Xia, J. Zhuang, and D. Yu, "Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data," *Pattern Recogn.*, vol. 46, no. 9, pp. 2562–2575, Sep. 2013.

[59] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. Genetic and Evol. Comput. Conf. (GECCO-01)*, 2001, pp. 283–290.

[60] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[61] K. S. N. Ripon, C.-H. Tsang, S. Kwong, and M.-K. Ip, "Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm," in *Proc. Int. Conf. Pattern Recognition (ICPR 2006)*, 2006, pp. 1200–1203.

[62] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, 2002, pp. 95–100.

[63] J. Horn and N. Nafpliotis, "Multiobjective Optimization using the Niched Pareto Genetic Algorithm," University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, Tech. Rep. IlliGAl Report 93005, 1993.

[64] A. Mukhopadhyay, S. Bandyopadhyay, and U. Maulik, "Clustering using multi-objective genetic algorithm and its application to image segmentation," in *Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC-06)*, vol. 3, Taipei, Taiwan, October 2006, pp. 2678–2683.

[65] S. Bandyopadhyay, A. Mukhopadhyay, and U. Maulik, "An improved algorithm for clustering gene expression data," *Bioinformatics*, vol. 23, no. 21, pp. 2859–2865, 2007.

[66] G. N. Demir, A. S. Uyar, and S. G. Ögüdücü, "Multiobjective evolutionary clustering of web user sessions: a case study in web page recommendation," *Soft Comput.*, vol. 14, no. 6, pp. 579–597, 2010.

[67] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, 2002.

[68] J. Handl and J. Knowles, "Exploiting the trade-off – the benefits of multiple objectives in data clustering," in *Proc. Third Int. Conf. Evol. Multi-Criterion Optimization*, ser. EMO'05, 2005, pp. 547–560.

[69] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[70] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 13, pp. 841–847, 1991.

[71] T. Özyer, Y. Liu, R. Alhajj, and K. Barker, "Multi-objective genetic algorithm based clustering approach and its application to gene expression data," in *Proceedings of the Third international conference on Advances in Information Systems*, ser. ADVIS'04. Berlin, Heidelberg: Springer-Verlag, 2004, pp. 451–461.

[72] J. Du, E. E. Korkmaz, R. Alhajj, and K. Barker, "Alternative clustering by utilizing multi-objective genetic algorithm with linked-list based chromosome encoding." in *MLDM*, ser. Lecture Notes in Computer Science, vol. 3587. Springer, 2005, pp. 346–355.

[73] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comp. App. Math*, vol. 20, pp. 53–65, 1987.

[74] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Unsupervised cancer classification through SVM-boosted multiobjective fuzzy clustering with majority voting ensemble," in *Proc. IEEE Cong. Evol. Comput. (CEC-09)*, Norway, May 2009.

[75] ——, "Multiobjective genetic fuzzy clustering of categorical attributes," in *Proc. 10th International Conference on Information Technology (ICIT-07)*. Rourkela, India: IEEE Computer Society, 2007, pp. 74–79.

[76] S. Bandyopadhyay, R. Baragona, and U. Maulik, "Clustering multivariate time series by genetic multiobjective optimization," *Metron - Int. J. Statistics*, vol. LXVIII, no. 2, pp. 161–183, 2010.

[77] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 1, pp. 224–227, 1979.

[78] J. C. Dunn, "Well separated clusters and optimal fuzzy partitions," *J. Cyberns.*, vol. 4, pp. 95–104, 1974.

[79] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Gene expression data analysis using multiobjective clustering improved with SVM based ensemble," *In Silico Biology*, vol. 11, no. 1-2, pp. 19–27, 2011.

[80] ——, "Multiobjective genetic clustering with ensemble among pareto front solutions: Application to MRI brain image segmentation." in *Proc. Int. Conf. Adv. in Patt. Recogn. (ICAPR'09)*, 2009, pp. 236–239.

[81] T. Özyer, M. Zhang, and R. Alhajj, "Integrating multi-objective genetic algorithm based clustering and data partitioning for skyline computation," *Applied Intelligence*, vol. 35, no. 1, pp. 110–122, Aug. 2011.

[82] D. Saxena, J. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, 2013.

