UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

MASTER IN COMPUTER SCIENCE AND
TELECOMMUNICATION ENGINEERING

MASTER THESIS

Adaptive K-means algorithm for overlapping graph clustering

Author: Gema Bello Orgaz
Advisor: D. David Camacho Fernández

May 4, 2012

Index

Abstract

The graph clustering problem has become highly relevant due to the growing interest of several research communities in social networks and their possible applications. Overlapping graph clustering algorithms try to find subsets of nodes that can belong, at the same time, to different clusters. Several techniques and methods, like fuzzy or genetic algorithms amongst others, have been applied to deal with this problem. In social-based applications it is quite usual for a node of the network to belong to different groups, or communities, in the graph. This can be represented as a set of overlapping sub-graphs in the network, containing a subset of common nodes. Therefore, algorithms that try to discover, or analyse, the behaviour of these networks need to handle this feature, detecting and identifying the overlapping nodes. This work shows a soft clustering approach based on a genetic algorithm where a new encoding is designed to allow two main goals. First, the automatic adaptation of the number of communities that can be detected (K). Second, the definition of several fitness functions that guide the searching process using some measures extracted from graph theory. Finally, this new approach has been experimentally tested using the Eurovision contest dataset, a well-known social-based data network, to show how overlapped communities can be found using this method.

1. Introduction

1.1. Clustering Techniques

The clustering problem can be described as a blind search on a collection of unlabelled data, where elements with similar features are grouped together in sets. There are three main techniques to deal with the clustering problem [1]: overlapping [2] (or non-exclusive), partitional [3] and hierarchical [4]. Partitional clustering consists in a disjoint division of the data where each element belongs only to a single cluster, overlapping clustering allows each element belongs to multiple clusters, and hierarchical clustering nests the clusters formed through a partitional clustering method creating bigger partitions, grouping the clusters by hierarchical levels. In this work, the approach are focused in the overlapping clustering techniques trying to "relax" a well-known classical partitional technique named K-means using a genetic algorithm approach. K-means is a clustering algorithm that uses a fixed number (K) of clusters and looks for the best division of the dataset (through a predefined metric or distance) in this number of groups.

In the process of community finding problems, K-means cannot be directly applied because it does not allow overlapping. In contrast, it is common for communities to share members. An alternative solution could be fuzzy k-means [5] which allows every one element to belong to several clusters giving a probability of membership, therefore same kind of overlapping for an element can be considered. Others community finding algorithms are CPM (Clique percolation method) and Edge Betweenness. CPM (Clique percolation method) [6] finds communities using k-cliques (where k is fixed at the beginning and the network is represented as a graph). It defines a community as the highest union of k-cliques. CPM has two variants: directed graphs and weighted graphs [7]. Edge Betweenness [8] is based on finding the edges of the network which connect communities and removing them to determine a good definition of these communities.

Several clustering algorithms, such as K-means, have been improved using genetic algorithms [1]. A genetic algorithm is inspired by biological evolution [9]: the possible problem solutions are represented as individuals belonging to a population. The individuals are encoded using a set of chromosomes (called the genotype of the genome). Later these individuals are evolved, during a number of generations, following a survival/selection model where a fitness function is used to select the best individuals from each generation. Once the fittest individuals have been selected, the algorithm reproduces,

crosses and mutates them trying to obtain new individuals (chromosomes) with better features than their parents. The new offspring and, depending on the algorithm definition, their parents, will pass to the following generation. This kind of algorithms have been usually employed in optimization problems [10], where the fitness function tries to find the best solution among a population of possible solutions which are evolving. In other approaches, such as clustering, the encoding and optimization algorithm are used to look for the best set of groups that optimizes a particular feature of the data. In this new approach each chromosome is used to define a set of K clusters which represents a solution to the clustering problem.

Clustering techniques can also be applied to different kinds of representations of the data collection like strings, numbers, records, text, images and semantic or categorical data [11, 12]. In this work, we apply a clustering technique to data that can be represented as a graph, trying to find groups whose nodes share similar graph-based features.

