*Manuscript (including abstract)
Click here to view linked References

Toward High Performance Solution Retrieval in Multiobjective Clustering

Alvaro Garcia-Piquer[a,*], Andreu Sancho-Asensio[b], Albert Fornells[c], Elisabet Golobardes[b], Guiomar Corral[d], Francesc Teixidó-Navarro[c]

[a]*Institut de Ciències de l'Espai (IEEC - CSIC), Campus UAB, Facultat de Ciències, Torre C5 - parell - 2a planta, E-08193 Bellaterra, Spain*
[b]*Research Group in Electronic and Telecommunications Systems and Data Analysis, Ramon Llull University, Quatre Camins 2, 08022 Barcelona, Spain*
[c]*Research Group in Tourism, Hospitality and Mobilities, School of Tourism and Hospitality Management - Sant Ignasi, Ramon Llull University, Marqués de Mulhacén 40-42, 08034 Barcelona, Spain*
[d]*Research Group in Internet Technologies and Storage, Ramon Llull University, Quatre Camins 2, 08022 Barcelona, Spain*

Abstract

The massive generation of unlabeled data of current industrial applications has attracted the interest of data mining practitioners. Thus, retrieving novel and useful information from these volumes of data while decreasing the costs of manipulating such amounts of information is a major issue. Multiobjective clustering algorithms are able to recognize patterns considering several objective function which is crucial in real-world situations. However, they dearth from a retrieval system for obtaining the most suitable solution, and due to the fact that the size of Pareto set can be unpractical for human experts, autonomous retrieval methods are fostered. This paper presents an automatic retrieval system for handling Pareto-based multiobjective clustering problems based on the shape of the Pareto set and the quality of the clusters. The proposed method is integrated in CAOS, a scalable and flexible framework, to test its performance. Our approach is compared to classic retrieval methods that only consider individual strategies by using a wide set of artificial and real-world datasets. This filtering approach is evaluated under large data volumes demonstrating its competence in clustering problems. Experiments support that the proposal overcomes the accuracy and significantly reduces the computational time of the solution retrieval achieved by the individual strategies.

Keywords: Soft-Computing, Genetic algorithms, Multiobjective optimization, Clustering,

1. Introduction

Clustering [39, 15, 32] is a trending data mining technique used in real-world situations to partition a data set into several groups according to some criteria and therefore identifying novel and potentially useful patterns from data. Conventional clustering algorithms are focused on obtaining groups by optimizing a single fitness function. In contrast, it can be difficult to obtain good data partitions in some real-world problems using a single objective function, and it is necessary to define several of them to obtain more accurate clusters [35]. These objective measures can be summarized in a single fitness function if they are disjoint. However, when the defined objectives conflict with each other it is necessary to define a fitness function for each objective in order to find a solution which would give acceptable values for all of them [7]. A widely used technique to competently carry out this is multiobjective clustering (MC) [30], which uses the concept of Pareto Optimum with a posteriori approach [8] for simultaneously optimizing a set of mutually confronted objectives in order to promote the definition of clusters. This technique returns a collection that contains a number of Pareto optimal solutions (the so called Pareto set), none of which can be further improved on any objective without degrading another one [12].

There are different strategies for multiobjective optimization such as Simulated Annealing [47] and Ant Colony Optimization [37], but Multiobjective Evolutionary Algorithms (MOEAs) [7] have become one of the most capable strategies to solve this kind of problems [17, 51] since they (1) work with a collection of solutions with different trade-offs among objectives, which are improved until a Pareto set with optimal trade-offs is obtained; (2) can be easily adapted to the type of data of the studied domain, due to the flexible knowledge

*Corresponding author
Email addresses: agarcia@ice.csic.es (Alvaro Garcia-Piquer), andreus@salleurl.edu (Andreu Sancho-Asensio), albert.fornells@tsi.url.edu (Albert Fornells), elisabet@salleurl.edu (Elisabet Golobardes), guiomar@salleurl.edu (Guiomar Corral), francesc.teixido@tsi.url.edu (Francesc Teixidó-Navarro)

representation used; and (3) are able to optimize different objectives without assuming any underlying structure of the objective functions. However, the performance of MOEAs can be compromised in large databases due to their high computational and memory usage requirements [19]. Moreover, one of the key challenges in Pareto-based MOEAs is the retrieval of the most suitable solution from the final Pareto set. This solution is typically identified by an expert in the domain. Nonetheless this process results in a subjective criterion and in a non trivial and tedious task if there are several solutions in the Pareto set. Thus, automatic methods are strongly required in order to help experts and simplify the identification of the most suitable solution, which can be beneficial in challenging domains such as health, smart networks or education. These are areas in which large volumes of data are generated.

In MC algorithms there are mainly two approaches to retrieve the most suitable solution from the Pareto set: (1) consider the shape of the Pareto set [43] or (2) consider the features related to the morphological properties of clusters [30]. The first method tries to identify the knee of the Pareto set to retrieve the solution with the best trade-off between objectives, but it does not take into account the resulting quality of clusters. The term quality is defined as how useful the solution is for the expert in the domain. Furthermore, quality is directly related to the shape, size and compactness of the clusters and the separation between them, characteristics which can be evaluated using clustering validation indexes [25, 26, 40]. The second method retrieves the best solution according to clustering validation indexes but its objective values could be unbalanced and the solution may only properly optimize a single objective.

The purpose of this paper is to propose a scalable retrieval filtering method that contemplates both the shape of the Pareto set and the quality of the clusters. The goal is to retrieve explanatory solutions with an acceptable trade-off between objectives in MC based on MOEAs. The proposed retrieval method is based on the observation that solutions with acceptable balance between objectives are placed around the knee of the Pareto front. The aim is to filter clustering solutions with less objective trade-off in order to retrieve the best solution from the remaining ones according to a clustering validation index. Thus, extra computations to evaluate non-interesting solutions are avoided. To test our approach we

use the *Clustering Algorithm based on multiObjective Strategies* (CAOS) [10, 22], a MC algorithm based on PESA-II [9]. CAOS uses a representation that does not depend on the number of instances of the data set, subsequently it is memory scalable [21]. Moreover, it scales the computational time of the clustering process by dividing the original data set to several subsets that are alternatively used in each generation of the MOEA process, thus it uses less data in each evolutionary cycle. This is performed in this way to avoid biasing the population by using only a single sample, while achieving low penalization in accuracy [2]. More specifically, the approach acts iteratively through the evolutionary cycle, being an automatic, adaptive system, thence fostering objectivity in the filtering parameters.

We compare the proposed method with the retrieval strategies based on (1) the shape of the Pareto set and (2) the morphological properties of clusters. All approaches are compared along a wide set of synthetic data sets [30] and real-world ones from the UCI [18] and KEEL [1] repositories. Furthermore, we carry out another set of experiments in data sets with large amounts of data in order to test the scalability capabilities of the method. Results show that accuracy and retrieval time are improved with this new proposal with a negligible additional cost to the evolutionary cycle. For a comparison between CAOS and other clustering methods, the reader is referred to [21].

The contributions of this paper are the following:

- It explores a filtering method that greatly increases the efficiency in retrieving solutions in two-objective clustering MOEAs.
- It integrates the proposed method in a scalable and flexible clustering framework.
- It tests the filtering method in a massive amount of data sets, including large ones.
- It shows a high performance in solution retrieval in both moderate and large data sets.
- It encourages practitioners to exploit the presented filtering technique to address the problem of retrieving the most suitable solutions from Pareto-based MOEAs.

The remainder of the paper is organized as follows. Section 2 briefly summarizes the related work on retrieving solutions in MC based on MOEAs. Section 3 introduces CAOS

and describes the required modifications in order to adapt it to (1) become memory scalable and (2) the new filtering method. Section 4 describes the proposed retrieval method. Section 5 describes the experimentation and discusses the results. Finally, Section 6 ends with conclusions and further work.

2. Related Work

Despite the huge popularity of MOEAs in the area of optimization due to their capabilities [7], there have been a few generic MC applications published in the literature. True MOEA-based MC algorithms did not appear until Handl and Knowles [28] introduced VIENNA. However, this algorithm needed to know the number of clusters in advance and did not provide any retrieval strategy from the Pareto set. To solve these issues, Handl and Knowles [30] proposed MOCK, the most well-known MC algorithm based on MOEAs. Another appealing approach is CAOS [10, 22] which has a flexible configuration and allows a high degree of customization. The main differences of CAOS in respect of MOCK is that the former is scalable and its retrieval step is based on several cluster validation indexes.