[83] J. Handl and J. Knowles, "Clustering criteria in multiobjective data clustering," in *Parallel Problem Solving from Nature - PPSN XII*, ser. Lecture Notes in Computer Science, C. A. Coello Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, Eds. Springer Berlin Heidelberg, 2012, vol. 7492, pp. 32–41.

[84] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "An interactive approach to multiobjective clustering of gene expression patterns," *IEEE Trans Biomed Eng*, 2012 (in press).

[85] K. Praditwong, M. Harman, and X. Yao, "Software module clustering as a multi-objective search problem," *IEEE Trans. Softw. Eng.*, vol. 37, no. 2, pp. 264–282, 2011.

[86] A. Mukhopadhyay, S. Bandyopadhyay, and U. Maulik, "Analysis of microarray data using multiobjective variable string length genetic fuzzy clustering," in *Proc. IEEE Cong. Evol. Comput. (CEC-09)*, 2009, pp. 1313–1319.

[87] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, "Quality scheme assessment in the clustering process," in *Proc. 4th European Conf. Principles of Data Mining and Knowledge Discovery*, ser. PKDD '00. London, UK, UK: Springer-Verlag, 2000, pp. 265–276.

[88] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intelligent Information Systems*, vol. 17, no. 2/3, pp. 107–145, 2001.

[89] J. Handl and J. D. Knowles, *Multiobjective Clustering and Cluster Validation*, ser. Comput. Intell. Springer, 2006, vol. 16, pp. 21–47.

[90] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[91] J. Handl and J. D. Knowles, "Improvements to the scalability of multiobjective clustering," in *Proc. Congress on Evolutionary Computation (IEEE CEC-05)*. IEEE, 2005, pp. 2372–2379.

[92] N. Matake and T. Hiroyasu, "Multiobjective Clustering with Automatic k-determination for Large-scale Data. Scalable automatic determination of the number of clusters," in *2007 Genetic and Evolutionary Computation Conference (GECCO'07)*, vol. 1, 2007, pp. 861–868.

[93] A. Strehl and J. Ghosh, "Cluster ensembles - A knowledge reuse framework for combining multiple partitions," in *Machine Learning Research*, vol. 3, 2002, pp. 583–617.

[94] J. Handl, J. D. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, no. 15, pp. 3201–3212, 2005.

[95] T. zyer, Y. Liu, R. Alhajj, and K. Barker, "Multi-objective genetic algorithm based clustering approach and its application to gene expression data," in *Advances in Information Systems*, ser. Lecture Notes in Computer Science, T. Yakhno, Ed. Springer Berlin / Heidelberg, 2005, vol. 3261, pp. 451–461.

[96] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Simultaneous informative gene selection and clustering through multiobjective optimization," in *IEEE Cong. on Evol. Comput. (CEC-10)*, 2010, pp. 1–8.

[97] A. Mukhopadhyay, S. Bandyopadhyay, and U. Maulik, "Multi-class clustering of cancer subtypes through svm based ensemble of pareto-optimal solutions for gene marker identification," *PloS ONE*, vol. 5, no. 11, p. e13803, 2010.

[98] A. Mukhopadhyay, S. Ray, and M. De, "Detecting protein complexes in a PPI network: a gene ontology based multi-objective evolutionary approach," *Mol Biosyst*, vol. 8, no. 11, pp. 3036–3048, Nov 2012.

[99] M. Jesús, J. A. Gámez, P. González, and J. M. Puerta, "On the discovery of association rules by means of evolutionary algorithms," *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, vol. 1, no. 5, pp. 397–415, 2011.

[100] P.-N. Tan, V. Kumar, and J. Srivastava, "Selecting the right interestingness measure for association patterns," in *Proc. eighth ACM SIGKDD Int. conf. Knowledge discovery and data mining (KDD-02)*, 2002, pp. 32–41.

[101] P. Lenca, P. Meyer, B. Vaillant, and S. Lallich, "On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid," *European Journal of Operational Research*, vol. 184, no. 2, pp. 610–626, 2008.

[102] J. Handl, D. B. Kell, and J. Knowles, "Multiobjective optimization in bioinformatics and computational biology," *IEEE/ACM Trans. Comput. Bio. and Bioinfo.*, vol. 4, no. 2, pp. 279–292, 2007.

[103] H. A. Abbass, R. Sarker, and C. Newton, "PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems," in *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, vol. 2. Piscataway, New Jersey: IEEE Service Center, May 2001, pp. 971–978.

