2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

# A Co-Evolutionary Multi-Objective Approach for a K-Adaptive Graph-based Clustering Algorithm

Héctor D. Menéndez David F. Barrero David Camacho

***Abstract*—Clustering is a field of Data Mining that deals with the problem of extract knowledge from data blindly. Basically, clustering identifies similar data in a dataset and groups them in sets named clusters. The high number of clustering practical applications has made it a fertile research topic with several approaches. One recent method that is gaining popularity in the research community is Spectral Clustering (SC). It is a clustering method that builds a similarity graph and applies spectral analysis to preserve the data continuity in the cluster. This work presents a new algorithm inspired by SC algorithm, the Co-Evolutionary Multi-Objective Genetic Graph-based Clustering (CEMOG) algorithm, which is based on the Multi-Objective Genetic Graph-based Clustering (MOGGC) algorithm and extends it by introducing an adaptative number of clusters. CEMOG takes an island-model approach where each island keeps a population of candidate solutions for $k_i$ clusters. Individuals in the islands can migrate to encourage genetic diversity and the propagation of individuals around promising search regions. This new approach shows its competitive performance, compared to several classical clustering algorithms (EM, SC and K-means), through a set of experiments involving synthetic and real datasets.**

## I. Introduction

Clustering is a field of Data Mining whose goal is to group data in entities named clusters. It exploits hidden similarities in collections of data. The large corpus of literature in this field is aligned with the high number of applications of clustering in several domains such as Biomedicine [1], marketing [2], image segmentation [3] and virtual worlds [4] amongst others.

There are several approaches for clustering. The classical ones are K-means [5] and Expectation Maximization (EM) [6]. These two algorithms come from Statistics, and both build a cluster model based on the assumption of an underlyng statistical distribution of the data. If this assumption is incorrect, or there is no knowledge about the model, the parametric clustering methods fail.

Non-parametric clustering comes to overcome the limitations of parametric clustering. Among the different non-parametric methods, Spectral Clustering (SC) [7] is revealing as one promising approach. SC represents data as a graph where each node is an instance and each edge represents the similarity among two data, then SC groups data in the graph by means of spectral analysis. SC has several problems related to its robustness and graph storage [8].

Héctor Menéndez and David Camacho are with the Computer Science Department, Escuela Politécnica Superior of Universidad Autónoma de Madrid. David F. Barrero is with the Departamento de Automática, Universidad de Alcalá. Spain (email: {hector.menendez, david.camacho}@uam.es, david@aut.uah.es).

This work has been partly supported by: Spanish Ministry of Science and Education under project TIN2010-19872.

Clustering has taken much benefit of using Evolutionary Algorithms (EAs). Clustering is essentially a search problem of a function that maps data into clusters, just the type of problem that EAs handle well. Not surprisingly there are several approaches of evolutionary clustering [9].

In a previous work, we proposed a Genetic Graph-based Clustering algorithm (GGC) [8]. It combines the classical K-Nearest Neighbourhood (KNN) algorithm and the Minimal Cut measure to search the best cut of the graph. Then we extended GGC with the Multi-Objective Genetic Graph-based Clustering Algorithm (MOGGC) [10]. This extension introduces a Multi-Objective Genetic Algorithm [11] with graph-continuity metrics to achieve lower memory consumption and increased solution quality. The main drawback of MOGGC is the need of a priori knowledge about the number of clusters, k, which limits the applications of the algorithm.

This paper presents the Co-Evolutionary Multi-Objective Genetic Graph-based Clustering (CEMOG) algorithm. The contribution of CEMOG is the development of a new partitional clustering algorithm that solves the k-determination problem of MOGGC. To this end, CEMOG uses coevolution to simulate variable-length chromosomes in a Genetic Algorithm. In this way, the value of k is introduced in the evolutionary search and eventually the Pareto front provide a set of k corresponding to the trade-off of the solutions.

The paper is structured as follows. First we introduce the related work, section 3 describes in detail CEMOG followed by the experimental evaluation in section 4. Finally, the last section summarizes the conclusions and future work.

## II. Related Work

Over the last years evolutionary clustering has attracted much research interest, yielding a large literature corpus. Evolutionary Computation is a vast field that includes many families of algorithms, all of them inspired in natural selection. Perhaps the most popular EA for clustering is Genetic Algorithms (GAs), where a population of candidate solutions is codified in strings named chromosomes. Then GAs apply a set of genetic operators (typically mutation and crossover) and a stochastic selection operator based on a fitness function to breed the next algorithm iteration. Hruschka et al. [9] presents a complete survey on this topic.

Another approach to evolutionary clustering with GAs comes from Multi-Objective Genetic Algorithm (MOGA). In this approach, the selection of the individual does not depend on one criteria, but several ones. Most of the approaches to multiobjective evolutionary clustering use, with

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

different names, inter and intra cluster distances, i.e., they try to minimize the distance between data and their cluster centroids, while maximizing the distance among the clusters centroids. Some authors claim the superiority of this approach, for instance, Ripon and Kwong [12] stated that tradicional single objective algorithms suffer premature convergence that multiobjective algorithms solve. It is clear that sometimes using a single criteria loses important pieces of information that would be exploited in benefit of the search.