A key aspect in Pareto Optimum MOEAs with a posteriori approach [8] lies in the identification of the fittest solution from the Pareto set at the end of the algorithm. An intuitive approach is to aggregate all the objectives into some kind of overall metric to sort the solutions, such as predicting the relative objective weighting [38]. Nevertheless, coming up with exact relative objective weights is a daunting task with complicated ramifications [44]. Other approaches are focused on ad-hoc methods. Those identify the desirable solution according to the specific domain of the problem [41] but they are not useful when the domain is not well-known and, unfortunately, this is the case of most real-world problems. Another kind of strategies not oriented to an specific domain consist in retrieving the solution according to the shape of the Pareto set by identifying the knee region or a solution in it [4, 48, 45, 13].

In the specific case of the MC based on MOEAs, Handl and Knowles [30] proposed the use of the GAP statistic [49] to identify the most suitable solution with a good trade-off between objectives in the knee of the Pareto set. The main drawback of this technique

lies in its high computational cost when applied to large data. To overcome this issue, Matake et al. [43] following the work of Branke et al. [4] proposed a technique based on the angle between solutions to find a clustering result in the knee of the Pareto set, and it was demonstrated that this technique improved the previous results. However, these techniques do not have into account the morphological characteristics of clusters, which is related to a poor explanation capacity for each cluster. On the other hand, Handl and Knowles [29] also proposed the use of some clustering validation indexes to retrieve the solution according to the properties of the clusters instead of taking into account the shape of the Pareto set. The main problem of doing this is that the validation indexes can return a solution that only properly optimizes a single objective, so the given result does not have a good trade-off between the desired objectives.

Dealing with the aforementioned issues are of the uppermost importance in many of today's industrial and scientific applications as these have increased dramatically the amount of data used and collected. Therefore, we investigate a reliable, accurate and scalable filtering method that tackles the drawbacks of MOEAs. The proposed retrieval method is based on the observation that solutions with acceptable balance between objectives are placed around the knee of the Pareto front. The aim is to filter clustering solutions with less objective trade-off in order to retrieve the best solution from the remaining ones according to a clustering validation index. Thus, extra computations to evaluate non-interesting solutions are avoided, which is an important aspect when dealing with large data.

Our contribution is focused on obtaining a solution with a balanced trade-off among the objectives to be optimized while getting also high quality clusters. For this reason, our proposal is based on combining the use of clustering validation indexes by filtering the solutions with less balanced objectives in order to obtain competitive clustering results. Notice that we do not propose to use a knee region identification algorithm but a filtering method based on the knee of a Pareto, whose mathematical foundation can be found at [48]. The main advantage of this proposal is that it is not sensitive to the type of Pareto front (concave or convex) and to the number of knee regions. Moreover, our aim is focused on obtaining a process able to improve the performance of the retrieval step when it is applied

to large data. To carry out this, we introduce this approach into CAOS by (1) modifying its individual representation with a scalable one, and (2) modifying its learning process to work with data sampling with the aim of using less data in each evolutionary cycle. CAOS, the modifications done in it and the retrieval method are detailed in the following sections.

3. CAOS

In order to overcome some limitations of traditional clustering algorithms and to obtain high-quality clustering solutions, multiple criteria optimization is contemplated. It is focused on optimizing several objectives simultaneously by obtaining a collection of non-dominated solutions with different trade-offs among objectives called Pareto set. Recall that, in the field of multiobjective optimization, a solution S is called non-dominated when there is not a single solution better than S in regard to all the objectives. Otherwise the solution is called dominated. Thus, to obtain a final solution it is necessary to retrieve the most suitable solution from the Pareto set according to the problem to be solved. The purpose of this section is to describe CAOS, a multiobjective evolutionary algorithm specifically designed to solve clustering problems [22]. The system evolves a set of mutually non-dominated clustering solutions that correspond to different trade-offs between objectives. CAOS adopts PESA-II [9] as main basis due to its competitiveness and its ability to evolve accurate solutions from domains with complex structures [30].

In what follows, the knowledge representation used by CAOS is detailed. Next, the process organization of the algorithm is reviewed, placing special focus on the genetic operators that manipulate the representation. Finally, the data subsets method for computational scalability is depicted.

3.1. Knowledge Representation

To successfully apply MOEAs to real-world problems it is important to choose a suitable individual representation according to the problem domain, because it defines the search space where solutions will be looked for. This has motivated many works focused on the analysis and design of several representations that have demonstrated their competitiveness

[36]. CAOS uses a prototype-based representation due to (1) its search space exploring capacity and (2) its scale-up capabilities [21]. This representation is made up of real numbers which represent the coordinates of the cluster prototype (centroid) by means of its features. Therefore, each individual consists of $n \cdot t$ genes $\{x_{11}, ..., x_{1t}, ..., x_{n1}, ..., x_{nt}\}$, being n the number of clusters described by the individual, t the number of features of the data set, and x_{ij} the value of the feature j of the cluster centroid i. The genotypic representation is transformed into the phenotypic representation by assigning each instance to the cluster with the nearest centroid to it. Notice that CAOS normalizes the attribute values between 0 and 1. Several objective functions are used to validate the quality of individuals in MOEAs. These are detailed in what follows.

3.2. Objective Functions

Two complementary optimization objective functions are used to measure the quality of a solution: (1) Deviation and (2) Connectivity. These objective functions are the most widely used [30] due to the fact that they indicate how nearby are the elements of each cluster (intra-cluster variance) and how separated are the clusters between them (inter-cluster variance), respectively. Deviation assesses the intra-cluster variance and it is computed as the overall summed distances between data items and their corresponding centroid. On the other hand Connectivity refers to the inter-cluster variance and it considers the degree to which data points that are close in the feature space have been placed in the same cluster. For the sake of brevity the reader is referred to [30, 21] for more information about these objective functions.

3.3. Evolutionary Process

CAOS evolves a population of individuals through a number of generations where individuals are selected, crossed and mutated following the typical evolutionary cycle [23]. Algorithm 1 presents the CAOS algorithm. Four aspects need further explanation to fully understand the genetic process that deals with the prototype-based representation: (1) the population initialization, (2) the selection operator, (3) the crossover operator, and (4) the mutation operator.

3.3.1. Population Initialization

The population initialization step is responsible for filling the population with individuals that contain potentially worthy clusters. This approach uses an initialization based on medoids to define the initial prototypes, following the same idea as the k-means algorithm

Let EP be a external population which stores a maximum of N_{EP} individuals ;
Let IP be an internal population which stores N_{IP} individuals where $N_{IP} < N_{EP}$;
Let $\alpha_i, \forall i \in \{1, 2\}$ be the angles, set to 0 in the beginning of the run
Initialize IP with N_{IP} individuals stochastically created;
Initialize the EP individuals with non-dominated clustering results from IP;
Evaluate all the individuals from EP according to the objectives ;
foreach *Generation* **do**
    Select N_{IP} individuals from EP to generate a new IP ;
    while *(*$|IP|$ *!=* $\emptyset$*)* **do**
        Select and remove two individuals from IP;
        Cross and mutate them to obtain 2 new individuals: I_{New_1} and I_{New_2} ;
        foreach I_{New_i} **do**
            Evaluate the I_{New_i} fitness according to the objectives ;
            if I_{New_i} *dominates any individual from* EP **then**
                Remove the dominated individuals by I_{New_i} from EP;
                Add I_{New_i} into EP;
            end
            else if I_{New_i} *is not-dominated and* I_{New_i} *not-dominates any individual* **then**
                if EP *is full* **then**
                    Remove an individual from the most crowded niche;
                end
                Add I_{New_i} into EP;
            end
        end
    end
    $\alpha_i, \forall i \in \{1, 2\} \leftarrow ParetoFilter(\alpha_i, \forall i \in \{1, 2\})$ (see Algorithm 2)
end
Select a individual from EP as a solution;

Algorithm 1: Scheme of CAOS algorithm.

[31]. The process for each initial individual is the following:

(a) Select randomly a number k of clusters between a minimum and a maximum value.

