A Preliminary Study on Evolutionary Clustering for Multiple Instance Learning

Aurora Esteban
Dept. of Computer Science
University of Cordoba
Cordoba, Spain
aestebant@uco.es

Amelia Zafra
Dept. of Computer Science
University of Cordoba
Cordoba, Spain
azafra@uco.es

Sebastián Ventura
Dept. of Computer Science
University of Cordoba
Cordoba, Spain
sventura@uco.es

***Abstract*—Since its beginnings, multiple instance learning studies have shown an excellent performance in the areas where it has been applied. This efficiency is due to multiple instance learning allows to represent a complex object by a set of feature vectors, being a more flexible representation to preserve more information than one based on single feature vector. This paper attempts to progress in this area carrying out a first study that introduces evolutionary algorithms for solving multiple instance cluster analysis. Specifically, we present four proposals of genetic algorithms for multi-instance partitional clustering: three of them are adaptations of existing algorithms for single-instance clustering, while the last one is a novel approach based on CHC evolutionary algorithm. Moreover, two classic non-genetic partitional algorithms are included in the final comparison. Experimental results considering ten representative datasets show promising results for our proposal.**

***Index Terms*—multiple instance learning, clustering, genetic algorithm**

I. Introduction

Clustering task appears as one of the most representative fields in data mining to extract useful information through the raw data. Thus, cluster analysis, or clustering, attempts to group objects in such a way that similar objects belong the same group or cluster and dissimilar objects are included in other clusters [1]. Clustering has been tackled by a lot of proposals that can be categorized in partitioning, hierarchical, density-based, grid-based or model-based methods [2]. Similarly, clustering has been effectively applied to a wide range of engineering and scientific disciplines such as biology, medicine, computer vision or pattern recognition [2].

Multiple Instance Learning (MIL) [3] is considered an extension of traditional learning that introduces more flexibility to represent information. MIL has been widely applied to tasks including text categorization, content-based image retrieval, image annotation or drug activity prediction [4]. However, the most applications have been addressed from supervised learning perspective. Concretely, the main dealt task has been classification [4]. Thus, only a few clustering methods based on MIL can be found in the bibliography [5]–[9]. Due to the high heterogeneous space of search in MIL, current clustering algorithms for MIL tend to get a local optimum. To deal with this problem, in this paper we propose several clustering methods based on Evolutionary Algorithms (EAs) for MIL. Evolutionary algorithms, and more concretely, Genetic Algorithms (GAs) [10], are stochastic optimization methods based on natural selection and evolutionary process. In this field, GAs have been applied to many optimization problems finding optimal solutions. Among their applications is clustering in traditional unsupervised learning, but, from our knowledge, no proposals have been found for GAs applied to MIL.

This work is focus on partitional clustering. Partitional clustering algorithms generate a single partition of the data with a specified or estimated number of non-overlapping clusters based on the distance between the instances [1]. In general, partitional clustering is iterative and hill climbing. Further, the associated objective functions are highly nonlinear and multimodal, so that it usually converges to local optimum. In this context, GAs are presented as efficient proposals which allow to increase the probabilities to reach the global optimum [11]. Thus, this work adapts some of the more representative genetic clustering algorithms in bibliography to work in MIL. It also introduces a new evolutionary proposal based on the CHC adaptive search [12]. In order to evaluate the performance of evolutionary proposals, an experimental study is carried out including non-genetic partitional clustering algorithms.

This paper is organized as follow. In Section II, the related work about clustering in MIL is presented. Section III briefly presents the studied classic partitional clustering methods and relevant definitions in the problem domain. Section IV addresses the description of the developed GAs. In Section V, the experimental results are studied. Finally, Section VI shows the conclusions obtained and future work.

II. Related Work

Multiple Instance Learning (MIL) was introduced by Dietterich et al. [3] as a form of learning where training instances are grouped in unordered collections called bags. This representation introduces an important flexibility on the composition of the bags, the types of data distribution and the relationships between instances of a bag. For that, this problem formulation has attracted much attention from scientific community, especially in the recent years, when the amount of available data has increased exponentially [4].

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

So far, MIL has been predominated addressed by weakly supervised learning [4], with several consolidated techniques for the classification task. In unsupervised learning, specifically in clustering, there are several proposals that adapt well-know algorithms by means of specific dissimilarity metrics for bags of instances. Henegar et al. [5] introduce a partitional clustering method based on an unsupervised version of Citation-kNN, UC-kNN, which minimizes Silhouette index. Kriegel et al. [6] propose MIEM-Clustering, an expectation-maximization clustering for MIL based on Gaussian distribution of bags. COSMIC [7] is a hierarchical density-based clustering that also uses concept lattices. BAMIC [8] proposes the adaptation of *K*-medoids algorithm but dealing with complete bags instead of instances. To measure the distance between bags, it uses the Hausdorff distance. Finally, Zhang et al. [9] introduce M^{3}IC-MBM, a clustering method based on maximization of margin between bags with some relaxations of constraints like the concave-convex procedure and the cutting plane method.

To the best of our knowledge, no clustering methods based on GAs have been proposed for MIL. However, GAs have been widely applied to clustering in traditional learning from the early 2000s to present [11], enhancing the performance of clustering in several complex tasks like text-mining, image analysis or building of social networks. Among the first proposals in the field, Krishna et al. [13] introduce Genetic K-Means Algorithm (GKA), a GA where individuals are possible partitions of the data and with a specific operator based on K-means and a mutator focused on minimize the Total Within Cluster Variation (TWCV). In the same line, Lu et al. [14] propose FGKA, with several modifications centered in reducing the computational time of GKA. Bandyopadhyay et al. [15] propose another approach, GCUK, an algorithm able to estimate the optimal number of clusters by minimizing the Davies-Bouldin index, and whose individuals are possible centroids of the clusters. Recently, new proposals have been developed, like NK hybrid GA [16], that includes in a genetic search a density based clustering with fixed *K* over *N* local groups; or HG-Means [17], a *H*ybrid *G*enetic search for multi-start *K*-Means focused on the scalability for large datasets.