There are some proposals of MOGAs with adaptative number of clusters. Handl et al. proposed the Multi-Objective Clustering with automatic K-determination (MOCK) [13], a graph-oriented clustering algorithm on a MOGA. In this approach the chromosomes represent non-weighted graphs with an integer representation. Each loci represents a data instance and the allele a link to another instance. With this representation, a cromosome may contain several subgraphs, i.e., graphs without links to other graphs. These isolated subgraphs represent the clusters. Despite the graph-based representation, this approximation cannot be considered spectral clustering because of the lack of spectral analysis. Matake et al. proposed an improvement of MOCK [14] to compute k more efficiently and make the algorithm well suited for large datasets.

Another example of adaptative k multiobjective clustering algorithm is the Variable-Length Real Jumping Genes Genetic Algorithm (VRJGGA), proposed by Ripon et al. [12]. VRJGGA is an adaptative version of another algorithm named JGGA. It uses a Variable-Length Genetic Algorithm with a standard cluster centroid representation in a chromosome of floats. The variance of chromosome lengths is introduced with two custom genetic operators: cut-and-paste and copy-and-paste.

On the contrary than previous partitional clustering algorithms, Banerjee [15] used a MOGA to solve the fuzzy clustering problem with adaptative k and noisy data. This approach uses a quite complex representation scheme with each individual divided into two independent strings: one distinguish between clean and noisy data while the other one keeps the result of the partition.

Multiobjective spectral clustering is a recent topic with a scarce literature. One example is Wang [16], who proposed an evolutive multiobjetictive spectral algorithm clustering algorithm for datasets that contain different views of the same data, for instance, because data come from heterogeneus sources. As a consequence, the dataset is represented by means of several graphs. In this context the algorithm is able to automatically determine k by means of Pareto optimization.

To the author's knowledge, the only attempt to address spectral clustering with multiobjective computational intelligence used Harmony Search Algorithm (HSA), this is a search method inspired by musicians improvisation that has an increasing number of applications. Li et al. proposed the Spectral Clustering-based Adaptive Hybrid Multi-Objective Harmony Search Algorithm (SCAH-MOHSA) [17], which is a complex algorithm for community detection in graphs; it uses spectral clustering with a Multi-Objective HSA and local search.

## III. The Co-Evolutionary Multi-Objective Genetic Graph-based Clustering (CEMOG) Algorithm

This section describes the Co-Evolutionary Multi-Objective Genetic Graph-based Clustering (CEMOG) algorithm. CEMOG is a continuity-based clustering algorithm that was created using MOGGC [10] as a starting point. MOGGC was created to improve the robustness of the solutions reducing the dependency to the metric parameters and the search space. The main improvement of this new algorithm, compared with MOGGC, is that it is not necessary to give an initial number of clusters.

This approach combines MOGA with two objectives to guide the heuristic search using a co-evolutionary structure. CEMOG is applied in three steps:

1) **Similarity Graph generation:** A Similarity Function (usually based on a kernel function) is applied to the data instances, connecting all the points with each other. It generates the Similarity Graph.
2) **Genetic search:** CEMOG uses a MOGA to find a good graph partition. Giving an initial range of possible number of clusters $[k_{min} - k_{max}]$, the MOGA generates an initial population, with a sub-population per $k$ value, of possible solutions and evolves them using a fitness function to guide the algorithm to find the best solution. It stops when a good solution is found, or a maximum number of generations is reached.
3) **Clustering association:** The best solution of the Pareto Front is chosen as a solution of the algorithm and the data instances are assigned to the clusters according to the solution chosen and the sub-population who has generated this solution. The selection criterion is explain in Section III.F.

### A. Encoding

The encoding is a simple label-based representation [9] that follows the classical integer representation of GAs. Each individual is a $n$-dimensional vector (where $n$ is the number of data instances) which has integer values between 1 and the number of clusters of the sub-population it belongs. Each individual represents a cluster selection of the dataset.

### B. The k-adaptive approach

The design which helps to achieve the k-adaptive number of clusters goal is a co-evolutionary approach combined with a multi-objective algorithm. The co-evolution is focused on two points of view which are described below: macro-evolution (i.e., the evolution of the whole population) and micro-evolution (i.e., the evolution of each sub-population).

### C. Macro-evolution and the exchanger operator

We use an arbitrary graph topology for migrating individuals from sub-populations. The assumption is that all sub-populations have the same representation and same goals to solve (see Fig. 1).

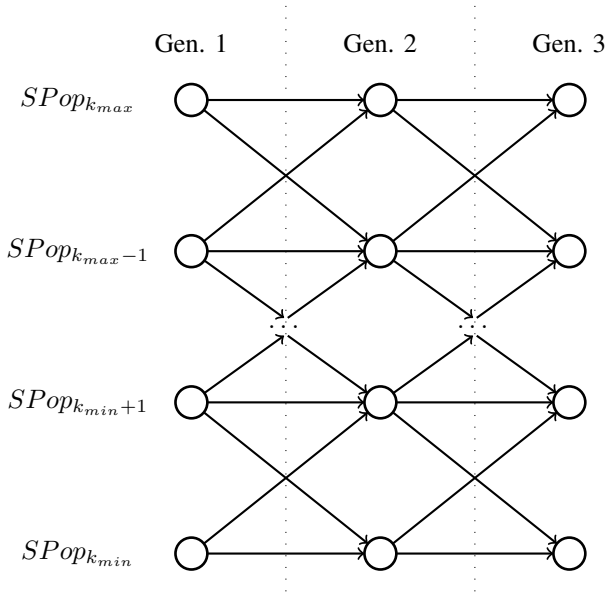


Fig. 1: Once the range of $k$ values is set between $k_{min}$ and $k_{max}$, the set of sub-populations is generated for each $k$ value. The different individuals are able to jump among sub-populations in each generation.