(b) Generate the individual by randomly choosing k elements of the data set, where each one represents the prototype of a cluster.

3.3.2. Selection Operator

This operator selects the individuals of the population used in each iteration. The population objective space is divided into hypercubes of equal size, creating an uniform hyper-grid and so each individual is mapped to one of these hypercubes according to its objective values. To select one individual, it chooses a non-empty niche from the population and selects randomly one of the individuals mapped into the chosen niche [23].

3.3.3. Crossover operator

Crossover mixes the genetic information of the selected individuals to obtain new potential solutions. In this case, a one-point crossover operator [23, 24] is used to generate two offspring from pairs of parents. One point is selected for each parent and parts of the chromosomes are interchanged, taking into account that individuals should be cut at the same attribute but not necessarily at the same cluster.

3.3.4. Mutation Operator

Mutation modifies a piece of the genetic information of an individual in order to explore new solutions. The probability P_μ determines when this operator is applied. To mutate the individuals, a cluster-oriented mutation operator [36] is used to promote the right search. This operator defines three different types of mutations and all of them have the same probability to be applied: (1) merge two clusters, (2) split a cluster, and (3) move the centroid of a cluster. The first mutation type merges a randomly selected cluster $s1$ with its nearest cluster $s2$, adding the new cluster centroid to the individual and erasing both original clusters. The new centroid is calculated with the weighted average between the original cluster centroids and the elements of each one. The second type splits a randomly

selected cluster s in two clusters $s1$ and $s2$. $s1$ is equal to s and $s2$ is the most distant element x from s using the Euclidean distance. The last type of mutation moves the centroid of a randomly selected cluster s by adding or subtracting a $delta_j$ value to each attribute. The range of each attribute has to be between the minimum and maximum value of the corresponding attribute in the data set. If it is out of the attribute range, the value is fixed to the corresponding maximum or minimum value.

Therefore, offspring could be inconsistent individuals with empty clusters after crossover and mutation operators. These clusters are eliminated from the individual to obtain a new individual where each cluster has at least one instance assigned.

3.4. Data Sampling

MC algorithms based on MOEAs are costly in terms of computational time due to the huge amount of calculations required in large data sets, understanding computational time as the elapsed CPU time (in seconds). One way to improve the computational time is by using a subset of the available data to evaluate the individuals [2]. The main idea behind this strategy is to stratify the initial data set into disjoint data subsets (strata) of equal size and with equal class distribution [2, 6], where the number of strata is selected by the user. However, in clustering problems the strata cannot be generated because classes are unknown. To avoid this limitation in CAOS, a random strata method is used to randomly assign the instances to each one of the strata. Moreover, in order to avoid the bias produced when only one random stratum is used, strata are alternated in each iteration of the evolutionary algorithm using a round-robin strategy [3]. Thus, if the stratum is changed in each cycle, the final individuals can generalize more than using a single strata.

Notice that the definition of the number of strata will influence in the algorithm performance. As the number of strata increases the computational time of the clustering process decreases but pattern extraction becomes more complex due to the lack of information. Furthermore, it must be emphasized that the retrieval process is computed using the complete data set, and this is an issue when working with large data sets. For this reason, a competent retrieval method is a practical approach.

4. Retrieving the Most Suitable Solution

Retrieving a clustering solution based on the shape of the Pareto set can obtain a solution with a good trade-off among objectives but without any warranty about the morphological properties of clusters. This is depicted in Fig. 1, which shows (1) a Pareto front, (2) the solution retrieved according to the Pareto front shape and (3) the most suitable solution to be retrieved. In this figure, the solution identified in the knee of the Pareto front does not properly generalize. Therefore other solutions are more interesting from the point of view of the morphological properties of clusters, in order to provide useful knowledge. On the other hand, methods based only on clustering validation indexes can obtain the desirable solution according to the quality of clusters, for instance by means of compactness, but they can be sensitive to outliers and to some specific shape of clusters that are unattractive as a solution. Fig. 2 shows a Pareto front jointly with (1) the solution identified by clustering validation indexes and (2) with the most suitable solution.

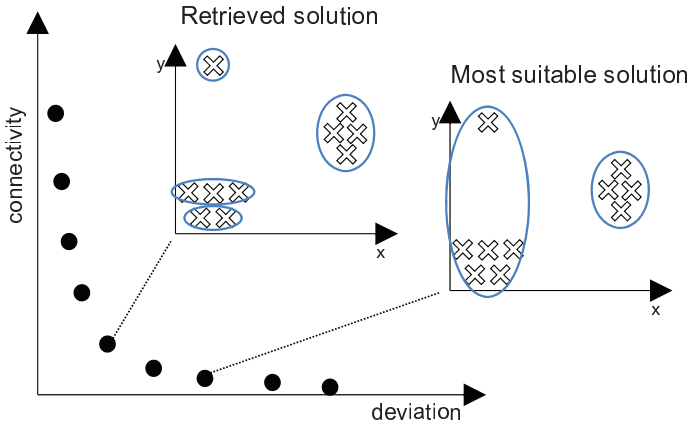


Figure 1: Pareto front representation where the bullets are several non-dominated clustering solutions. The solution retrieved using a method based on identifying the knee of the Pareto front and the most suitable solution are identified. Their corresponding clusters are detailed.

In this figure, the indexes select a solution with a bad trade-off between objectives, thus the solution given is not properly optimized and does not add any useful knowledge to

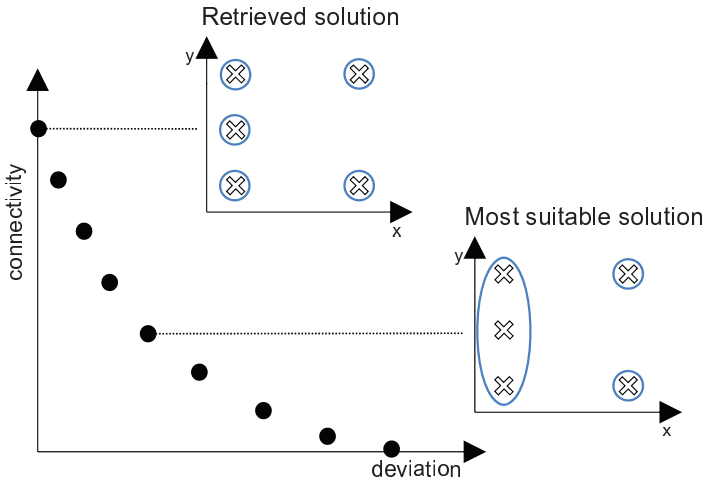


Figure 2: Pareto front representation where the bullets are several non-dominated clustering solutions. The solution retrieved with clustering validation indexes and the most suitable solution are identified. Their corresponding clusters are detailed.

experts. It must be emphasized that CAOS does not discard solutions according to the number of clusters that they contain, due to the fact that this is a subjective decision. Taking into consideration these aspects, the combination of both approaches for tackling both drawbacks can be an interesting win-win situation. The proposed hybrid approach is explained in what follows.

4.1. Retrieval Method

The proposed retrieval method filters the solutions that are in the boundaries of the Pareto set, because they barely take into account more than a single objective. Thus, solutions characterized by having very large or small clusters are discarded. The objective of the proposed technique is to apply clustering validation indexes to the remaining solutions of the Pareto set. For this reason, the indexes can obtain better results because the solutions with unbalanced objectives are discarded. The difficulty of this approach is to determine the solutions to be omitted. This issue is important because if the regions of solutions to be discarded are very large, some valuable solutions from the point of view of the quality of clusters will not be considered. On the other hand, if the regions are very small, the

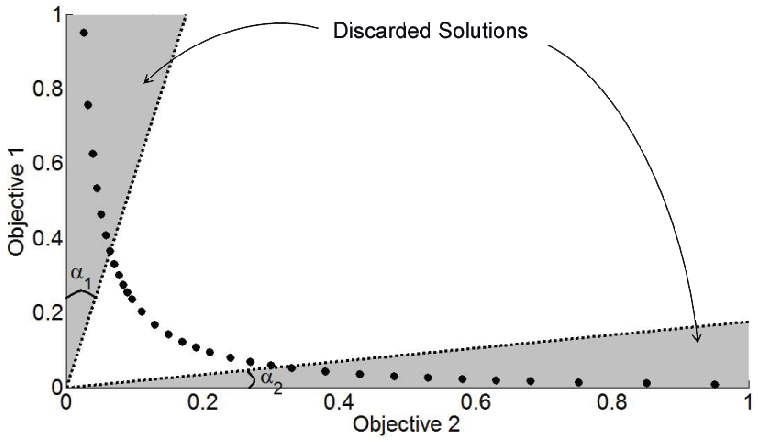


Figure 3: Graphical representation of the regions of solutions to be discarded, so that the solutions (represented by bullets) in the grey areas are discarded. α_1 and α_2 are the angles that determine the hyperplanes (dashed lines) and consequently the size of the discarded regions.

solutions that are not interesting from the point of view of clustering will be also considered. The identification of the regions to be discarded in a two-objective clustering problem is subsequently detailed.