In this paper, we propose an adaptation of the mentioned classic GAs applied to partitional clustering by means of the development of a specific distance metric and genetic operators.

III. Background

In this section, we introduce the adaptation of two classic partitional clustering algorithms to MIL. These proposals are used to evaluate the performance with respect to evolutionary approaches. Firstly, it is specified the nomenclature and the metric used to compute the dissimilarity between bags.

A. Nomenclature

In this section, the specific nomenclature used in the problem domain is introduced. MIL datasets are composed of N bags, named as $U = \{b_1, ..., b_N\}$. Each bag has a variable number of instances m, such that $b_i = \{x_{1i}, ..., x_{mi}\}$ and every instance of every bag in U is a vector of D dimensions (the attributes of the instances in the MIL datasest), such that $x_{ij} = (y_{1ij}, ..., y_{Dij})$. The aim of the proposed algorithms will be to form K clusters $C = \{C_1, ..., C_K\}$, being each one a set of variable number of bags l, i. e. $C_i = \{b_{1i}, ..., b_{li}\}$. Moreover, each C_i is represented by a centroid c_i, which is either a bag or an instance depending on the algorithm, and all the centers are represented by $c = \{c_1, ..., c_K\}$. The size of element sets is defined with the operator $||$.

B. Hausdorff Distance

In MIL, every pattern is composed of a variable number of instances, therefore the distance between patterns must be based on this specification. Two main approaches for measuring the dissimilarity between MI objects can be distinguished: metrics which treat bags as point sets in a high-dimensional space and those which treat them as instance distributions [18]. In this work, we focus on the first group. Specifically, we use Hausdorff distance H_d, under which two bags are close to each other when every instance in one is close to an instance of the other. Closeness is defined through the underlying distance d employed between instances, which in our case is Euclidean distance, a widely used distance in this context [18]. Thus, classic H_d uses the maximum mismatch between the instances of the respective bags:

$$H_d(b_i, b_j) = \max_{x_{ki} \in b_i} \min_{x_{lj} \in b_j} d(x_{ki}, x_{lj}) \tag{1}$$

C. BAMIC algorithm

BAMIC was proposed by Zhang and Zhou [8] as an adaptation of the partitional clustering algorithm *K*-medoids. It groups the bags in *K* disjoints groups. Firstly, for each cluster k, a bag is chosen randomly as its centroid: $c_k = b_i \in U$ being i a random number in $[1, N]$. Secondly, the rest of bags are assigned to its closest center, building the clusters up. Then, for each C_k, the new center $c_k(next)$ is calculated as the bag with the minimum average distance to the rest of bags in C_k.

$$c_k(next) = \arg\min_{b_i \in C_k} \sum_{b_j \in C_k, i \neq j} \frac{1}{|C_k|} H_d(b_i, b_j) \tag{2}$$

If the new centers are the same bags than previous ones, i. e. $\forall k \in [1, K]\ c_k = c_k(next)$, the algorithm has converged so it finishes; if not, $c(next)$ overwrites c and the algorithm repeats the second and third steps.

D. MIKM algorithm

MIKM is an adaptation of the classic partitional clustering algorithm *K*-means to MIL. Its main characteristic is that, while it builds K partitions of bags, the centers of the clusters are instances. Thus, firstly it randomly picks K bags of U, and, inside of each one of them, it picks one instance, to form initial centers c. Secondly, each one of the bags are assigned to its closest center, building the clusters up. Then, for each C_k, new centroid $c_k(next)$ is calculated as the average of all the instances contained by its bags:

$$c_k(next) = \frac{1}{|C_k|} \sum_{b_i \in C_k} \frac{1}{|b_i|} \sum_{x_{ji} \in b_i} x_{ji} \tag{3}$$

If the new centers are the same bags than previous ones, i. e. $\forall k \in [1, K]\ c_k = c_k(next)$, the algorithm has converged and finishes; if not, $c(next)$ overwrites c and the algorithm repeats second and third steps.

IV. Evolutionary Multi-Instance Clustering

Classic partitional clustering methods are too sensitive to initial points to build the clusters up [13]. This problem also affects MIL, so in this paper we propose several MI clustering methods based on GAs to avoid the stagnation in a local optimum considering proposals with different individual representations and genetic operators.

A. MIGKA algorithm

Multi-Instance Genetic K-Means algorithm (MIGKA) is based on the proposal of Krishna and Murty [13] adapted to MIL. The goal of this genetic MI clustering is to find an optimal partition of the population in K given clusters, maximizing the within-homogeneity of them. With this aim, a population of individuals that code possible solutions is crossed and mutated for generations. Below, the main characteristics of the algorithm are introduced.

1) Individual representation: individuals are coded by means of a chromosome with so many genes as bags are in the dataset. Each gene takes the value of the cluster to which it has been assigned. Thus, each individual is coded as a vector of natural numbers $s_i = [g_1, ..., g_N]$ where $g_i \in \{1, ..., K\}$.

2) Initialization: the initial population is selected randomly. Thus, each gene of each individual of the population is initialized to a cluster number randomly selected from the uniform distribution over the set $\{1, ..., K\}$.