The proposed algorithm in this work is based on genetic algorithm methods for graph-based clustering techniques that are described in the next section. We are trying to combine these approximations to improve the results of graph clustering through classical optimization methods. The main contribution of this work can be summarized as follows: our approach tunes up the centroid positions and the number of clusters (K), maximizing the distance between them, and minimizing the distance between the elements found in each cluster.

We also based this new algorithm on network analysis [13]. The main measures used to analyse networks are the average distance between nodes, and the clustering coefficient (CC). The CC can be seen as the number of triangles formed by the edges of the network over the total possible number of triangles. Both these measures are usually employed to define the nature of the network [13].

Distance between nodes and clustering coefficient measures can be used to guide a genetic clustering algorithm with the goal of finding groups in a graph which minimize or maximize these measures. Although each of the measures can be used separately, the new genetic algorithm approach combines them using a hybrid function which gives different weights to each measure. This combination generates some problems specially when it is necessary to decide which measure is more relevant than the others. That is the reason why some experimental tests have been carried out to obtain the final weight for each

measure that will be used in the hybrid fitness function.

Once a particular encoding and several fitness functions were designed, the new algorithm is applied to the Eurovision Song Contest dataset. This well-known contest provides interesting data which has been deeply studied and analysed from different perspectives (social, political, economical and historical, among others) over the last decades [14, 15]. This data has been preprocessed and represented as a social network.

Finally, the main contribution of this Master Thesis can be briefly summarized as follows:

1. A new genetic-based community finding algorithm has been designed an implemented using:
   - Clustering Techniques
   - Genetic Algorithms
   - Graph Metrics
2. Several experiments have been carried out to analyse the behaviour of the new implemented approach.

1.2. Historical Background of Eurovision Song Contest

The Eurovision Song contest can be understood as a complex system [16], where interactions between countries are heavily influenced by factors like geography, shared history, culture and migration patterns. Voting patterns for each country seem to be dictated by a latent affinity between countries, and not by the artistic value of the song. It provides an active forum, where countries are free to give opinions about the rest of the participants without fear of economic or political backlash [17][18].

This song contest is an annual competition among members of the *European Broadcasting Union* [19], running continuously ever since it's inauguration in 1956. The contest is executed in the following fashion: each country submits a song and performer with which to compete. All songs are then performed live, in a transmission available to all participating countries. Once all songs have been performed, votes are casted (previously by a *jury*, currently through *televotes* and a *jury*), and a winner is selected.

The contest has undergone a series of changes throughout the years, in an effort to keep it fresh and maximize viewer attention. From 1956 to 1996, votes where casted by a jury of representatives sent from each of participating countries. Jurors casted all of ten individual point-votes ranging from 1-8, 10 and 12 points, with no repetitions. Points are given in decreasing order: the participant with the better song receives 12 points, the next receives 10, and so on. In 1997 *televoting* was introduced in five countries (Austria, Germany, Sweden, Switzerland and United Kingdom), to gradually displace the jury-based system until 2004 when *televoting* was made mandatory for all participants. *Televote* technology allows viewers to cast their votes via phone, sms or the internet for a set window of time–normally within the live broadcast.

In 2004 a *semi-finals* round was introduced to offset the increasing number of participant in the contest. In order to participate in the Eurovision contest, participants must pass this preliminary round, thereby limiting the number of participants to a manageable size. That last winner and the so-called *Big Four* are exempt from this filter (they are the four highest contest contributors: France, Germany, Spain and the United Kingdom). However, all countries, finalists and not, are allowed to vote in the final round, which inflates the number of countries that vote and overall score of the winners each year. Critics contested that because of migration patterns, *televoting* had a tendency to favor certain countries, and in 2009 started the implementation of the current voting system. A hybrid system of *televoting* and a jury was implemented, whereby each part contributes half of the total vote tally for each country.

2. Related Work

This section starts with a general introduction to clustering techniques. After this brief introduction, how genetic algorithms have been applied to clustering techniques is described. Later, an overview of graph clustering methods and some current applications to social networks that uses the clustering coefficient are presented. Following, some community finding algorithm methods are showed, paying close attention to social network analysis. Finally, past studies on the Eurovision contest using data mining techniques are presented.