4.2. Identification of the Solutions to Be Discarded in a Two-Objective Clustering Problem

The most useful objectives to promote the compactness and separation among clusters are Deviation and Connectivity [30] as they were described in Section 3. In two-objective optimization problems, the Pareto set can be represented in a two-dimensional graph where each axis correspond to each objective. The proposed method creates a hyperplane per objective to filter the solutions, in such a way that the regions outside the area comprised between the hyperplanes are discarded (that is, the grey areas in Fig. 3). Each hyperplane is described by one angle α_i in regard to the corresponding objective axis. Thus, the size of the discarded regions is determined by each angle. If both angles are 0 degrees, no solution is discarded. It is important to highlight that both angles would not be equal or higher than 45 degrees due to the fact that the area between the hyperplanes cannot comprise any solution of the Pareto front. The angles α_1 and α_2 are calculated in the evolutionary process. Specifically, initially they start with 0 degrees and are adjusted in

each iteration.This adjustment is calculated in two steps:

1. $\forall i \in \{1, 2\} : \ \alpha'_i \leftarrow \alpha_i + rand(\delta_{min}, \delta_{max})$, where δ_{min} and δ_{max} are two user defined parameters in the range [0, 45).

2. Compare the regions between both hyperplanes with α_i and α'_i using a quality measure estimator. If the new angles α'_i define a better region, update α_i with them.

Algorithm 2 shows the complete process of the presented filtering method, which is called in each generation of the GA as Algorithm 1 indicates. The quality of the region delimited by the hyperplanes is averaged from the quality of a random subset of the solutions contained in it. This random subset can have a maximum size of p_{max} solutions and a minimum size of p_{min}. These two parameters indicate the proportion of solutions of the Pareto set to be evaluated. If the minimum size is not achieved the current iteration does not update the angles with α'_i. An approximative and fast measure to estimate the quality of each solution is proposed. This measure takes into account the overall compactness of the clusters and the overall distance among clusters for evaluating the quality of a solution. The Estimator is calculated as Equation 1 shows, where C is the clustering obtained; n is the number of clusters; $d(x, y)$ is the Euclidean distance between the element x and y; C_i is the cluster i and ν_i is its corresponding centroid; m is the number of examples in the training data set; and t is the number of features of the instances.

$$
\begin{aligned}
Estimator(C) &= \frac{Comp(C)}{Dist(C)}, \ where \\
Comp(C) &= 1 - \frac{\sum_{i=1}^{n} \sum_{x \in C_i} d(x, \nu_i)}{m \cdot t}, \\
Dist(C) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d(\nu_i, \nu_j).
\end{aligned}
\tag{1}
$$

This process allows the system to filter the solutions during the evolutionary process without removing them from the population in order not to lose generalization capacity. It

Let $\alpha_i, \forall i \in \{1, 2\}$ be the angles that receive as parameters

Let δ_{min} and δ_{max} two user defined parameters $\in$ [0, 45)

Let Q be the overall quality estimator of the filtered solution, set to 0 in the beginning of the run

Let I_{sel} be the number of individuals selected between α_i for quality estimation

Let p_{min} be the user-defined minimum population threshold

$\alpha'_i \leftarrow \alpha_i + rand(\delta_{min}, \delta_{max}), \forall i \in \{1, 2\}$

$I_{sel} \leftarrow$ SelectIndividualsBetweenAngles(α'_i)

if *The number of individuals in* I_{sel} *is greater than* p_{min} **then**

  $Q' \leftarrow$ ComputeEstimation(I_{sel}) `//Using Equation 1`

  if $Q' > Q$ **then**

    $Q \leftarrow Q'$

    $\alpha'_i \leftarrow \alpha_i, \forall i \in \{1, 2\}$

  end

end

return $\alpha'_i, \forall i \in \{1, 2\}$

Algorithm 2: A high-level description of the proposed Pareto filter algorithm.

is worth noting that the process filters the non-interesting solutions, and it is not focused on identifying the knee region. Thus, the method is not sensitive to the type of Pareto front (concave or convex) and to the number of knee regions in it.

4.3. Clustering Validation Indexes Selection

After discarding the non-interesting clustering solutions, it is necessary to select the most suitable one from this region according to cluster properties. Therefore, clustering validation indexes are used to achieve this by using a relative criteria method [25, 27, 40], which consists in comparing all the solutions among themselves and then selecting the fittest one. In the experimentation, the most known validation indexes were integrated into the framework. Those indexes are the following: (1) Adjusted Rand index [50], (2) Davies-Bouldin index [11], (3) Dunn's index [16], (4) Silhouette index [46] and (5) Calinski-Harabasz index [5].

Adjusted Rand Index is the supervised index of reference used. It retrieves the clustering solution from the Pareto set regarding to the original classes of the problem, it returns values between 0 and 1 and it should be maximized. Specifically, it compares two clustering results (the original one and the proposed as solution) counting the number of pairwise co-assignments of instances between them and introducing a statistically induced normalization in order to yield values close to 0 for random partitions (see Equation 2). In the equation, n is the number of clusters of the evaluated solution C, n_o is the number of the original classes of the data set O, m is the number of instances of the data set, m_{ij} is the number of data items that have been assigned to both class i and cluster j, $m_{i.}$ is the number of instances assigned to class i and $m_{.j}$ is the number of instances assigned to cluster j. The other four indexes are based on inherent information of the data set in order to obtain a solution with clusters of high quality. Each one of these indexes makes different calculations and they can return a different clustering solution from the collection of potential solutions, so the use of one index or another depends on the point of view of the expert. Having explained in detail the intrinsics of CAOS, in the next section, the different strategies are analyzed in a variety of experiments.

$$R(C,O) = \frac{\sum_{i=1}^{n_o}\sum_{j=1}^{n}\binom{m_{ij}}{2} - \left[\sum_{i=1}^{n_o}\binom{m_{i.}}{2}\cdot\sum_{j=1}^{n}\binom{m_{.j}}{2}\right]\Big/\binom{m}{2}}{\frac{1}{2}\left[\sum_{i=1}^{n_o}\binom{m_{i.}}{2}+\sum_{j=1}^{n}\binom{m_{.j}}{2}\right] - \left[\sum_{i=1}^{n_o}\binom{m_{i.}}{2}\cdot\sum_{j=1}^{n}\binom{m_{.j}}{2}\right]\Big/\binom{m}{2}} \quad (2)$$

5. Experiments, Results and Discussion

This section analyzes the performance of the retrieval strategies to select the most suitable solution using CAOS. First, 35 artificial data sets and 35 real-world data sets are analyzed. Specifically, the proposed filtering method is compared with respect to the technique presented by Matake [43] that is based on the shape of the Pareto set, and with respect

to other strategies based on using clustering validation indexes to assess cluster quality. The technique based on adjacent angles proposed by Matake returns a solution in the knee of the Pareto front and has demonstrated a high degree of competitiveness. Moreover, another series of experiments applied to large data are performed using the same methodology in order to analyze the approaches behavior in this kind of data. In what follows, the experimental methodology and the results of the comparison are presented and discussed.

5.1. Experimental Methodology

This section presents the experimental methodology followed in order to evaluate the performance of the different retrieval strategies to select the most suitable solution from the Pareto set found by CAOS. The analysis enables us to emphasize the benefits and the drawbacks of each one. In the followings, we provide details about (1) the data set collection chosen for the experimentation, (2) the CAOS configuration, and (3) the comparison metrics.