3) Fitness: the fitness of each individual is based on the homogeneity of the clusters that it codes. This is calculated through a MIL version of the total within-cluster variation (TWCV) [13] based on Hausdorff distance H_d (Section III-B). First, centroids c are computed as the average of all the instances contained by the bags in every C_k, see (3). Then, the variation of each C_k is computed on its bags in relation to its centroid c_k, and the TWCV is obtained by adding all the variations:

$$TWCV_{MI} = \sum_{k=1}^{K} \sum_{b_i \in C_k} H_d(b_i, c_k)^2 \tag{4}$$

4) Parent selection: the well-known roulette selector is used over the normalized fitness of the individuals. This selector follows a probability proportional to the goodness of the individual fitness.

5) Genetic operators: there are two operators to establish a high selection pressure during a generation, since the search space is very wide. They are based on Krishna and Murty proposal [13]. Thus, these operators perform modifications over random individuals (selected with a given probability) of the group of parents selected previously. The first operator performs a uniform mutation over individual genes, the next one performs a one-step *K*-means algorithm over the cluster assignation given by the individual. The details of the operators are addressed below.

- Mutation. This operator modifies each gene g_j of the chromosome s_i of an individual with a given probability, like the classic uniform mutation operator. However, the modification is not uniformly random between clusters, but with a probability based on the improvement of its fitness $F(s_i)$ with the change. Its steps are:
  1) It calculates the fitness of k modified versions of the selected individual, whose chromosome, s'_i, is equal to the previous one s_i but with gene $g_j = k$ for every $k \in \{1, ..., K\}$. Then, the sum of all them is calculated:
  $$F_{total}(s_i) = \sum_{k=1}^{K} F(s'_i) \tag{5}$$
  2) g_j is reassigned to k with a probability following a roulette approach. Thus, for each possible k it is calculated the probability of setting:
  $$P(g_j = k) = \frac{F(s'_i)}{F_{total}(s_i)} \tag{6}$$
- *KM* operator. This operator performs a single-step *K*-means algorithm over the solution given by the individual to be mutated. Thus, the generational drift is reduced increasing the GA converge possibilities. Given an individual s_i, the operator follow these steps:
  1) It calculates the new centroids $c(next)$ of the solution coded by s_i following (3).
  2) Each gene g_j is reassigned to closest $c_k(next)$ of its corresponding bag.
  3) It is checked if there is any empty cluster. If so, the gene of the closest bag to its centroid is reassigned to this cluster.

B. MIFGKA Algorithm

Multi-Instance Fast Genetic K-Means (MIFGKA) is based on Lu et al. proposal [14] adapted to MI clustering. It is a modification of MIGKA (see Section IV-A) with the aim of reducing the convergence time. Although most characteristics of MIFGKA are similar to those of MIGKA, the main differences to reduce computational time are the following:

1) Fitness: the fitness of each individual is calculated via a modified version of $TWCV_{MI}$ based on the work of [14] that we have called Fast $TWCV_{MI}$ ($FTWCV_{MI}$). This is not based in H_d like MIGKA (see Section IV-A3), but in the average of the instances that compose a bag. Thus, it is based on the variance definition. Firstly it is added the mean squared point of every bag of the whole dataset U. Then, it is subtracted the mean squared point of every cluster C_k as follow:

$$FTWC_{MI} = \sum_{i=1}^{N} \sum_{d=1}^{D} \frac{\sum_{x_{ji} \in b_i} y_{dji}}{|b_i|}^2 - \sum_{k=1}^{K} \frac{\sum\limits_{b_l \in C_k} \sum\limits_{d=1}^{D} \frac{\sum_{x_{jl} \in b_l} y_{djl}}{|b_l|}^2}{|C_k|} \tag{7}$$

2) Parent selection: one important difference of MIFGKA with respect to MIGKA is that illegal individuals are allowed in order to relieve the computational cost of correcting them. However, they are highly penalized in the selection process to minimize their probabilities to survive. Thus, although the selection process is analogous to MIGKA, the fitness of an individual $F(s_i)$ is modified following the criterion [14]:

$$F(s_i) = \begin{cases} G(s_i) \times F_{min}, & if\ s_i\ is\ illegal \\ 1.5 \times F_{max} - F(s_i), & otherwise \end{cases} \quad (8)$$

where $G(s_i)$ is the number of non-empty clusters presented in the solution given by s_i, F_{min} is the smallest fitness value of the legal individuals in the current population, if they exist, otherwise $F_{min} = 1$, and F_{max} is the maximum fitness value encountered until the present generation.

3) Genetic operators: similar features to those of MIGKA operators are followed (see Section IV-A5). However, the implementation details of both genetic operators are different to decrease their computational time.

- Mutation. This operator applies a uniform mutation to the genes of an individual chromosome based on the probability to improve its fitness. Thus, the probability of changing a gene j in s_i to k (i.e. assigning a bag b_j to a cluster k) is given by the distance between b_j and c_k. Further, illegal individuals must be considered, i. e. those with any of their genes taking the value of some cluster, which implies that cluster is empty. If a cluster is empty, the distance between bags and it is defined as 0. Thus, a bias is added to the mutation operation to avoid 0 division and promotes the conversion of illegal individuals to legal ones.

$$P(s_{ij} = k) = \frac{1.5 \cdot FarC(b_j) - H_d(b_j, c_k) + 0.5}{\sum_{l=1}^{K}(1.5 \cdot FarC(b_j) - H_d(b_j, c_l) + 0.5)} \quad (9)$$

  where $FarC(b_j) = \max_{k=1}^{K} H_d(b_j, c_k)$ is the farthest centroid from the bag.
- *KM* operator. This operator performs one iteration of *K*-means algorithm over the solution given by the individual to be mutated. It is equivalent to MIGKA version, but without the conversion of illegal individuals to legal ones.