2.1. Clustering

Clustering techniques are frequently used in data mining and machine learning methods. A popular clustering technique is K-means. Given a fixed number of clusters, K-means tries to find a division of the dataset [3] based on a set of common features given by distances or metrics that are used to determine how the cluster should be defined. Other approximation, such as Expectation-Maximization (EM) [20], uses a variable number of clusters. EM is an iterative optimization method that estimates some unknown parameters computing probabilities of cluster membership based on one or more probability distributions; its goal is to maximize the overall probability or likelihood of the data being in the final clusters [21].

Other research lines are trying to improve these algorithms. For example, some *online* methods have been developed to avoid the K-means convergence problem to local solutions which depend on the initial values [22]. Some other improvements of K-means algorithm are related to deal the different kind of data representation, for example, mixed numerical data [23] and categorical data [24]. There are also some studies comparing methods with different datasets, for example, Wang et al. [25] compare self-organizing maps, hierarchical clustering and competitive learning where establishing molecular data models of large size sets. Other approaches related to genetic algorithms, and directly related to this work, will be described in the following subsection.

Machine learning techniques have also been improved through the k-means algorithm, for example, reinforcement learning algorithms[26, 27] or using topological features of the data set [28, 27] which can also be helpful for data visualization.

As we mentioned before, in this new approach we are working with over-

lapping clustering instead of partitional clustering (which is the case of the original K-means). In overlapping clustering there are two main approaches[1]: soft (each object fully belongs to zero or more clusters) and fuzzy (each object belongs to zero or more clusters with a membership probability). Fuzzy instances are important when there is not a complete deterministic separation in the data set, a good example is human activity recognition [29]. One of the first approximations was fuzzy K-means [5], which can also benefit from combining with a genetic approach [30, 31]. In this work (overlapped clustering in social data) soft computing allows each node in the graph to belong to one or more subgraphs, and no membership probability is considered.

2.2. Genetic Algorithms for Clustering

Genetic algorithms have been traditionally used in optimization problems. The complexity of the algorithm depends on the codification and the operations that are used to reproduce, cross, mutate and select the different individuals (chromosomes) of the population [32, 33].

These algorithms have also been used for general data and information extraction [10]. The operators of the genetic algorithms can also be modified. Some examples of these modifications can be found in (Poli and Langdon, 2006)[34] where the algorithm is improved through backward-chaining, creating and evaluating individuals recursively reducing the computation time. Other applications of genetic clustering algorithms can be found in swarm systems, [9] software systems [35], file clustering [36] and task optimization [37], amongst others.

The genetic clustering approximation tries to improve the results of the clustering algorithm using different fitness functions to tune up the cluster sets selection. In (Cole, 1998)[38], different approaches of the genetic clustering problem, especially focused in codification and clustering operations, can be found. There is also a deep revision in (Hruschka et al., 2009)[1] which provides a complete up to date state of the art in evolutionary algorithms for clustering.

There are several methods using evolutionary approaches from different perspectives, for example: (Aguilar, 2007)[39] modifies the fitness considering cluster asymmetry, coverage and specific information of the studied case; (Tseng and Yang, 2001)[40] uses a compact spherical cluster structure and a heuristic strategy to find the optimal number of clusters; (Maulik and

Bandyopadhyay, 2000)[41] use the clustering algorithm for metric optimization trying to improve the cluster centre positions; (Shi et al., 2011)[42] based the search of the genetic clustering algorithm in their Extend Classifier Systems which is a kind of Learning Classifier System, in which a fitness of the classifier is determined by the measure of its prediction's accuracy; (Das and Abraham, 2008)[43] use Differential Evolution, a method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

Some of those previous methods are based on K-means, for example: (Krishna and Murty, 1999)[44] replace the crossover of the algorithm using K-means as a search operator, and (Wojciech and Kwedlo, 2011)[45] also use differential evolution combined with K-means, where it is used to tune up the individuals obtained from mutation and crossover operators. Finally, other general results of genetic algorithm approaches to clustering can be found in (Adamska, 2005)[46]. There are also other complete studies for multi-objective clustering in (Handl et al., 2004)[47] and for Nearest Neighbour Networks in (Huttenhower et al., 2007)[48].