Each sub-population represents a possible k-value ranged from $k_{min}$ to $k_{max}$. These values generate a higher search space which complicates the genetic search; nevertheless, the new methodology helps to find a satisfactory solution in a range of possible number of clusters instead of a fixed one. The algorithm looks for the solutions in the different sub-populations. Moreover, it also uses a exchanger to send individuals from a population to another, modifying the environment of the different sets of chromosomes and encouraging genetic diversity. This exchanger improves the quality of the solutions and reduces the local solution convergence (e.g., local minimum), compared with the simple modification of the number of clusters in MOGGC which does not improve the convergence. The exchanger developed in CEMOG exchanges two random couples of each sub-population with its neighbour sub-populations, i.e., a couple of individuals of population $SPop_n$ is send to $SPop_{n+1}$ and other couple to $SPop_{n-1}$ (see Fig. 1). The genetic operators will modify the number of clusters of these solutions to adapt them to the new population.

### D. Micro-evolution and the MOGA operators

On the one hand, CEMOG, as a MOGA, uses the SPEA2 algorithm for the genetic evolution of the set of solutions whithin the sub-populations. SPEA2 starts with two populations $P_0$ and $\overline{P_0}$, the first is known as the internal population and the second is the external population which is initially empty (see line 1 of Algorithm 1). During each generation, the algorithm computes the fitness of both populations ($P_t$ and $\overline{P_t}$), and takes the non-dominant individuals to the external population of the next generation (see lines 3 and 4 of algorithm 1). Whether the external population is bigger than the initial size, it is reduced, and when the size is smaller, it is filled with dominated individuals of the original populations using a truncation method (see lines 5 to 9 of Algorithm 1). Next, it fills a mating pool with individuals of $\overline{P_{t+1}}$ selected by binary tournament and applies the genetic operations to generate the new population $P_{t+1}$ (see lines 13 and 14 of Algorithm 1). This algorithm keeps a copy of the best Pareto Front selection of each generation in the external population.

On the other hand, as a clustering algorithm, CEMOG begins with a $K_{size}$-Similarity Graph in the same way that the Spectral Clustering algorithm [7]. The $K_{size}$ value limits the memory used to a matrix $K_{size} \times N$ where $N$ is the number of data instances.

Finally, the MOGA operators used can be briefly summarized as follows:

- **Selection**: The selection process is a tournament selection with size $n$.
- **Crossover**: The crossover exchanges strings of numbers between the two chromosomes (both strings have the same length).
- **Mutation**: The mutation probability is adaptive, when an allele is selected for mutation, the operator changes its value with a random integer. It works as follows:
  1) For each chromosome, it randomly chooses if the mutation is applied. The mutation probability is fixed at the beginning.
  2) When a chromosome is chosen, it decides the alleles which are mutated. The decision considers the probability of the allele to belong to the cluster which has assigned. If the probability is high, the allele has a low mutation probability and vice versa. In this algorithm, this probability is calculated applying the metric defined in the fitness function to one allele.
  3) The alleles are mutated. The new value is a random number between 1 and the number of clusters.

### E. The fitness objectives

The fitness function is divided into two objectives: improve the data continuity degree and cluster separation.

*1) Data continuity degree:* This objective function is applied to each cluster. It calculates the total edges sum for each minimal spanning tree of each connected component of the $K_{size}$-Graph $G$ (see Algorithm 2). Starting in the first node (it supposes, without loss of generality, that the nodes are numerically ordered), the algorithm generates two lists: the first one initially contains all the nodes and the second one is empty (see line 1 of Algorithm 2). While any of the lists contain at least one element, the first list will give to the second all nodes connected within the neighbourhood of the current node and internally will count the minimal spanning tree edges (see lines 3 to 9 of Algorithm 2). Due to the graph is not full-connected, this process will follow with each connected component (see lines 10 to 17 of Algorithm 2). This metric measures the continuity of the data as a graph structure inside the clusters. The arithmetic average value of the metric is the result of this objective.