C. MIGCUK Algorithm

Multi-Instance Genetic Clustering for Unknown K (MIGCUK) is based on the work of Bandyopadhyay and Maulik [15] adapted to MI clustering. It follows a different approach with respect to GAs previously presented. Thus, it uses a representation of individuals based on centroids. Further, it has the ability to automatically find an optimal number of clusters within a given range. The main characteristics of this GA are introduced following.

1) Individual representation: individuals represent the clusters' centroids. Since centroids are bags, the genetic individuals are coded as integer arrays where each gene takes the value of a bag index. In this case, the number of clusters for every solution coded by an individual is variable in a given range $[K_{min}, K_{max}]$.

2) Initialization: initial population is selected randomly but with some conditions. Thus, each gene of an individual is initialized to bag index in the range $[1, N]$, or alternatively, it can take an invalid value indicating that this possible centroid will be ignored. In this process, two conditions have to be checked to create valid individuals:

1) The number of valid centroids is greater than K_{min}.
2) There are no repeated indexes in genes that code the current clusters.

3) Fitness: the fitness of each individual is calculated using the Davies-Bouldin index [19] adapted to MIL. For each cluster, similarities between it and all other clusters are computed, then it averages the maximum similarities of all the clusters as following:

$$DB_{MI} = \frac{1}{K}\sum_{k=1}^{K} \max_{l,l\neq k} \frac{\frac{1}{|C_k|}\sum_{b_i \in C_k} H_d(b_i, c_k) + \frac{1}{|C_l|}\sum_{b_j \in C_l} H_d(b_j, c_l)}{H_d(c_k, c_l)} \quad (10)$$

The distance between bags and centroids is measured with the Hausdorff Distance H_d (Section III-B).

4) Parent selection: like in previous described proposals, the selection is carried out by the roulette selector.

5) Genetic operators: during a generation, crossover and mutation operations are performed over the selected parents. The details of these operators are addressed below:

- Crossover. Given two individuals previously selected in the population. This operator applies one-point crossover between them to produce two new individuals. Thus, a random point is selected and genes to the right of it are swapped between the two parent chromosomes.
- Mutation. Giving an individual to mutate s_i, this operator changes each gene with a given probability. Thus, the centroid changes to close bags in the dataset. This closeness is defined as $1/4$ of the number of bags (N). Specifically, the new value of a gene $g_j \in s_i$ is generated randomly with a variation of $[0.25N - g_j, 0.25N + g_j]$.

D. CHCMIC Algorithm

CHC for Multi-Instance Clustering (CHCMIC) is a new proposal developed in this work. It is based on the application of the adaptive search of CHC [12] to the problem of MI clustering. CHC is a classic GA proposed by Esthelman in 1991. It stands out because of its combination of diversification and high convergence. Its diversification is given by incest prevention in the parent selection and population restart when it is stagnant. While the algorithm high convergence is given by its elitist selection as well as the preservation of best individuals in population restart. These characteristics make CHC an appropriate approach to address the wide space of search of multi-instance clustering. The main features of our proposal are addressed below:

1) Individual representation: each individual is a integer array where each gene correspond to a bag and the value that it takes is the cluster to which the bag is assigned. See Section IV-A1 for more details.

a) Initialization: the population is generated randomly like in MIGKA and MIFGKA. See Section IV-A2 for more details.

2) Fitness: it is used the Davies-Bouldin index based on the Hausdorff distance to measure similarities between bags, defined previously in Eq. (10).

3) Parent selection: the main characteristic of CHCMIC is its *incest prevention*, i.e. only sufficiently dissimilar individuals can be crossed to produce the offspring. Thus, incest prevention is measured with a threshold d based on Hamming distance. Two individuals have to be at a higher distance than d to be crossed. d is a threshold that adapts itself to the state of the search: if during a generation no crossing can be made, d decreases in one unit, so in the next generation, the needed Hamming distance between two parents will be less restrictive. The initial value of d is set automatically depending on the length of the chromosome, or what is the same, the number of bags in the dataset N. Specifically, $d = 1/4N$ [12].

4) Population restart: another important feature of CHCMIC is the population restart when the search is stagnant. This happens when the *incest prevention* threshold d reaches 0 value. Then, the n best individuals founded are kept to the next generation and the rest of population is regenerated randomly. Specifically, we set $n = 10$. With this process, on the one hand it is kept the elitist generation and, on the other hand, diversity in the population is promoted.

5) Genetic operators: three operators perform changes over the population. Firstly, the offspring is generated by means of the crossover between selected parents. Then, the mutation operator and the *KM* operator previously defined in MIGKA are also applied over the population in order to promote the selective search. The details of these operators are the following:

- Crossover. The operator takes two parents to produce two offspring using the one-point crossover previously defined on Section IV-C5.
- Mutation. This operator applies a uniform mutation over each gene of the individual with a given probability. The value of the gene changes randomly following a distribution of how the change improves the individual fitness. For more details, it can be seen Section IV-A5.
- *KM* operator. This operator performs a single-step *K*-means algorithm over the solution given by the individual to be mutated. The aim of this operator is to reduce the generational drift by increasing the possibilities of convergence. For more details, it can be seen Section IV-A5.

V. Experimental study

The experimentation carried out compares the performance of both evolutionary and classic proposals of MI clustering in different MIL problems. This section presents datasets, parameters configuration, validation metrics and results.