2.3. Graph Clustering

Graph theory has also proved to be an area of important contribution for research in data analysis, especially in the last years with its application to manifold reconstruction [49] using data distance and graph representation to create a structure which can be considered as an Euclidean space (which is the manifold).

Graph models are useful for diverse types of data representation. They have become especially popular over the last years, being widely applied in the social networks area. Graph models can be naturally used in these domains, where each node or vertex can be used to represent an agent, and each edge is used to represent their interactions. Later, algorithms, methods and graph theory have been used to analyse different aspects of the network, such as: structure, behaviour, stability or even community evolution inside the graph [13, 50, 51, 52].

A complete roadmap to graph clustering can be found in (Schaeffer, 2007)[12] where different clustering methods are described and compared using different kinds of graphs: weighted, directed, undirected. These methods are: cutting, spectral analysis and degree connectivity (an exhaus-

tive analysis of connectivity methods can be found in (Hartuv and Shamir, 2000)[53]), amongst others. This roadmap also provides an overview of computational complexity from a theoretical and experimental point of view of the studied methods.

In network analysis, is common to use a graph representation, especially for the social network approach where users are connected by affinities or behaviours. This approximation has been studied in some of the small world networks based on two main variables: the average distance between elements and the clustering coefficient of the graph [13, 51, 52].

The present work is closer to the network approach and has been developed over different kinds of graphs (undirected and directed graphs). The clustering coefficient of these kind of graphs are used to find clusters in the network [51].

2.4. Community Finding Approach

The main application of the communities approach are social networks. The clustering problem is more complex when applied it to find communities in networks (subgraph identifications). A community can be considered as a subset of individuals with relatively strong, direct, and intensive connections [50] between them. Some algorithms such as Edge Betweenness [8] or Clique Percolation Method (CPM) [6] have been designed to solve this problem following a deterministic process. CPM [6] finds communities using k-cliques (where k is a fixed value of connections in a graph) which are defined as complete (fully connected) subgraphs of k vertices. It defines a community as the highest union of k-cliques. CPM has two variants: directed graphs and weighted graphs [7]. The Edge Betweenness algorithm [8] is based on finding the edges of the network which connect communities and removing them to determine a good definition of these communities.

Other approximations related to the finding-community problem can be found in (Reichardt and Bornholdt, 2006)[54] where different statistical mechanics for community detection are used. (Pons and Latapy, 2005)[55] use random walks to compute the communities. However, in this work genetic algorithms are used because we are interested in optimization methods for tuning up the definition of the clusters, allowing to adapt the size and membership of these clusters using metrics and features selected from graph characteristics.

Finally, another work based on metrics used to measure the quality of the communities can be found in (Newman and Girvan, 2004)[56], and metrics that can be used to find the structure of a community in very large networks in (Clauset et al., 2004)[57]. Genetic algorithms have also been applied to find communities or clusters through agglomerative genetic algorithms [58] and multi-objective evolutionary algorithms [59] amongst others.

2.5. Studies on the Eurovision Contest using data mining techniques

Past studies on the Eurovision Contest have centered around social and historical facts, coupled with data clustering methods [60, 61], regression analysis [62], dynamical networks [17], or analytical identification of statistically significant tends [14], all of which were able to group the participating countries into blocs of like behavior.

In [60] and [61], one of the earliest analyses, the Eurovision community was split into three blocs: The *Mediterranean Bloc*, the *North Bloc* and the *West Bloc*. In this model, the west bloc consistently amassed the highest number of votes, and was the largest of the three. In [14], two large blocs are identified, *The Viking Empire* (Scandinavian and Baltic countries) and *The Warsaw Pact* (Russia, Romania and the old republic of Yugoslavia), and a number of other smaller blocs. The work of [17] uses dynamic network analysis to study voting partnerships, observing that these may not be static, but are instead susceptible to change over time.