Test Bed. The experimentation is divided into two kinds of experiments. The first kind is oriented to non-large data sets and assess the algorithm performance using different typologies of artificial and real-world problems (see Table 1). First, 35 artificial data sets were selected according to different number of instances (from 900 to 2990), attributes (from 2 to 100) and classes (from 2 to 10). They were built using the tool presented by Handl and Knowles [30]. Also, 35 real-world problems were selected according to different number of instances (from 101 to 7494), attributes (from 3 to 60) and classes (from 2 to 11). The second kind of experiments uses large data for assessing the algorithms performance (see Table 2). Specifically, it uses 6 data sets with a number of instances between 19000 to 581012, a number of attributes from 9 to 54, and a number of classes between 2 and 26. All these data sets were obtained from the UCI [18], KEEL [1] and KDD [33] repositories.

CAOS Configuration. CAOS was run with 50 different random seeds with the synthetic and the real-world problems and with 20 different random seeds with the large data sets. The system was configured using the following parameters (the author is referred to [22] for notation details): ℓ was 5% of the number of data set instances, the maximum size of the

Data set	nI	nA	nC	Data set	nI	nA	nC
100d-10c	2198	100	10	appendicitis	106	7	2
100d-4c	1218	100	4	balance	625	4	3
10d-10c	2122	10	10	biopn	1027	24	2
10d-4c	1092	10	4	bpa	345	6	2
2d-10c	2990	2	10	contraceptives	1473	9	3
2d-4c	1261	2	4	crx	690	15	2
curves1	1000	2	2	dermatology	366	35	6
curves2	1000	2	2	echocardiogram	132	12	2
dartboard1	1000	2	4	ecoli	336	8	8
dartboard2	1000	2	4	glass	214	9	6
donut1	1000	2	2	haberman	306	3	2
donut2	1000	2	2	heart-statlog	270	13	2
donut3	999	2	3	hepatitis	155	19	2
donutcurves	1000	2	4	housevotes	435	16	2
long1	1000	2	2	ionosphere	351	34	2
long2	1000	2	2	iris	150	4	3
long3	1000	2	2	liver-disorders	345	6	2
longsquare	900	2	6	mammographic	961	5	2
sizes1	1000	2	4	pendigits	7494	17	10
sizes2	1000	2	4	pim	768	8	2
sizes3	1000	2	4	segment	2310	19	7
sizes4	1000	2	4	sonar	208	60	2
sizes5	1000	2	4	tae	151	5	3
smile1	1000	2	4	thyroids	215	5	2
smile2	1000	2	4	transfusion	748	4	2
smile3	1000	2	4	vehicle	846	18	4
spiral	1000	2	2	vertebral	310	6	3
spiralsquare	1500	2	6	vowel	990	13	11
square1	1000	2	4	waveform	5000	40	3
square2	1000	2	4	wdbc	569	30	2
square3	1000	2	4	wine	178	13	3
square4	1000	2	4	wisconsin	699	9	2
square5	1000	2	4	wpbc	198	33	2
triangle1	1000	2	4	yeast	1484	9	10
triangle2	1000	2	4	zoo	101	16	7

Table 1: Summary of the characteristics of the 35 artificial data sets (left block) and real-world data sets (right block) used. The columns of each block are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC).

initial population was 100, N_{EP} was 1000, N_{IP} was 50, N_{niches} was 5, the number of generations was 400, the probability of crossover (P_c) was set to 0.7 and the probability of mutation (P_μ) was set to $1/m$. The filtering method was configured with the next parameters: δ_{min} was 0.1, δ_{max} was 0.75, p_{min} was 50% of the numbers of solutions in the Pareto set and p_{max} was 10% of them. As we are interested in robust systems that perform competently on average, the same configuration was used for all the data sets. To set these parameters to their optimal values, the iterated F-Race procedure [42] was followed. Moreover, the experiments done with large data sets use data sampling as Section 3.4 explains. Each data set has been divided in four strata (i.e., each stratum contains a 25% of the instances of the original data set). The reader is referred to [2] for more information about this issue.

Retrieval Strategies Analyzed. The proposed filtering technique was applied with some of the most used clustering validation indexes such as Davies, Dunn, Silhouette and Calinski-

Data set	nI	nA	nC	Data set	nI	nA	nC
covtype	581012	54	7	letter	20000	16	26
kddcup	494021	41	23	magic	19022	10	2
census	299324	41	2	2d-20c-125m	16097	2	20
shuttle	58000	9	7	5d-20c-175m	15675	5	20
10d-30c-175m	23898	10	30	20d-20c-125m	15508	20	20
10d-30c-75m	23471	10	30	2d-20c-75m	15012	2	20
5d-30c-75m	23234	5	30	20d-20c-175m	14970	20	20
100d-30c-175m	22788	100	30	10d-20c-75m	14830	10	20
20d-30c-75m	22470	20	30	20d-20c-75m	14491	20	20
2d-30c-175m	22229	2	30	5d-20c-125m	14261	5	20
5d-30c-125m	22038	5	30	10d-20c-175m	14023	10	20
10d-30c-125m	21974	10	30	10d-20c-125m	13875	10	20
2d-30c-125m	21846	2	30	100d-20c-75m	13790	100	20
20d-30c-175m	21491	20	30	100d-20c-125m	13702	100	20
5d-30c-175m	21129	5	30	100d-20c-175m	13421	100	20
20d-30c-125m	20986	20	30	2d-20c-175m	13355	2	20
2d-30c-75m	20370	2	30	5d-20c-75m	13289	5	20
100d-30c-125m	20156	100	30				

Table 2: Summary of the characteristics of the 35 large data sets used. The columns of each block are referred to the number of instances (nI), to the number of attributes (nA) and to the number of classes (nC).

Harabasz. Next, these results were compared with the ones obtained with the same clustering validation indexes and the adjacent angles approach using the overall Pareto set. In addition to these strategies, we also contemplated the best solution from the overall Pareto set according to the Adjusted Rand index [50]. It must be emphasized that the Adjusted Rand index is based on obtaining the best solution according to a prespecified structure of the data set, in our case, the classes assigned to each instance–that are known in benchmark problems–. This strategy is used to compare our proposal with the ideal solution.

Comparison Metrics. The accuracy of each solution was quantified using the Adjusted Rand index in order to evaluate them according to the original classes of the problems. The recommendations pointed out by [14] were followed to perform the statistical analysis of the accuracy results, which is based on the use of nonparametric tests. More specifically, the following methodology was employed. First, the Friedman test [20] was applied to contrast the null hypothesis that all the learning algorithms obtained the same results on average. If the Friedman test rejects the null hypothesis, we perform pair-wise comparisons by means of the Holm's step-down procedure [34]. Following this procedure, we distinguish pairs of retrieval strategies that are significantly different in performance.

5.2. Massive Comparison in Non-Large Data Sets

The analysis of the performance among all the strategies using the overall Pareto set and the proposed filtering method was carried out with all the presented data sets. Table 3 shows the results using a pairwise comparison by means of Holm's procedure. In it, the strategy used to retrieve the most suitable solution is indicated by Dv, Dn, Sl, CH for the Davies, Dunn, Silhouette and Calinski-Harabasz indexes respectively. Also, the symbols of each strategy are preceded by an F when the filtering method is used and by an A when the overall Pareto set is used. Moreover, AA indicates the adjacent angles strategy and AR the supervised solution retrieved with the Adjusted Rand index, which only takes into account the overall data set. The symbols $\oplus$ and $\ominus$ show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column at $\alpha = 0.05$. Similarly, the symbols $+$ and $-$ denote a non-significant higher/lower

	AR	AA	ADv	FDv	Friedman
AR					1.29
AA	$\ominus$				3.26
ADv	$\ominus$	$-$			3.91
FDv	$\ominus$	$\oplus$	$+$		2.54

(a)

	AR	AA	ADn	FDn	Friedman
AR					1.22
AA	$\ominus$				3.05
ADn	$\ominus$	$-$			3.06
FDn	$\ominus$	$+$	$+$		2.65

(b)

	AR	AA	ASl	FSl	Friedman
AR					1.28
AA	$\ominus$				3.24
ASl	$\ominus$	$-$			2.97
FSl	$\ominus$	$\oplus$	$+$		2.51

(c)

	AR	AA	ACH	FCH	Friedman
AR					1.24
AA	$\ominus$				3.19
ACH	$\ominus$	$-$			2.88
FCH	$\ominus$	$+$	$+$		2.70

(d)

Table 3: Pairwise comparison of all the strategies in non-large data sets in respect of (a) Davies index, (b) Dunn's index, (c) Silhouette index and (d) Calinski-Harabasz index. Dv, Dn, Sl, CH represent the results of the Davies, Dunn, Silhouette and Calinski-Harabasz indexes respectively. Also, the symbols of each strategy are preceded by an F when the filtering method is used and by an A when the overall Pareto set is used. Moreover, AA indicates the adjacent angles strategy and AR the supervised solution retrieved with the Adjusted Rand index, which only takes into account the overall data set. The symbols $\oplus$ and $\ominus$ show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column at $\alpha = 0.05$. Similarly, the symbols $+$ and $-$ denote a non-significant higher/lower results. The last column shows the Friedman rank, where the minimum value indicates the best rank.

results. Likewise, the Friedman rank value of each method is indicated, where lower values are better.