A. Datasets

MIL has been applied successfully on numerous and interesting applications in different domains. In this study, several datasets belonging to different domains are used [4]: drug activity prediction, molecules mutagenicity, content-based image retrieval and the well-known East–West challenge. The most representative information attending to number of bags, instances, attributes and classes is specified in Table I.

TABLE I
DATASETS INFORMATION

Dataset	Bags			Attributes	Instances	Avg.bag size
	Positive	Negative	Total			
Musk1	47	45	92	166	467	5.17
Musk2	39	63	102	166	6598	64.69
MutAtoms	125	63	188	10	1618	8.61
MutBonds	125	63	188	16	3995	21.25
MutChains	125	63	188	24	5349	28.45
ImgElephant	100	100	200	230	1391	6.96
ImgTiger	100	100	200	230	1220	6.10
ImgFox	100	100	200	230	1320	6.60
EastWest	10	10	20	24	213	10.65
WestEast	10	10	20	24	213	10.65

According to the study of Cooper and Milligan [20], attribute values have been normalized in order to avoid misleading in calculating the distance between bags when the attributes ranges are different. Thus, given the d^{th} attribute of the i^{th} instance in the j^{th} bag, y_{dij}, the normalization applied because of its high performance is Z_5 [20]. It is defined as:

$$Z_5(y_{dij}) = \frac{y_{dij} - \min_{y_{dkl} \in U}(y_{dkl})}{\max_{y_{dkl} \in U}(y_{dkl}) - \min_{y_{dkl} \in U}(y_{dkl})} \qquad (11)$$

B. Experimental setup

All algorithms have been developed using Java language. On the one hand, classic partitional methods, BAMIC and MIKM, have been implemented from the WEKA data mining tool [21]. On the other hand, genetic clustering methods have been developed using JCLEC software [22].

A specific study of parameters has been carried out for each algorithm due to its particularities, such as, individual representation, genetic operators and computational time, but keeping a fair comparison by performing the same number of evaluations for every GA. The specific parameters of evolutionary clustering proposals are detailed in Table II. Moreover, K value is a parameter used for all proposals and it represents the number of clusters to form. As, in all datasets is known the number of classes, K has been set to number of classes of each dataset.

Finally, as algorithms are stochastic, experiments have been repeated with 15 different seeds and results shown their average.

C. Validation metrics

To evaluate and compare the experimental results, both internal validity criteria metrics (i.e. how well the clustering results are based on information intrinsic to data) and external validity criteria metrics (i.e. how well the clustering results match some prior knowledge about the data) are considered.

TABLE II
EVOLUTIONARY ALGORITHM PARAMETERS

	MIGKA	MIFGKA	MIGCUK	CHCMIC
Population size	150	150	150	150
Generations number	150	150	150	150
Mutation probability	0.8	0.8	0.3	0.8
Gene mutation probability	0.7	0.7	0.7	0.7
KM operator probability	0.2	0.2	-	0.2
Crossover probability	-	-	0.2	-

For internal validation, several of the most representative indexes in the field [19] are studied. These metrics have been adapted to MIL, which is denoted by the subscript MI:

- Silhouette index. It validates the clustering performance based on the pairwise difference of between-cluster and within-cluster distances. Thus, it relates the minimal mean distance between a bag b_i and all the bags in the rest of clusters and the mean distance between b_i and all other bags in its cluster. It is a metric to maximize. It is defined as following:

$$S_{MI} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{b_i\in C_k}\frac{Ext_{MI}(b_i) - Int_{MI}(b_i)}{\max(Ext_{MI}(b_i), Int_{MI}(b_i))} \quad (12)$$

  where

$$Ext_{MI}(b_i) = \min_{l,l\neq k}\frac{1}{|C_l|}\sum_{b_j\in C_l} H_d(b_i, b_j) \quad (13)$$

$$Int_{MI}(b_i) = \frac{1}{C_k - 1}\sum_{b_j\in C_k, j\neq i} H_d(b_i, b_j) \quad (14)$$

- Davies-Bouldin index. It considers both compactness and separation of formed clusters. It has been defined previously in (10). It is a metric to minimize.
- S_Dbw index. It considers density to metric the inter-cluster separation. The basic idea is that, for each pair of cluster centers c_i and c_j, at least one of their densities should be larger than the density of their midpoint u_{ij}. The index is the summation of this separation and the intra-cluster compactness. It is a metric to minimize. It is defined as following:

$$S\_Dbw_{MI} = Scat_{MI}(K) + Den_{MI}(K) \quad (15)$$

  where

$$Scat_{MI}(K) = \frac{1}{K}\sum_{i=1}^{K}\frac{||\sigma(C_k)||}{||\sigma(U)||} \quad (16)$$

$$Den_{MI}(K) = \frac{1}{K^2 - K}\sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\frac{\sum_{b_k\in C_i\cup C_j} H_d(b_k, u_{i,j})}{\max(\sum_{b_l\in C_i} H_d(b_l, c_i), \sum_{b_m\in C_j} H_d(b_m, c_j))} \quad (17)$$

In order to study external validation, it is used a confusion matrix with real classes as rows and clusters as columns. Thus, the cluster with more bags of one specific class is assigned to that class. From this, it can be followed the classification approach of true positives (TP), true negative (TN), false positive (FP) and false negative (FN).