All these studies show that stable communities throughout time can be identified using data mining techniques in this web dataset. Therefore, this well-known social-based dataset is used to show how overlapping communities can be found using the new genetic-based community finding algorithm proposed in this work.

3. Genetic-based Community Finding Algorithm Description

The Genetic-based Community Finding (GCF) Algorithm developed uses a genetic algorithm to find the best K communities in a dataset that could be represented as a graph, and where any particular neighbour could belong to different clusters. In an initial designing phase a simple version of the algorithm, with a binary encoding using a fixed value for K, has been developed. This first algorithm version is called *K-fixed GCF*, or simply K-fixed algorithm. The experiments carried out show that some important improvements could be made to obtain better solutions for the communities detected, and to increase the performance of the clustering process. To achieve these goals, a more complex encoding has been designed to include the value of K in the evolutionary process. This new version is called *K-adaptive GCF* algorithm.

This section describes both algorithms including the encoding, the genetic algorithm (crossover and mutation operators) and the fitness functions designed for each one.

3.1. K-fixed GCF Algorithm

The initial version of the algorithm was based on a standard genetic algorithm with a binary codification to represent a community. The number of possible K communities was fixed to a predefined value. The goal of this algorithm was to find overlapping communities in a dataset represented as an undirected graph.

3.1.1. Encoding

In this version of the algorithm the genotypes are represented as a set of binary values. Each allele represents the membership of a node of the graph and each chromosome is used to represent a community. The chromosome length will be equal to the graph size.

This encoding defines a direct relationship between each node in the graph and the allele of the chromosome. In this binary representation the value "1"means that the node belongs to a community and the value "0"the opposite (see Figure 1).

Figure 1: A Chromosome representing a community. Each allele represents a node of the graph and its belonging, or not, to the current community. In this example, a community built by three nodes of the graph is shown.

3.1.2. The algorithm

The simply GCF Algorithm with a fixed K value works as follows:

1. A random population of communities is generated.
2. The population evolves using a standard GA. Therefore, the following steps are repeated until a fixed number of iterations, or a convergence value, are reached:
   a) Evaluate the fitness function of each chromosome in the population.
   b) Copy the n-best chromosomes to the new population (Elitism Selection). It prevents losing the n-best found solutions.
   c) Generate the rest of the new population by repeating the following steps:
      1) **Selection**: select two parent chromosomes from the population.
      2) **Crossover**: crossover the parents to form a new offspring.
      3) **Mutation**: using a given mutation probability, the value for each bit in the allele is changed.
   d) Replace the old population with the new population.
3. The chromosomes which are the K-best solution of the algorithms are selected. Our selection process **subsumes** the communities which have better fitness and belong to a bigger community. An individual subsumes another when the subgpraph that represents its community, contains at least all the nodes and connections of the other one. This **subsumption** process has the following steps:
   a) An empty list of K elements is created.
   b) The chromosomes are sorted by their fitness value.

c) While the list is not full, a new chromosome is selected. If the new individual represents a new community, it is included in the list. However, if this individual represents a community that currently is contained by some other individual in the list (the nodes encoding this chromosome are a subset of a currently stored chromosome), the more general chromosome is selected.

d) The process stops when the K best individuals are found, or when there are no more individuals to select.

Finally, the rest of the main characteristics of the GA algorithm; selection, crossover and mutation operators, are briefly described:

- **Selection**. The parent selection can be done in different ways, but the main idea is to select the better parents to produce better offspring in each generation. When creating the new population by crossover and mutation, there is a big probability of loosing the best community (chromosome). So we have used the elitism selection method which first copies the n-best communities to the new population. The rest of the population is generated in a classical way, as we have described in the previous steps of the algorithm.

- **Crossover**. To do the crossover, the algorithm chooses two crossover points at random. Then everything between these two points is copied from the first parent to the second and vice versa, see Figure 2.