Table 3 summarizes the results obtained with the proposed filtering method (FDv, FDn, FSl and FCH) and the results obtained according only to (1) the morphological characteristics of clusters (ADv, ADn, ASl and ACH), (2) the shape of the Pareto front (AA) and (3) the supervised solution (AR). It can be observed that the solutions obtained with the filtered method are better ranked in terms of accuracy than the solutions obtained using the morphological properties of clusters from the overall Pareto set. Moreover, the filtered method solutions are better, and in some cases significantly better (i.e., Davies and Silhouette indexes) than the solution which only takes into account the shape of the Pareto front. In regard to the supervised solution, it is obvious that it is always significantly better than the other strategies due to the fact that it considers the original classes of the data set.

Furthermore, the proposed filtering method improves the computational time of the retrieval step due to the fact that the non-interesting solutions are not analyzed. This is depicted in Fig. 4, where the computational time of the retrieval step is shown for both filtered and overall Pareto set for each data set. We considered as retrieval step the process that selects the solutions according to each one of the strategies for each method. Particularly, the computational time of the retrieval step is calculated as the average time of all the strategies. Recall that filtering methods are always faster than the non-filtering ones and, in some data sets, the speedup can become faster in an order of magnitude. It is worth noting that the time required for computing the hyperplanes of the filtering method is not expensive. The evolutionary cycle only adds in average an extra 4.12%$\pm$4.73 of computational cost but, in return, the retrieval step time is reduced on average a 89.53%$\pm$7.79. In respect of the angles needed for building the hyperplanes, on average they take small values (α_1=3.82$\pm$2.86 and α_2=3.57$\pm$2.94). Thus, not a huge quantity of solutions is discarded, just only the solutions that are in the extremes of the Pareto front.

An interesting observation lies in the result of applying the filtering technique to Pareto fronts with concave shapes or with discontinuities. Because this method does not assume a particular shape or continuity in the Pareto front, and because it only filters non-interesting

solutions, it can be safely applied to any kind of MC problem. Fig. 5 shows four cases of Pareto fronts with these features. It can be observed that the filtered regions do not consider the solutions with a bad trade-off among objectives. Thus, they help clustering validation indexes to avoid the problem of obtaining solutions far from the knee of the Pareto front. It is important to highlight that in the Pareto front there are not solutions with a very high value of the Deviation objective due to the fact that the genetic operators defined in the evolutionary algorithm tend to obtain a reasonable number of clusters. For example, the maximum value of the Deviation is achieved if all the elements of the data set are in a different cluster. Thus, hyperplane in the Deviation objective area filters few solutions in comparison with the hyperplane of the Connectivity area.

5.3. Comparison in Large Data Sets

In order to analyze the performance of the presented method with large data, similar experiments to the ones in the previous section are carried out but using large data sets. Table 4 summarizes the results obtained by means of the Holm's procedure using the aforementioned nomenclature. It can be observed that solutions obtained with the filtered method are better ranked in terms of accuracy than the solutions obtained using the clustering validation indexes from the overall Pareto set. Moreover, the majority of the filtered results are not significantly different than the supervised results. Nevertheless, they are slightly behind the ones of the strategy based only in the shape of the Pareto set.

In terms of retrieval time, as it is depicted in Fig. 6, the proposed filtering method highly improves the computational time, on average, in three orders of magnitude, being reduced in a 98.79%$\pm$0.03. It is worth mentioning that the time required for computing the hyperplanes slightly increments the time of the evolutionary algorithm in 8.43%$\pm$6.07. Regarding to the angles needed for building the hyperplanes, they take small values on average (α_1=4.51$\pm$2.00 and α_2=4.98$\pm$2.14), so only the solutions with a bad trade-off between objectives are discarded.

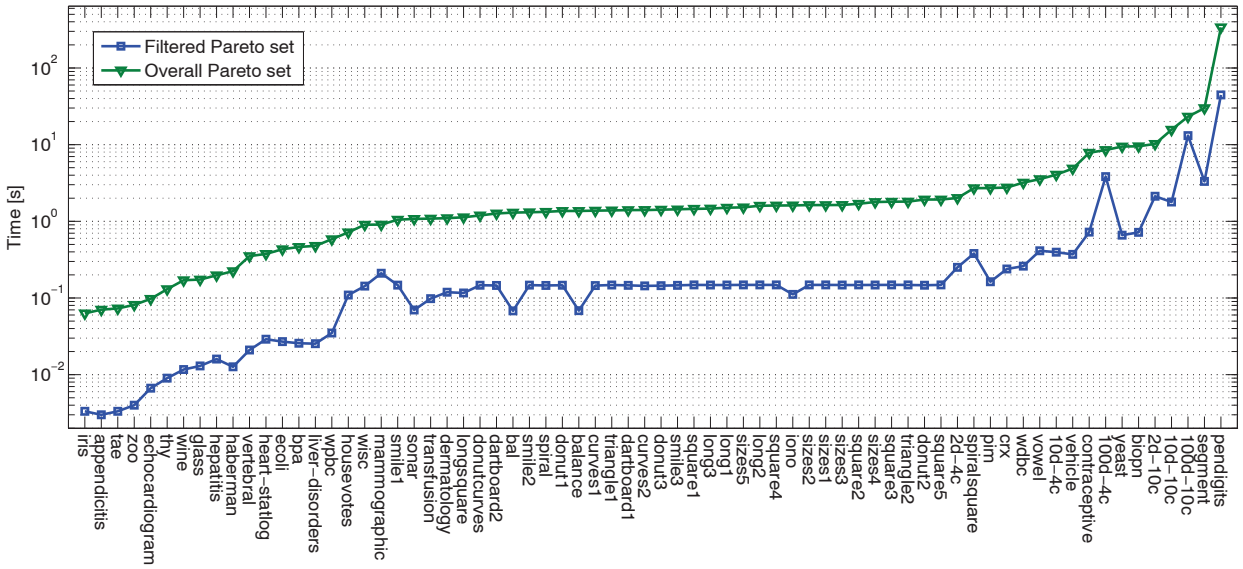
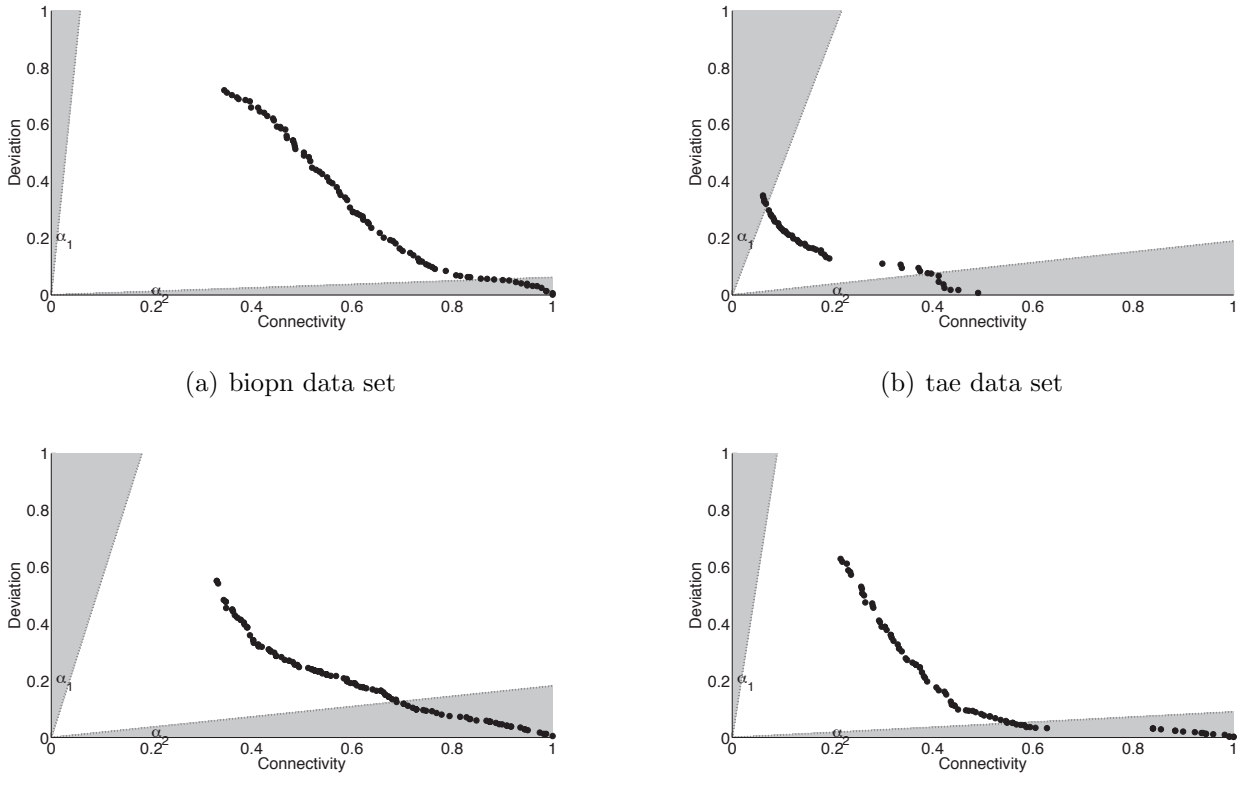