In this work, several of the most representative external validation metrics are included [23]. These metrics have been adapted to MIL, which is denoted by the subscript MI:

- Rand index. It counts the pair-match between classes and clusters. It is a metric to maximize. It is specified below:

$$RI_{MI} = \frac{TP + TN}{TP + FP + TN + FN} \quad (18)$$

- Entropy. It computes the bag probability of cluster k to belong to the class i. It is a metric to minimize. It is specified below:

$$E_{MI} = \sum_{k=1}^{K}\frac{|C_k|}{N}\left(-\sum_{i=1}^{I} p_{ik}\log(p_{ik})\right) \quad (19)$$

- F1 measure. It combines the precision $Pr = TP/(TP + FP)$ and the recall $Re = TP/(TP + FN)$ of the clustering. It is a metric to maximize. It is specified below:

$$F1_{MI} = \frac{2RePr}{Pr + Re} \quad (20)$$

D. Comparative study

This section presents and discusses experimental results. Both internal and external validity measures are considered for comparing classical and evolutionary partitional MI clustering algorithms. Statistical tests are used to evaluate the performance.

Attending to internal validation metrics (Table III), GAs usually obtain better results than classic partitional MI clustering algorithms. Concretely, CHCMIC tends to reach the best results for all metrics, specifically DB_{MI} and S_Dbw_{MI} indexes. Thus, it obtains more compact, differentiated and dense clusters than the rest of approaches. Regarding to Silhouette index, results are not so conclusive: while CHCMIC obtains the best average results, the differences with respect other algorithms are more reduced. If the rest of evolutionary proposals are evaluated, there is no a clear tendency compared to classic approaches. The results of the Friedman's test for internal evaluation, including the Friedman's statistics and the p-values are shown in Table V. Ranking assigned for Friedman's test (see Table IV) shows that CHCMIC obtains the lowest ranking for all measures, indicating that it is the best proposals for the most datasets. In DB_{MI} and S_Dbw_{MI} indexes, the Friedman's test rejects the null hypothesis and, therefore, it determines that significant differences exist in the performance of the algorithms at 99% confidence, so the Shaffer's post-hoc test was also performed. Significant differences among algorithms for these measures at 99% confidence level are shown in Figure 1. CHCMIC performs significantly better than BAMIC and other proposals being the control algorithm in metrics considered.

Attending to external validation metrics (Table VI), similar results can be found. Again, CHCMIC usually reaches the best values for the different metrics in the most datasets. The results of the Friedman's test for external evaluation including the Friedman's statistics and the p-values are shown in Table VIII. Ranking assigned for Friedman's test (see Table VII) shows

TABLE III
RESULTS FOR INTERNAL VALIDATION METRICS

		BAMIC	MIKM	MIGKA	MIFGKA	MIGCUK	CHCMIC
Musk1	S_{MI}	0.1656	0.1519	0.1597	0.0381	0.1156	**0.1715**
	DB_{MI}	2.4735	2.0318	2.5041	4.4574	2.9341	**2.0035**
	S_Dbw_{MI}	0.8856	**0.7577**	0.8558	0.9586	0.8839	0.8008
Musk2	S_{MI}	0.1545	**0.1754**	0.1717	0.0239	0.1081	0.1448
	DB_{MI}	3.3703	2.1625	2.5934	5.0817	3.9003	**2.0819**
	S_Dbw_{MI}	0.8699	**0.8163**	0.8226	0.9616	0.8641	0.8489
MutA	S_{MI}	0.5610	0.2114	0.5678	0.2226	0.1677	**0.6043**
	DB_{MI}	10.9839	18.8995	10.4630	12.7672	21.5018	**8.6559**
	S_Dbw_{MI}	0.9952	1.0328	**0.9940**	0.9770	0.9948	1.0065
MutB	S_{MI}	0.1085	0.1326	**0.3003**	0.1743	0.0956	0.2912
	DB_{MI}	21.2972	22.2912	15.2224	21.9042	20.7270	**8.7337**
	S_Dbw_{MI}	1.0306	1.0235	1.0121	0.9942	0.9994	**0.9985**
MutC	S_{MI}	0.0938	0.1301	0.0636	-0.0355	0.1265	**0.2484**
	DB_{MI}	6.7666	6.6140	8.6407	6.8004	19.9563	**4.6057**
	S_Dbw_{MI}	0.9733	1.0028	**0.9450**	0.9547	1.0160	0.9724
ImgE	S_{MI}	0.0208	0.0103	0.0178	0.0083	**0.0271**	0.0139
	DB_{MI}	9.0270	9.1321	9.5317	12.1334	10.6597	**6.9825**
	S_Dbw_{MI}	1.0114	0.9985	**0.9949**	1.0006	1.0145	0.9961
ImgT	S_{MI}	0.0091	0.0283	0.0372	0.0418	**0.0467**	0.0280
	DB_{MI}	13.1027	6.4840	6.2685	5.2874	7.2043	**4.8710**
	S_Dbw_{MI}	1.0105	0.9979	0.9994	0.9809	1.0331	**0.9749**
ImgF	S_{MI}	0.0353	0.0510	0.0296	0.0550	**0.0789**	0.0339
	DB_{MI}	6.7059	6.1328	6.6835	7.2814	6.2654	**5.5788**
	S_Dbw_{MI}	1.0084	1.0129	1.0101	1.0196	1.1205	**1.0044**
EastW	S_{MI}	0.0160	0.0829	0.0468	0.0661	0.0111	**0.1189**
	DB_{MI}	5.3412	4.8629	6.7544	**3.2563**	6.6069	3.7951
	S_Dbw_{MI}	1.0320	0.9566	0.9821	**0.8954**	1.0178	0.9365
WestE	S_{MI}	0.0160	0.0829	0.0469	0.0661	0.0280	**0.1122**
	DB_{MI}	5.3412	4.8629	6.7730	**3.2563**	6.9294	3.8603
	S_Dbw_{MI}	1.0320	0.9566	0.9821	**0.8954**	1.0040	0.9371