[104] Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *Trans. Sys. Man Cyber. Part B*, vol. 29, no. 6, pp. 716–725, 1999.
[105] F. de Toro, J. Ortega, J. Fernandez, and A. F. Diaz, "Parallel genetic algorithm for multiobjective optimization," in *Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based processing*, 2002, pp. 384–391.
[106] L. Oliveira, R. Sabourin, F. Bortolozzi, and C. Suen, "Feature selection for ensembles:a hierarchical multi-objective genetic algorithm approach," in *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, Aug. 2003, pp. 676–680.
[107] L. S. Oliveira, M. Morita, R. Sabourin, and F. Bortolozzi, "Multi-objective Genetic Algorithms to Create Ensemble of Classifiers," in *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Guanajuato, México: Springer. Lecture Notes in Computer Science Vol. 3410, March 2005, pp. 592–606.
[108] Z. Zhang and P. Yang, "An ensemble of classifiers with genetic algorithm based feature selection," *IEEE Intelligent Informatics Bulletin*, vol. 9, no. 1, pp. 18–24, 2008.
[109] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "On biclustering of gene expression data," *Current Bioinformatics*, vol. 5, no. 3, pp. 204–216, 2010.
[110] P. A. D. de Castro, F. O. de França, H. M. Ferreira, and F. J. Von Zuben, "Applying biclustering to text mining: an immune-inspired approach," in *Proc. 6th Int. Conf. Artificial immune systems (ICARIS'07)*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 83–94.
[111] D. I. Ignatov, J. Poelmans, and V. Zaharchuk, "Recommender system based on algorithm of bicluster analysis RecBi," *CoRR*, vol. abs/1202.2892, 2012.
[112] F. O. d. Franca, H. M. Ferreira, and F. J. V. Zuben, "Applying biclustering to perform collaborative filtering," in *Proc. Seventh Int. Conf. Intell. Sys. Des. and Appl. (ISDA-07)*, 2007, pp. 421–426.
[113] Y. Cheng and G. M. Church, "Biclustering of gene expression data," in *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB'00)*, 2000, pp. 93–103.
[114] P. Larranaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. MA: Kluwer Academic Publisher, 2001.
[115] B. S. P. Mishra, A. K. Addy, R. Roy, and S. Dehuri, "Parallel multi-objective genetic algorithms for associative classification rule mining," in *Proc. Int. Conf. Commun., Comput. & Security*, ser. ICCCS '11, 2011, pp. 409–414.
[116] T. Özyer and R. Alhajj, "Parallel clustering of high dimensional data by integrating multi-objective genetic algorithm with divide and conquer," *Applied Intelligence*, vol. 31, no. 3, pp. 318–331, Dec. 2009.
[117] H. Trautmann, T. Wagner, B. Naujoks, M. Preuss, and J. Mehnen, "Statistical methods for convergence detection of multi-objective evolutionary algorithms," *Evol. Comput.*, vol. 17, no. 4, pp. 493–509, Dec. 2009.

Anirban Mukhopadhyay is currently an Associate Professor and Head of the Department of Computer Science and Engineering, University of Kalyani, Kalyani, India. He did his B.E. from National Institute of Technology, Durgapur, India, in 2002 and M.E. from Jadavpur University, Kolkata, India, in 2004, respectively, both in Computer Science and Engineering. He obtained his Ph.D. in Computer Science from Jadavpur University in 2009. Dr. Mukhopadhyay is the recipient of the University Gold Medal and Amitava Dey Memorial Gold Medal from Jadavpur University in 2004. Dr. Mukhopadhyay visited University of Heidelberg and German Cancer Research Center (DKFZ), Heidelberg, Germany during 2009-10 as a post-doc feloow, I3S laboratory, University of Nice Sophia-Antipolis, Nice, France in 2011 as a Visiting Professor, and University of Goettingen, Germany, as a Visiting Scientist with DAAD scholarship in 2013. He has coauthored one book and about 100 research papers in various International Journals and Conferences. His research interests include soft and evolutionary computing, data mining, multiobjective optimization, pattern recognition, bioinformatics, and optical networks.