**Algorithm 1** Pseudo-code of the SPEA2 algorithm [18]

**Require:** $N$ (population set); $\overline{N}$ (archive size); $T$ (generations)
**Ensure:** $A$ (non-dominated set) .
1: $P_0 =$ random population; $\overline{P}_0 = \emptyset$;
2: **for** $t = 0 \rightarrow T$ **do**
3: Calculate Fitness of $P_t$ and $\overline{P_t}$.
4: Copy non-dominated individuals in $P_t$ and $\overline{P_t}$ to $\overline{P_{t+1}}$
5: **if** size($\overline{P_{t+1}}$) $> \overline{N}$ **then**
6: reduce $\overline{P_{t+1}}$
7: **else**
8: Fill $\overline{P_{t+1}}$ with dominated individuals in $P_t$ and $\overline{P_t}$
9: **if** $t == T$ or any stopping condition is satisfied **then**
10: Break the loop.
11: Fill the mating pool with individuals of $\overline{P_{t+1}}$ selected by binary tournament.
12: Apply the recombination and mutation to the mating pool and set $P_{t+1}$ to the resulting population.
13: **return** $A = \{$non-dominated individuals in $P_{t+1}\}$

**Algorithm 2** Data continuity degree algorithm

**Require:** $C$ cluster with an order relationship
**Ensure:** $\nu$ (connectivity factor) .
1: Let $L1 = C$ and $L2 = \emptyset$ and set $\nu = 1$;
2: Move the first element of $L1$ to $L2$;
3: **while** $L1 \neq \emptyset$ or $L2 \neq \emptyset$ **do**
4: Set $v_i =$ the first element of $L2$ (Extract it from the list);
5: **for** $v_j \in G$ **do**
6: **if** $vj \in L1$ and $v_j > v_i$ **then**
7: Move $v_j$ from $L1$ to $L2$;
8: $\nu + +$;
9: **if** $L2 = \emptyset$ **then**
10: **if** $L1 = \emptyset$ **then**
11: break;
12: Move the first element of $L1$ to $L2$;
13: **return** $\nu/|C|$;

*2) Clusters separation:* The second objective of the fitness function is the cluster separation. To ensure the cluster separation the following metric has been applied to each cluster:

$$\frac{\sum_{v_i \in C} \frac{\sum_{v_j \in G}\{w_{ij} |\ v_j \notin C\}}{|G|-|C|}}{|C|} \quad (1)$$

where C is a cluster, G is the $K_{size}$-Graph, $v_i$ is the vertex $i$, $w_{ij}$ is the edge weight value from node $i$ to node $j$. It calculates the arithmetic average value of the edge weights between the different clusters.
The MOGA implementation is necessary because both objectives are opposites: the first tries to improve the inter-clusters distance and the second the intra-cluster distance. In the first case, a single cluster would guarantee a maximum value while, in the second case, a cluster per instance would guarantee the maximum value.

### *F. Choosing the solution from the Pareto Front*

Due the necessity to choose one of the solutions from the Pareto Front, the experimental results (see Section IV) show that the solution with the highest value of the cluster separation metric in the Pareto Front always obtains better accuracy values compared with human-based classification. Therefore, this value has been chosen as the algorithm solution.

## IV. Experimental Results

This section shows the results of the experiments that we carried out to assess the hability of CEMOG to find good k-values and compare its performance with similar algorithms. The first part presents the datasets which have been used to test the algorithm. The second one describes the evaluation metrics and the experimental set-up. Finally, the last part shows the results on the synthetic and real-world datasets which have been taken from the literature.

### *A. Evaluation Datasets*

This section describes the different datasets which have been used for the algorithm testing phase. Synthetic and real world datasets have been used to check the algorithm accuracy. These datasets have been extracted from different works related to clustering problems.

*1) Synthetic datasets:* The datasets which have been chosen are:

- *Aggregation* (Ag) [19]: This dataset is composed by 7 clusters, some of them can be separated by parametric clustering.
- *Jain* (Jn) [20]: This dataset is composed by two surfaces with different density and a clear separation.
- *R15* [21]: This dataset is divided in 15 clusters which are clearly separated.
- *Spiral* (Sp) [22]: In this case, there are 3 spirals close to each other.

*2) Real-World datasets:* The datasets which have been chosen have been extracted from the UCI database [23]. They are the following ones:

- *Glass* (Gl): It contains 6 clusters with 9 attributes each and 214 instances. It also has been analysed in some clustering works as [24].
- *Libras Movement* (LM): It contains 15 clusters with 90 attributes each and 24 instances per class (total 360). It is identified for classification and clustering in the UCI database [23].
- *Ozone Level Detection* (OL): It contains 2 clusters with 73 attributes and 2536 instances. It has been chosen because of its simplicity according to the number of classes.
- *Wine Quality* (WQ) [25]: It contains 6 clusters with 11 attributes each and 4898 instances of white wine. it is also identified for classification and clustering in the UCI database [23].

| | SPop Size | $k_{min}$ | $k_{max}$ | Gen. | Cross. | Mut. | Eli. |
|---|---|---|---|---|---|---|---|
| Ag | 200 | 5 | 9 | 500 | 0.1 | 0.01-10$^-$4 | 10 |
| Jn | 50 | 2 | 6 | 200 | 0.2 | 0.2-10$^-$4 | 5 |
| R15 | 50 | 13 | 17 | 500 | 0.2 | 0.3-10$^-$4 | 5 |
| Sp | 200 | 2 | 6 | 200 | 0.1 | 0.01-10$^-$4 | 10 |

TABLE I: Best parameter selection (SubPopulation size, Generations, Crossover probability, Mutation probability and Elitism size) used in CEMOG algorithm for synthetic datasets and the best fitness value obtained. The tournament size is 2.

- *Page Block* (PB): It contains 5 clusters with 10 attributes each and 5473 instances. It has been chosen because of its complexity.