TABLE VI
RESULTS FOR EXTERNAL VALIDATION METRICS

		BAMIC	MIKM	MIGKA	MIFGKA	MIGCUK	CHCMIC
Musk1	RI_{MI}	0.5304	0.5000	0.5348	0.5174	**0.5674**	0.5152
	E_{MI}	0.9942	0.9995	0.9941	0.9965	**0.9704**	0.9957
	$F1_{MI}$	0.5057	0.3658	0.4279	0.4448	**0.6453**	0.4373
Musk2	RI_{MI}	0.5725	0.5863	0.5373	0.5667	0.5569	**0.6588**
	E_{MI}	0.9503	0.9572	0.9585	0.9361	**0.9219**	0.9243
	$F1_{MI}$	**0.4179**	0.3259	0.3625	0.3438	0.4080	0.3828
MutA	RI_{MI}	0.6915	0.6649	**0.6957**	0.5904	0.5862	0.6872
	E_{MI}	0.8901	0.9123	0.8859	0.8741	**0.8703**	0.8962
	$F1_{MI}$	0.8027	0.7953	**0.8060**	0.7425	0.6196	0.8051
MutB	RI_{MI}	0.6596	0.6553	0.6702	0.6266	0.6170	**0.6840**
	E_{MI}	0.9169	0.9165	0.8880	0.9070	**0.8451**	0.8892
	$F1_{MI}$	0.7949	0.7915	0.7669	0.7666	0.6460	**0.8081**
MutC	RI_{MI}	0.6543	0.6585	0.6596	0.6596	0.6106	**0.6702**
	E_{MI}	0.9138	0.9163	0.9182	0.9169	0.9121	**0.9116**
	$F1_{MI}$	0.7910	0.7941	0.7895	0.7949	0.7154	**0.8013**
ImgE	RI_{MI}	0.5522	0.5860	0.5260	0.5870	0.5550	**0.6900**
	E_{MI}	0.9918	0.9570	0.9974	0.9679	0.9832	**0.8870**
	$F1_{MI}$	0.5283	0.5430	0.5218	0.5065	0.5956	**0.6555**
ImgT	RI_{MI}	0.5150	0.5540	0.5290	0.5380	0.5170	**0.5750**
	E_{MI}	0.9993	0.9880	0.9959	0.9926	0.9978	**0.9789**
	$F1_{MI}$	**0.5611**	0.4750	0.5133	0.4474	0.6012	0.4473
ImgF	RI_{MI}	0.6250	0.6250	0.6230	0.6120	0.5920	**0.7470**
	E_{MI}	0.9540	0.9459	0.9545	0.9599	0.9453	**0.7775**
	$F1_{MI}$	0.6073	0.5929	0.6066	0.5548	0.4155	**0.7854**
EastW	RI_{MI}	0.6400	0.5400	0.5700	0.5500	**0.6500**	0.6300
	E_{MI}	0.9221	0.9885	0.9822	0.9481	0.8825	**0.8553**
	$F1_{MI}$	0.7063	0.3747	0.4935	0.1818	**0.7156**	0.4103
WestE	RI_{MI}	**0.6400**	0.5400	0.5800	0.5500	0.5700	0.6200
	E_{MI}	0.9221	0.9885	0.9749	0.9481	0.9797	**0.8798**
	$F1_{MI}$	0.5238	0.6619	0.6249	0.6898	0.5219	**0.7205**

that CHCMIC obtains the lowest ranking for all measures, indicating that it is the best proposals for most datasets. In Rand index and Entropy metrics, the Friedman's test rejects the null hypothesis and, therefore, it determines that significant differences exist in the performance of the algorithms at 95% confidence, so the Shaffer's post-hoc test was also performed. Significant differences among algorithms for these metrics at 95% confidence level are shown in Figure 2. CHCMIC is the control algorithm for all metrics considered and performs significantly better than MIKM and other proposals.

TABLE IV
FRIEDMAN'S AVERAGE RANKINGS FOR INTERNAL VALIDATION METRICS

	BAMIC	MIKM	MIGKA	MIFGKA	MIGCUK	CHCMIC
S_{MI}	4.0	3.2	3.3	4.1	3.9	**2.5**
DB_{MI}	3.8	3.2	3.8	4.1	4.9	**1.2**
S_Dbw_{MI}	4.8	3.4	2.8	2.9	4.8	**2.3**

TABLE V
FRIEDMAN'S TEST RESULTS FOR INTERNAL VALIDATION METRICS

Metric	p-value	Statistic
S_{MI}	0.3658	5.4286
DB_{MI}	**0.0004**	22.5140
S_Dbw_{MI}	**0.0062**	16.2290

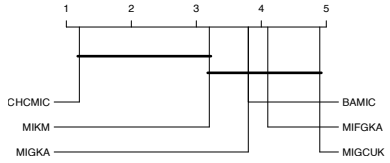
TABLE VII
FRIEDMAN'S AVERAGE RANKINGS FOR EXTERNAL VALIDATION METRICS

	BAMIC	MIKM	MIGKA	MIFGKA	MIGCUK	CHCMIC
RI_{MI}	3.25	3.95	3.45	4.05	4.40	**1.90**
E_{MI}	4.00	4.50	4.30	3.80	2.40	**2.00**
$F1_{MI}$	2.70	4.10	3.70	4.40	3.70	**2.40**