Figure 4: Average computational time of the retrieval step in seconds for each one of the non-large data sets of (1) filtered Pareto sets retrieval strategies and (2) overall Pareto set retrieval strategies. Notice the logarithmic scale of time axis. Results are averages of ten runs.



(a) biopn data set

(b) tae data set

(c) wpbc data set

(d) wdbc data set

Figure 5: Examples of the filtering method applied to problems with a complex Pareto front. These examples come from (a) the biopsia problem, (b) the tae problem, (c) the wisconsin problem and (d) the wdbc problem.

	AR	AA	ADv	FDv	Friedman
AR					1.16
AA	$\ominus$				2.94
ADv	$\ominus$	$-$			3.37
FDv	$\ominus$	$+$	$\oplus$		2.53

(a)

	AR	AA	ADn	FDn	Friedman
AR					1.17
AA	$\ominus$				2.93
ADn	$\ominus$	$-$			3.31
FDn	$\ominus$	$+$	$+$		2.59

(b)

	AR	AA	ASl	FSl	Friedman
AR					1.13
AA	$\ominus$				3.04
ASl	$\ominus$	$-$			3.17
FSl	$\ominus$	$+$	$+$		2.66

(c)

	AR	AA	ACH	FCH	Friedman
AR					1.13
AA	$\ominus$				3.04
ACH	$\ominus$	$-$			3.17
FCH	$\ominus$	$+$	$+$		2.66

(d)

Table 4: Pairwise comparison of all the strategies in large data sets in respect of (a) Davies index, (b) Dunn's index, (c) Silhouette index and (d) Calinski-Harabasz index. Dv, Dn, Sl, CH represent the results of the Davies, Dunn, Silhouette and Calinski-Harabasz indexes respectively. Also, the symbols of each strategy are preceded by an F when the filtering method is used and by an A when the overall Pareto set is used. Moreover, AA indicates the adjacent angles strategy and AR the supervised solution retrieved with the Adjusted Rand index, which only takes into account the overall data set. The symbols $\oplus$ and $\ominus$ show that the method in the row obtained results that were significantly higher/lower than those obtained with the method in the column at $\alpha = 0.05$. Similarly, the symbols $+$ and $-$ denote a non-significant higher/lower results. The last column shows the Friedman rank, where the minimum value indicates the best rank.

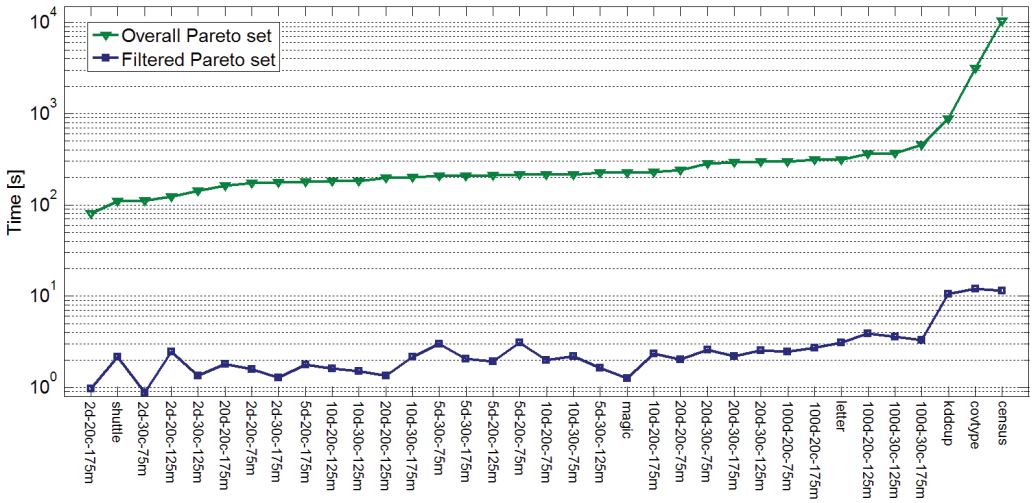


Figure 6: Average computational time of the retrieval step in seconds for each one of the large data sets of (1) filtered Pareto sets retrieval strategies and (2) overall Pareto set retrieval strategies. Notice the logarithmic scale of time axis. Results are averages of ten runs.

6. Conclusions and Further Work

The solution returned by a Pareto-based MOEA is a Pareto set of non-dominated solutions in which none of those solutions can be further improved on any objective without degrading the other ones. Although there is not a winner solution according to all the optimizing objectives, the most suitable solution to solve a specific problem can be manually retrieved with the help of an expert. This has motivated the necessity of proposing methods for automatically retrieving the most suitable solution, specially in the case of large volumes of data. In the case of MC, these methods usually select the solution in regard to (1) the shape of the Pareto set, which correspond to the value of the objectives to optimize, or (2) the quality of the solutions conforming to specific characteristics of the problem. The main drawback of the first method is that it retrieves a solution without taking into account the morphological characteristics of clusters and it can return a solution with a good trade-off between objectives but with poor quality clusters (i.e., non useful for expert proposals). On

the other hand, the second method retrieves a solution according to the quality and shape of clusters using clustering validation indexes but it does not consider the value of the objectives, so it can return a solution with an inadequate trade-off between them. For these reasons, we proposed the combination of both methods to obtain a new hybrid mechanism which filters and selects a solution according to a clustering validation index from the region of the Pareto set where all the solutions with a good trade-off between objectives are placed. Moreover, this filtering technique can be applied to any kind of Pareto-based MOEA.

The proposed filtering method was analyzed using several clustering validation indexes in both large and non-large data sets. Traditional approaches were also included in the analysis in order to compare the results. To carry out the experimentation, CAOS algorithm was used to build the Pareto set with clustering solutions. Experiments show that, in the case of non-large data sets, the proposed filtering technique is the most accurate and the one that requires less computation. Furthermore, the proposed method can obtain solutions that are not significantly different to the solutions retrieved by a supervised method, so they work as well as a method that uses the classes of the problem to retrieve the best solution. In the case of large data, the results highlight a huge improvement in the retrieval step time without losing generalization capacity, demonstrating that the proposed technique is memory scalable and useful to tackle large data sets. It must be emphasized that the performance of the filtering technique does not depend on the way that the Pareto set is built, so the obtained results are not influenced by the CAOS algorithm.