### B. Evaluation Techniques and Experimental Setup

The CEMOG algorithm has been compared against different clustering algorithms. These algorithms have been taken from the literature and from our previous work. The classical algorithms which have been chosen are K-means, Expectation Maximization and Spectral Clustering. In addition CEMOG has been compared against MOGGC, its previous implementation [10].

The similarity between the clusters has been calculated using the following similarity metric:

$$sim(C_i, C_j) = \frac{1}{2}\left(\frac{\sum_{q=1}^{n}\delta_{C_i}^q\delta_{C_j}^q}{|C_i|} + \frac{\sum_{q=1}^{n}\delta_{C_i}^q\delta_{C_j}^q}{|C_j|}\right) \quad (2)$$

where $n$ is the number of instances, $C_i$, $C_j$ the clusters which are compared, $|C_i|$ is the number of instances of cluster $C_i$ and $\delta_{C_i}^q$ is the Kronecker $\delta$ defined by:

$$\delta_{C_i}^q \equiv \delta_{C_i}(x_q) = \begin{cases} 0 & \text{if } x_q \notin C_i \\ 1 & \text{if } x_q \in C_i \end{cases} \quad (3)$$

where $x_q$ is an element. The evaluation process has calculated the maximum accuracy for all the algorithms. All of them have been executed 150 times per dataset. The metric which has been applied with K-means and EM is the Euclidean Metric defined by:

$$||x_i - x_j|| = \sqrt{\sum_{q=1}^{d}(x_i^q - x_j^q)^2} \quad (4)$$

Where $x_i = (x_i^1, \ldots, x_i^d)$ and $x_j = (x_j^1, \ldots, x_j^d)$.

And the metric for SC, MOGGC and CEMOG which has been used in the Similarity Graph Generation is the Radial Basis Function (RBF) defined by [26]:

$$s(x_i, x_j) = e^{-\sigma||x_i - x_j||^2} \quad (5)$$

The $\sigma$ value has been calculated using the approximation method elaborated by Andrew Ng in [27].

### C. Synthetic results for the CEMOG algorithm

Tables I and II show the parameters selection for MOGGC and CEMOG, respectively. This parameters have been chosen after a deep search of parameters in several ranges. In these cases, the $\sigma$ parameter to generate the similarity graphs of MOGGC and CEMOG is 100 (it has been approximated using the method described by Ng et al. [27]). The best accuracy results were selected for the algorithms. Table III shows the comparison of all the algorithms considered.

Fig. 2 depicts the Pareto Front of CEMOG applied to the synthetic datasets. Fig. 3 shows the accuracy values for different sub-populations of each dataset. The criterion used to choose these solutions prioritizes the cluster separation objective in the Pareto Front considering the solution of the data continuity objective.

CEMOG and MOGGC correctly cluster the Aggregation dataset. The Pareto Front defined by MOGGC (see Fig. 2) shows a dominant solution (mark with 'x' at the top-rigth corner). This winner solution belongs to the $k = 7$ sub-population which is the solution with the best accuracy (see Fig. 3). EM, K-means and SC have problems related to the data form. These problems might be a consequence of local minimum convergence in the search space.

The Spirals dataset is impossible to cluster using parametric algorithms and the Euclidean distance (that is, K-means and EM). This dataset is a perfect example for continuity-cluster separation algorithms such as SC, MOGGC or CEMOG. For that reason, all of them achieves the best accuracy values (see Table III). Furthermore, analysing the Pareto Front defined by CEMOG for Spirals, there is only one dominant solution (marked with 'x' at the to right corner), corresponding to $k = 3$ sub-population, which is the sub-population with the best accuracy (see Fig. 3).

The Jain dataset is also difficult for parametric clustering. It produces low accuracy values for EM and k-means compared with SC, MOGGC and CEMOG. This dataset is usually used to test continuity-clustering algorithms modifying the density of the clusters, in this case, the first cluster has clearly lower data density than the second cluster. According to the Pareto Front, CEMOG also has a clear dominant solution (marked with 'x' at the top-right corner) which is the $k$ value with best accuracy. MOGGC and SC have also good results.

In the case of the R15 dataset, the experiments show that EM obtains the best results for classical algorithms. SC obtains worse results than EM due to the noisy information and, therefore, the boundaries are not clearly defined. MOGGC obtains the maximum accuracy value. In the CEMOG case, the Pareto Front is defined by several solutions which cover different k values (see Fig. 2). In order to define the best solution, the cluster separation objective has been prioritized over the data continuity objective, as was mentioned before. Applying this criterion, the solution chosen is the top 'x' instead of the top right. The accuracy value of the solution is the maximum as Fig. 3 shows.