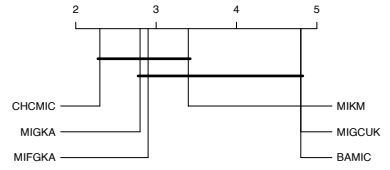
TABLE VIII
FRIEDMAN'S TEST RESULTS FOR EXTERNAL VALIDATION METRICS

Metric	p-value	Statistic
RI_{MI}	**0.0465**	11.2570
E_{MI}	**0.0083**	15.5430
$F1_{MI}$	0.1149	8.8571

These results show promising results of CHCMIC for all evaluation metrics considering both other evolutionary proposals adapted to MIL and classic approaches. Nevertheless, there are no a conclusive results for all GAs showing that MI

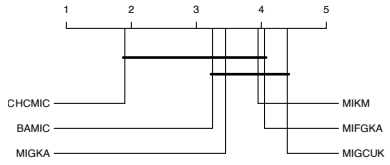


(a) DB_{MI} index

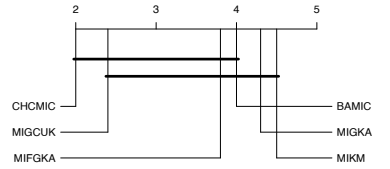


(b) $\mathrm{S\_Dbw}_{MI}$ index

Fig. 1. Critical distance for internal metrics of Shaffer's procedure. 99% conf.



(a) Rand_{MI} index



(b) $\mathrm{Entropy}_{MI}$ measure

Fig. 2. Critical distance for external metrics of Shaffer's procedure. 95% conf

clustering is a complex field that should be addressed with specific methods beyond adaptation of traditional clustering techniques. A more exhaustive study with more datasets and algorithms could confirm the excellent performance of specific GAs in this field.

VI. Conclusions and future work

This paper has presented a first adaptation to MIL of three classical GA applied to partitional clustering. Moreover, a novel approach is presented, CHCMIC, that applies the adaptive search and the population restart to genetic MI clustering. These algorithms are compared with classical approaches for partitional clustering. CHCMIC results are promising, as it obtains the best balance between internal and external validation metrics showing the relevance of GA in this field.

As future work, a more exhaustive study with more datasets and MI clustering methods should be carried out. Thus, more proposals beyond partitional clustering could be developed, as well as more MI-based dissimilarity metrics in order to extend the obtained conclusions.

VII. Acknowledgment

Authors gratefully acknowledge the financial subsidy pro-vided by Spanish Ministry of Science and Innovation and the European Fund of Regional Development under the Project TIN2017-83445-P.

References

[1] M. N. Murty, P. J. Flynn, and A. K. Jain, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, p. 60, 1999.

[2] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[3] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, 2002.

[4] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sánchez-Tarragó, and S. Vluymans, *Multiple Instance Learning*. Springer, 2016.

[5] C. Henegar, K. Clément, and J. D. Zucker, "Unsupervised multiple-instance learning for functional profiling of genomic data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4212 LNAI, 2006, pp. 186–197.

[6] H. P. Kriegel, A. Pryakhin, M. Schubert, and A. Zimek, "COSMIC: Conceptually specified multi-instance clusters," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2006, pp. 917–921.

[7] H.-p. Kriegel, A. Pryakhin, and M. Schubert, "An EM-Approach for Clustering Multi-Instance Objects," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2006, pp. 139–148.

[8] M. L. Zhang and Z. H. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *Applied Intelligence*, vol. 31, no. 1, pp. 47–68, 2009.

[9] D. Zhang, F. Wang, L. Si, and T. Li, "Maximum margin multiple instance clustering with applications to image and text clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 5, pp. 739–751, 2011.

[10] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Choice Reviews Online*, vol. 27, no. 02, pp. 27–0936–27–0936, 1989.

[11] A. José-García and W. Gómez-Flores, "Automatic clustering using nature-inspired metaheuristics: A survey," *Applied Soft Computing Journal*, vol. 41, pp. 192–213, 2016.

[12] L. J. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination," *Foundations of genetic algorithms*, vol. 1, pp. 265–283, 1991.

[13] K. Krishna and M. Narasimha Murty, "Genetic K-Means Algorithm K." *IEEE Transactions on Systems, Man and Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999.

[14] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown, "FGKA: A fast genetic K-means clustering algorithm," in *Proceedings of the ACM Symposium on Applied Computing*, 2004, pp. 622–623.

[15] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification," *Pattern Recognition*, vol. 35, no. 6, pp. 1197–1208, 2002.

[16] R. Tinós, L. Zhao, F. Chicano, and D. Whitley, "NK Hybrid Genetic Algorithm for Clustering," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 748–761, 2018.

[17] D. Gribel and T. Vidal, "HG-MEANS: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering," *Pattern Recognition*, vol. 88, pp. 569–583, 2019.

[18] V. Cheplygina, D. M. Tax, and M. Loog, "Multiple instance learning with bag dissimilarities," *Pattern Recognition*, vol. 48, no. 1, pp. 264–275, 2015.

[19] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Understanding and enhancement of internal clustering validation measures," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 982–994, 2013.

[20] M. C. Cooper and G. W. Milligan, "A study of standardization of variables in cluster analysis," *Journal of Classification*, vol. 5, no. 2, pp. 181–204, 1988.

[21] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

[22] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervás, "JCLEC: A Java framework for evolutionary computation," *Soft Computing*, vol. 12, no. 4, pp. 381–392, 2008.

[23] M. Rezaei and P. Franti, "Set matching measures for external cluster validity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2173–2186, 2016.