Clustering is not focused on classifying a data set according to a specified structure and, consequently, the morphological properties of the obtained clusters are key to understand the proposed patterns. It is for this reason that the solutions retrieved with clustering validation indexes consider this issue and if they are obtained from the filtered Pareto set, the solutions consider an acceptable trade-off between objectives–the aim of MC.

As future work we are working on analyzing the effects of using other retrieval strategies and the application of the filtering technique to Pareto sets with more than two objectives.

References

[1] Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., 2011. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Multiple-Valued Logic and Soft Computing 17 (2-3), 255–287.

[2] Bacardit, J., october 2004. Pittsburgh genetic-based machine learning in the data mining era: Representations, generalization, and run-time. Ph.D. thesis, Enginyeria i Arquitectura La Salle, Universitat Ramon Llull, Barcelona (Spain).

[3] Bacardit, J., Llorà, X., 2009. Large scale data mining using genetics-based machine learning. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers. GECCO '09. ACM, pp. 3381–3412.

[4] Branke, J., Deb, K., Dierolf, H., Osswald, M., 2004. Finding knees in multi-objective optimization. In: 8th Conference on Parallel Problem Solving from Nature (PPSN VIII). Lecture Notes in Computer Science. Springer-Verlag, pp. 722–731.

[5] Caliński, T., Harabasz, J., 1974. A dendrite method for cluster analysis. Communications in Statistics Simulation and Computation 3 (1), 1–27.

[6] Cano, J. R., García, S., Herrera, F., 2008. Subgroup discovery in large size data sets preprocessed using stratified instance selection for increasing the presence of minority classes. Pattern Recognition Letters 29, 2156–2164.

[7] Coello, C. A., 1999. A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowledge and Information Systems 1, 269–308.

[8] Coello, C. A., Lamont, G. B., Veldhuizen, D. A. V., 2007. Evolutionary algorithms for solving multi-objective problems. Springer-Verlag New York, Inc.

[9] Corne, D. W., Jerram, N. R., Knowles, J. D., Oates, M. J., 2001. PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann Publishers, pp. 283–290.

[10] Corral, G., Garcia-Piquer, A., Orriols-Puig, A., Fornells, A., Golobardes, E., 2011. Analysis of Vulnerability Assessment Results based on CAOS. Applied Soft Computing Journal 11, 4321–4331.

[11] Davies, D., Bouldin, D., 1979. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (4), 224–227.

[12] Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Ltd, England.

[13] Deb, K., Gupta, S., 2011. Understanding knee points in bicriteria problems and their implications as preferred solution principles. Engineering Optimization 43 (11), 1175–1204.

[14] Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine

Learning Research 7, 1–30.

[15] Duda, R. O., Hart, P. E., Stork, D. G., 2000. Pattern classification. John Wiley and Sons, Inc., New York.

[16] Dunn, J. C., 1974. Well separated clusters and optimal fuzzy partitions. In: Journal of Cybernetics. Vol. 4. pp. 95–104.

[17] Fonseca, C. M., Fleming, P. J., 1995. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 3, 1–16.

[18] Frank, A., Asuncion, A., 2010. UCI machine learning repository.
URL `http://archive.ics.uci.edu/ml`

[19] Freitas, A. A., 2002. Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[20] Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. Annals of Mathematical Statistics 11, 86–92.

[21] Garcia-Piquer, A., Fornells, A., Bacardit, J., Orriols-Puig, A., Golobardes, E., 2014. Large-scale experimental evaluation of cluster representations for multiobjective evolutionary clustering. IEEE Transactions on Evolutionary Computation 18 (1), 36–53.

[22] Garcia-Piquer, A., Fornells, A., Orriols-Puig, A., Corral, G., Golobardes, E., 2012. Data Classification through an Evolutionary Approach Based on Multiple Criteria. Knowledge and Information Systems 33 (1), 35–56.

[23] Goldberg, D. E., 1989. Genetic Algorithm in Search, Optimization, and Machine Learning. Addison-Wesley, Inc., Boston, MA, USA.

[24] Goldberg, D. E., 2002. The Design of Innovation. Kluwer Academic Publishers, Massachusetts, US.

[25] Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2001. On clustering validation techniques. Journal of Intelligent Information Systems 17, 107–145.

[26] Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2002. Cluster validity methods: Part i. SIGMOD Record 31 (2), 40–45.

[27] Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2002. Clustering validity checking methods: Part ii. SIGMOD Record 31 (3), 19–27.

[28] Handl, J., Knowles, J., 2004. Evolutionary multiobjective clustering. Lecture notes in computer science, 1081–1091.

[29] Handl, J., Knowles, J., Sept 2005. Improvements to the scalability of multiobjective clustering. In: Evolutionary Computation, 2005. The 2005 IEEE Congress on. Vol. 3. pp. 2372–2379.

[30] Handl, J., Knowles, J., 2007. An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation 1 (1), 56–76.

[31] Hartigan, J., Wong, M., 1979. A k-means clustering algorithm. In: Applied Statistics. pp. 28:100–108.

[32] Herrera, F., Carmona, C., Gonz·lez, P., del Jesus, M., 2010. An overview on subgroup discovery: foundations and applications. Knowledge and Information Systems 10.1007/s10115-010-0356-2.

[33] Hettich, S., Bay, S. D., 1999. The UCI KDD archive university of california.
URL `http://kdd.ics.uci.edu`

[34] Holm, S., 1979. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6, 65–70.

[35] Hruschka, E., Campello, R. J. G. B., Freitas, A., De Carvalho, A. C. P. L. F., 2009. A survey of evolutionary algorithms for clustering. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39 (2), 133–155.

[36] Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., de Carvalho, A. C. P. L. F., 2009. A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 39 (2), 133–155.

[37] Iredi, S., Merkle, D., Middendorf, M., 2000. Bi-criterion optimization with multi colony ant algorithms. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001). Springer, pp. 359–372.

[38] Kasprzak, E. M., Lewis, K. E., 2001. Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method. Structural and Multidisciplinary Optimization 22 (3), 208–218.

[39] Kaufman, L., Rousseeuw, P. J., 1990. Finding groups in data: An introduction to cluster analysis. John Wiley & Sons.

[40] Leg·ny, C., Juh·sz, S., Babos, A., 2006. Cluster validity measurement techniques. In: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases. pp. 388–393.

[41] Liu, Y., Yoshioka, M., Homma, K., Shibuya, T., 2009. Efficiently finding the 'best' solution with multi-objectives from multiple topologies in topology library of analog circuit. In: Proceedings of the 2009 Asia and South Pacific Design Automation Conference. ASP-DAC '09. IEEE Press, pp. 498–503.

[42] López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M., 2011. The irace Package: Iterated Racing for Automatic Algorithm Configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium.

[43] Matake, N., Hiroyasu, T., Miki, M., Senda, T., 2007. Multiobjective clustering with automatic k-determination for large-scale data. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM, pp. 861–868.

[44] Messac, A., Sundaraj, G. J., Tappeta, R. V., Renaud, J. E., 2000. Ability of objective functions to generate points on nonconvex Pareto frontiers. AIAA Journal 38 (6), 1084–1091.

[45] Rachmawati, L., Srinivasan, D., 2009. Multiobjective evolutionary algorithm with controllable focus on the knees of the pareto front. IEEE Transactions on Evolutionary Computation 13 (4), 810–824.

[46] Rousseew, P. J., 1987. Silhouttes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational Applications in Math 20, 53–65.

[47] Saha, S., Bandyopadhyay, S., 2010. A new multiobjective clustering technique based on the concepts of stability and symmetry. Knowledge and Information Systems 23, 1–27.

[48] Schütze, O., Laumanns, M., Coello, C. A., 2008. Approximating the knee of an mop with stochastic search algorithms. In: Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X. Springer-Verlag, Berlin, Heidelberg, pp. 795–804.

[49] Tibshirani, R., Walther, G., Hastie, T., 2000. Estimating the number of clusters in a dataset via the gap statistic. Journal of the Royal Statistical Society 63, 411–423.

[50] Yeung, K., Ruzzo, W., 2001. Details of the adjusted rand index and clustering algorithms. supplement to the paper "an empirical study on principal component analysis for clustering gene expression data". Science 17 (9), 763–774.

[51] Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation 8 (2), 173–195.

Figure