Finally, Fig. 3 shows an interesting remark. The k-values close to the best accuracy values have generally more accuracy

| Data | Pop. | Gen. | Cross. | Mut. | Eli. |
|---|---|---|---|---|---|
| Ag | 1000 | 200 | 0.1 | $0.01\text{-}10^{-4}$ | 50 |
| Jn | 100 | 200 | 0.4 | $0.03\text{-}10^{-4}$ | 10 |
| R15 | 100 | 2000 | 0.4 | $0.01\text{-}10^{-4}$ | 50 |
| Sp | 1000 | 200 | 0.4 | $0.03\text{-}10^{-4}$ | 50 |

TABLE II: Best parameter selection (Population, Generations, Crossover probability, Mutation probability and Elitism size) used in MOGGC algorithm for the different synthetic datasets and the best fitness value obtained. The tournament size is 7.

| Data | K-means | EM | SC | MOGGC | CEMOG |
|---|---|---|---|---|---|
| Ag | 86.29% | 78.68% | 88.66% | **100%** | **100%** |
| Jn | 78.28 % | 56.83% | **100%** | **100%** | **100%** |
| R15 | 80.50 % | 99.66% | 81.33% | **100%** | **100%** |
| Sp | 34.61 % | 34.93% | **100%** | **100%** | **100%** |

TABLE III: Best accuracy values obtained by each algorithm using the synthetic datasets.

than the rest, except for Jain, i.e., it seems that the distance of k to the optimal k is correlated with its accuracy. This observation suggests a way to automatically set the $k_{min}$ and $k_{max}$ values.

### *D. Real-world results for the CEMOG algorithm*

This section shows the experimental results of the CEMOG algorithm applied to real world datasets. We first describe the datasets preprocessing followed by the experimental results. As in the previous section, CEMOG is compared against the classical clustering algorithms (K-means, EM and SC) and MOGGC.

*1) Preprocessing:* The preprocessing process is divided in two steps: The first step studied the variables through histograms and correlation diagrams to reduce the dimensionality. The information provided by this phase showed the attributes which were useless because, for example, were constants or had a high correlation (more than 0.85 if we consider that the correlation values is in range $[0, 1]$) with other variables. This means that they may have variated the clustering results if they are not eliminated because of the redundant information. Those instances with missing values have also been deleted. Table IV shows results of the dimension reduction.

The second preprocessing step consisted on normalization. First, the attributes with outliers were recentralized, then the same range was applied for all. We combined Z-score to recentralize the distributions and MinMax to fix the range of all the values between 0 and 1.

| Data | I. Attributes | F. Attributes | I. Instances | F. Instances |
|---|---|---|---|---|
| LM | 90 | 18 | 360 | 360 |
| OL | 73 | 28 | 2536 | 1867 |

TABLE IV: Dimension reduction. This table shows the Initial Attributes and Instances with the Final Attributes and Instances after removing highly correlated variables and variables with missing values.

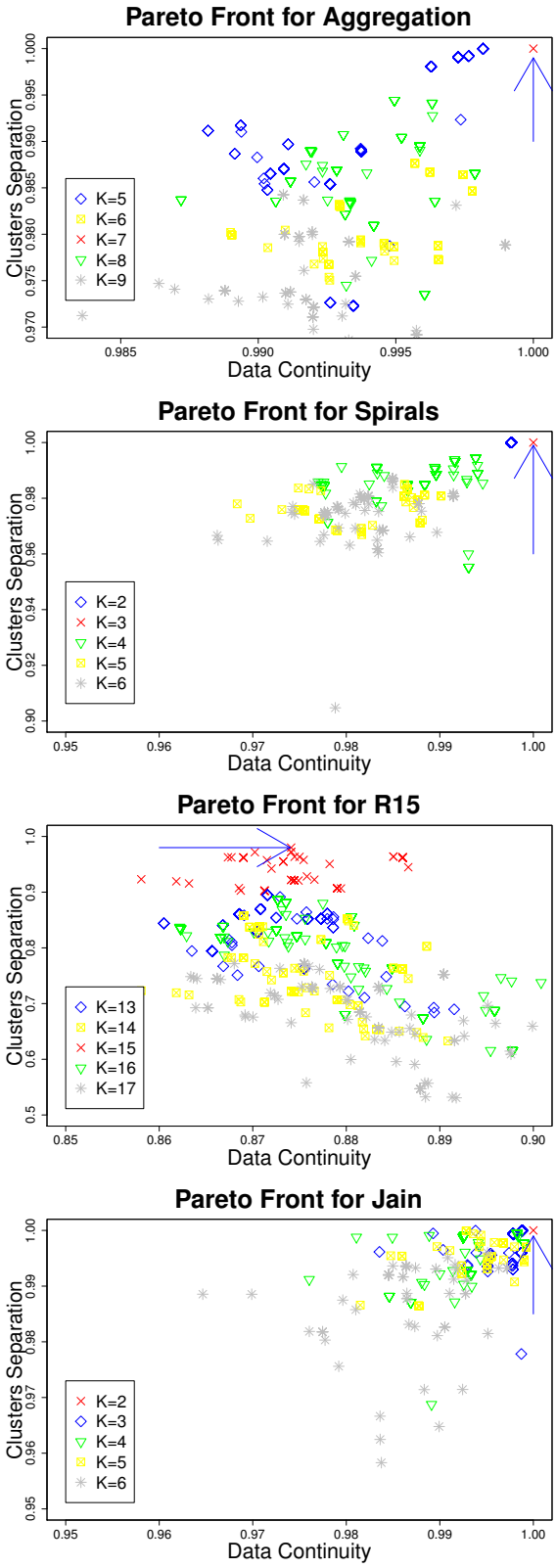


Fig. 2: Pareto Front generated by the CEMOG sub-populations chosen for the synthetic datasets. From top to bottom: "Aggregation", "Spiral","R15" and "Jain". The arrows mark the best solution.