Ujjwal Maulik is currently a Professor in the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India. Dr. Maulik did his Bachelors in Physics and Computer Science in 1986 and 1989 respectively. Subsequently, he did his Masters and Ph.D in Computer Science in 1991 and 1997 respectively. Dr. Maulik has worked in Center for Adaptive Systems Application, Los Alamos, New Mexico, USA in 1997, University of New South Wales, Sydney, Australia in 1999, University of Texas at Arlington, USA in 2001, Univ. of Maryland Baltimore County, USA in 2004, Fraunhofer Institute AiS, St. Augustin, Germany in 2005, Tsinghua University, China in 2007 and University of Rome, Italy in 2008, University of Heidelberg, Germany in 2009, German Cancer Research Center (DKFZ) in 2010, 2011 and 2012, Grenoble INP, France in 2010 and 2013 and ICM, Univ. of Warsaw, Poland in 2013. He is a Fellow of IETE, India. Dr. Maulik is a co-author of 7 books and about 250 research publications. He is the recipient of the Govt. of India BOYSCAST fellowship in 2001, Alexander Von Humboldt Fellowship for Experienced Researchers in 2010, 2011 and 2012 and Senior Associate of ICTP, Italy in 2012. He is the founder Member of IEEE Computational Intelligence Society (CIS) Chapter, Calcutta Section and has worked as Secretary and Treasurer in 2011, as Vice Chair in 2012 and currently working as the Chair. His research interests include Computational Intelligence, Bioinformatics, Combinatorial Optimization, Pattern Recognition, Data Mining.

Sanghamitra Bandyopadhyay did her B.Tech., M.Tech. and Ph.D. in Computer Science from Calcutta University, India, IIT Kharagpur, India, and ISI Kolkata, India, respectively. She is currently a Professor at the Indian Statistical Institute, Kolkata, India. She has worked at the Los Alamos National Laboratory, Los Alamos, New Mexico, University of New South Wales, Sydney, Australia, University of Texas at Arlington, University of Maryland at Baltimore, Fraunhofer Institute, Germany, Tsinghua University, China, University of Rome, Italy, University of Heidelberg, Germany and Max Planck Institute for Informatik, Saarbrucken, Germany. She has published five books more than 250 technical articles. She is a Fellow of the National Academy of Sciences, Allahabad, India (NASI), and Indian National Academy of Engineering (INAE). Dr. Bandyopadhyay is the recipient of several prestigious awards including the Dr. Shanker Dayal Sharma Gold Medal and also the Institute Silver Medal from IIT, Kharagpur, India, in 1994, the Young Scientist Awards of the Indian National Science Academy (INSA), 2000, the Indian Science Congress Association (ISCA), 2000, the Young Engineer Award of the Indian National Academy of Engineering (INAE), 2002, the Swarnajayanti fellowship from the Department of Science and Technology (DST), 2007, prestigious Shanti Swarup Bhatnagar Prize in Engineering Science, 2010, and the Humboldt Fellowship from Germany, 2009. She has been selected as a Senior Associate of ICTP, Italy in 2013. Her research interests include bioinformatics, soft and evolutionary computation, pattern recognition and data mining.

Carlos A. Coello Coello received the B.Sc. degree in civil engineering from the Universidad Autónoma de Chiapas, México, and the M.Sc. and the PhD degrees in computer science from Tulane University, USA, in 1991, 1993, and 1996, respectively. He is currently Professor with Distinction (CINVESTAV-3F Researcher) and Chair of the Computer Science Department of CINVESTAV-IPN, in Mexico City, México. Dr. Coello has authored and co-authored over 355 publications which currently report over 19,400 citations according to *Google Scholar* (his h-index is 56). He is currently associate editor of the *IEEE Transactions on Evolutionary Computation* and serves in the editorial board of 14 other international journals. He has received the *2007 National Research Award* from the Mexican Academy of Science in the area of *exact sciences*, the *Medal to the Scientific Merit* from Mexico City's congress, the *Ciudad Capital: Heberto Castillo 2011 Award in Basic Science*, the *2012 National Medal of Science and Arts* in the area of *Physical, Mathematical and Natural Sciences*, and the *2013 IEEE Kiyo Tomiyasu Award*. Since January 2011, he is IEEE Fellow for his "contributions to multi-objective optimization and constraint-handling techniques." He is also a member of the ACM, Sigma Xi, and the Mexican Academy of Sciences. His major research interests are: evolutionary multi-objective optimization and constraint-handling techniques for evolutionary algorithms.