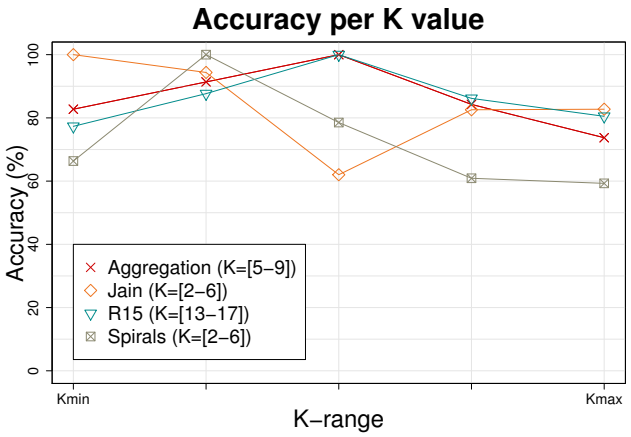


Fig. 3: Accuracy results for CEMOG algorithm ranged from $k_{min}$ to $k_{max}$. The legend shows the values for the range per synthetic dataset.

| Data | SPop. | $k_{min}$ | $k_{max}$ | Gen. | Cross. | Mut. | Eli. |
|---|---|---|---|---|---|---|---|
| Gl | 100 | 5 | 9 | 100 | 0.5 | 0.01-$10^{-4}$ | 10 |
| LM | 100 | 13 | 17 | 200 | 0.5 | 0.01-$10^{-4}$ | 10 |
| OL | 100 | 2 | 6 | 200 | 0.5 | 0.01-$10^{-4}$ | 10 |
| WQ | 100 | 5 | 9 | 2000 | 0.4 | 0.01-$10^{-4}$ | 50 |
| PB | 1000 | 3 | 7 | 1000 | 0.4 | 0.2-$10^{-4}$ | 50 |

TABLE V: Best parameter selection (Sub-Population, Generations, Crossover probability, Mutation probability and Elitism size) used in CEMOG algorithm for the different real datasets and the best fitness values obtained. The tournament size is 7.

| Data | K-means | EM | SC | MOGGC | CEMOG |
|---|---|---|---|---|---|
| Gl | 45.79% | *47.20%* | *47.20%* | **47.66%** | **52.34%** (k=5) |
| LM | *46.94%* | 43.61% | 46.11% | **50.00%** | **48.05%** (k=14) |
| OL | 76.06% | 60.15% | *94.38%* | **96.46%** | **96.46%** (k=2) |
| WQ | 23.64% | *28.50%* | **40.08%** | **40.08%** | **40.08%** (k=7) |
| PB | 45.30% | 56.97 % | **75.15%** | **75.15%** | **75.15%** (k=5) |

TABLE VI: Best accuracy values obtained by each algorithm during the experimental results applied to the UCI datasets.

*Glass* (Gl), *Wine Quality* (WQ) and *Page Block* (PB) datasets contain a few number of attributes. After the analysis of the variables, the correlation showed that the dimensionality reduction was not necessary. However, in the case of *Libras Movement* (LM) and *Ozone Level Detection* (OL) datasets, there were a lot of attributes that did not contribute to the analysis due to the high correlation between them (see Table IV). These attributes were eliminated leaving 18 of 90 attributes for Libras Movement and 28 of 73 for Ozone Level Detection. In the Ozone Level Detection dataset, there were several instances with missing values, all these instances were omitted for the analysis (see Table IV).

*2) Algorithm execution:* The experiments followed the same procedure than the previous synthetic datasets experiments. The value of $\sigma$ was approximated to 100 for *Glass*, 2 for *Libras Movement* and *Ozone Level Detected* and 0.1 for *Wine Quality* and *Page Block*. The results and best k value for CEMOG are shown in Table VI.

In the following, we describe the results for each dataset.

*Glass* dataset is difficult for clustering; the results show that both, the classical and the new algorithms have problems to blindly separate the classes. In this case, SC obtains the best classical algorithms results. MOGGC obtains good results - better than the classical algorithms indeed; however, CEMOG obtains the best results. A remarkable fact is that the k selection process has chosen k $= 5$ while the original number of classes is 7. This might explain why the other clustering algorithms have worse results than CEMOG.

*Libras Movements* dataset is also a difficult classification case, again the classical and the new algorithms have problems to blindly separate the classes. In this case, k-means obtains the best results from the classical algorithms. MOGGC obtains the best general results and CEMOG obtains similar results than MOGGC because the k selection has chosen a different k value of the number of classes.

*Ozone Level Detected* is easier for the continuity-clustering algorithms. In this case, SC, MOGCC and CEMOG obtain the best classification results, all with the same accuracy value.

*Wine Quality* is a difficult problem for clustering techniques. The worst results are achieved by the parametric algorithms (the accuracy is lower than the 30%). The results of the non-parametric techniques are the same.

*Page Block* is also a difficult problem for the parametric approximation. These algorithms have achieved low accuracy while SC and MOGGC and CEMOG have achieved the same solutions, with better accuracy. Analysing the parametric clustering results, they show that the data instances within the clusters should be separated between them instead of having a clear union between them.

Due to SC, MOGGC and CEMOG are focused on continuity-based clustering, it might be the reason because these algorithms achieve similar results. The main advantage of CEMOG compared against the others is that it does not need a fixed number of clusters and also find more solutions which can be promising.

## V. Conclusions and Future Work

This work proposes CEMOG, a new k-adaptive spectral clustering algorithm, inspired by MOGGC, that combines Co-Evolution with Multi-Objective Genetic Algorithms. CEMOG uses a simple integer encoding in a GA combined with the SPEA2 algorithm. In comparison to MOGGC, the new algorithm is a k-adaptive approach,which obtains good results in a bigger search space. The results show that the new algorithm obtains excellent results that are better than the classical algorithms, and has a similar (or better) clustering results than previous obtained using MOGGC.

The future work will be focused on several improvements that could be made to CEMOG. On the one hand, the effects of noisy information should be deeply analysed, whereas, on the other hand, the $k_{min}$, $k_{max}$ selection might be also an

adaptive process. Finally, other fitness functions which could improve the CEMOG algorithm convergence, and the clusters quality, could be studied.

## REFERENCES


[1] I. Yoo and X. Hu, "A comprehensive comparison study of document clustering for a biomedical digital library medline," in *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, ser. JCDL '06. New York, NY, USA: ACM, 2006, pp. 220–229.

[2] P. Haider, L. Chiarandini, and U. Brefeld, "Discriminative clustering for market segmentation," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 417–425.

[3] A. Pascual, M. Barcéna, J. Merelo, and J.-M. Carazo, "Application of the fuzzy kohonen clustering network to biological macromolecules images classification," in *Engineering Applications of Bio-Inspired Artificial Neural Networks*, ser. Lecture Notes in Computer Science, J. Mira and J. Sánchez-Andrés, Eds. Springer Berlin Heidelberg, 1999, vol. 1607, pp. 331–340.

[4] G. Bello-Orgaz, M. D. R-Moreno, D. Camacho, and D. F. Barrero, "Clustering avatars behaviours from virtual worlds interactions," in *Proceedings of the 4th International Workshop on Web Intelligence; Communities*. New York, NY, USA: ACM, 2012, pp. 4:1–4:7.

[5] D. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[7] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[8] H. Menéndez and D. Camacho, "A genetic graph-based clustering algorithm," in *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, ser. Lecture Notes in Computer Science, H. Yin, J. Costa, and G. Barreto, Eds. Springer Berlin / Heidelberg, 2012, vol. 7435, pp. 216–225.

[9] E. Hruschka, R. Campello, A. Freitas, and A. de Carvalho, "A survey of evolutionary algorithms for clustering," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 2, pp. 133 –155, march 2009.

[10] H. Menéndez, D. F. Barrero, and D. Camacho, "A multi-objective genetic graph-based clustering algorithm with memory optimization," in *2013 IEEE Conference on Evolutionary Computation*, vol. 1, June 20-23 2013, pp. 3174–3181.

[11] C. Coello, G. Lamont, and D. Van Veldhuisen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, ser. Genetic and evolutionary computation series. Springer Science+Business Media, LLC, 2007.

[12] K. Ripon and S. Kwong, "Multi-Objective Data Clustering using Variable-Length Real Jumping Genes Genetic Algorithm and Local Search Method," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 3609–3616.

[13] J. Handl and J. Knowles, "Exploiting the Trade-Off - The Benefits of Multiple Objectives in Data Clustering," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds., vol. 3410. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 547–560.

[14] N. Matake, T. Hiroyasu, M. Miki, and T. Senda, "Multiobjective clustering with automatic k-determination for large-scale data," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07*. New York, New York, USA: ACM Press, Jul. 2007, p. 861.

[15] A. Banerjee, "An improved genetic algorithm for robust fuzzy clustering with unknown number of clusters," in *2010 Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, Jul. 2010, pp. 1–6.

[16] X. Wang, B. Qian, J. Ye, and I. Davidson, "Multi-objective multi-view spectral clustering via pareto optimization." SDM, 2013, pp. 234–242.

[17] Y. Li, J. Chen, R. Liu, and J. Wu, "A spectral clustering-based adaptive hybrid multi-objective harmony search algorithm for community detection," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[18] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Gloriastrasse 35, CH-8092 Zurich, Switzerland, Tech. Rep. 103, 2001.

[19] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, Mar. 2007.

[20] A. Jain and M. Law, "Data clustering: A user's dilemma," in *Pattern Recognition and Machine Intelligence*, ser. Lecture Notes in Computer Science, S. Pal, S. Bandyopadhyay, and S. Biswas, Eds. Springer Berlin / Heidelberg, 2005, vol. 3776, pp. 1–10.

[21] C. Veenman, M. Reinders, and E. Backer, "A maximum variance cluster algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 9, pp. 1273 – 1280, sep 2002.

[22] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recogn.*, vol. 41, no. 1, pp. 191–203, Jan. 2008.

[23] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[24] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 36–.

[25] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decis. Support Syst.*, vol. 47, no. 4, pp. 547–553, Nov. 2009.

[26] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr, "Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection," *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 2, pp. 512 –518, feb. 2008.

[27] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2001, pp. 849–856.