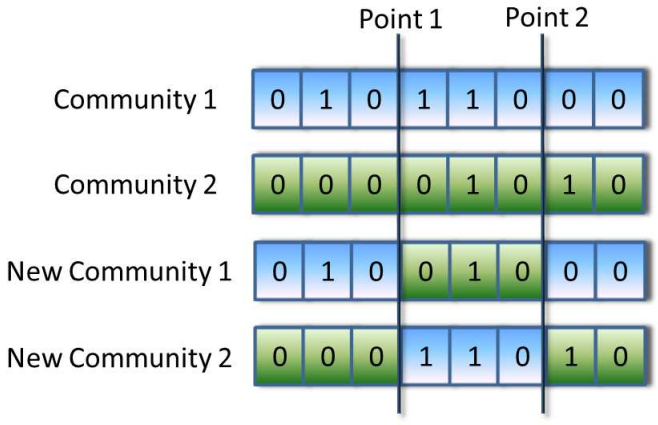

Figure 2: Crossover of two communities.

- **Mutation**. Once the crossover process has finished the mutation is executed. This operator is applied to prevent the falling of all solutions into a local optimum of the problem. In our approach, for a binary encoding, we have chosen a few alleles (nodes of a community) at random and changed their values from 1 to 0 or viceversa using a mutation probability. The mutation operator will work as we can see in Figure 3.

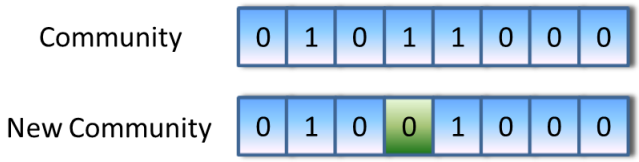

Figure 3: Mutation of a community. The fourth allele has been selected, and the bit has been changed using the mutation probability, so this node is now excluded from the community.

3.1.3. Fitness Functions

In this initial approach three kind of fitness functions were implemented, each of them with a different goal. The first one tries to find nodes with a similar rating behaviour (minimal distance fitness), the second one tries to find clusters using the clustering coefficient (maximum clustering coefficient fitness) and, finally, the last fitness function (hybrid fitness) combines both strategies to find communities with a similar rating behaviour and whose members are connected between them. These fitness functions can be described as follows:

- **Minimal Distance Fitness Function(MDF)**. The goal of this fitness function is to find similar node communities. The evaluation of this fitness function is done using the following criteria:

  1. Each node belonging to a community is represented as a vector of attributes. The definition of these attributes depends on the problem being solved.
  2. The average euclidean distance between vectors of attributes within a community is calculated. The fitness calculates distances to be taken into account from peer to peer, between all vectors.

3. The fitness value for the community is the average distance of the values calculated in the previous step (we are trying to minimize the fitness). It is a measure of similarity for those rows, hence it checks if they follow the same similarity pattern. We call this average distance d_{in} (see Figure 4).
4. This fitness penalizes those cases where the community has a single node, giving it a value of zero.

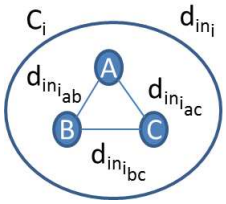

Figure 4: Sample sub-graph illustrating a community and the distances that are calculated in the MDF fitness function. The distance d_{in} represents the average distance calculated between the nodes which belong to a community.

- **Maximum Clustering Coefficient Fitness Function (**MC^{2}F**)**. The goal of this fitness is to discover communities whose members are connected between them. It is measured through the clustering coefficient, defined as follows:

  Definition 1 *Let $G = (V, E)$ be a graph where E is the set of edges and V the set of vertices. Let $v_i \in V$ be a vertex and $e_{ij} \in E$ an edge from v_i to v_j. Let Σ_{v_i} be the neighbourhood of the vertex v_i defined as $\Sigma_{v_i} = \{v_j \mid e_{ij}, e_{ji} \in E\}$. If k is considered as the number of neighbours of a vertex, we can define the clustering coefficient of a vertex as follows:*

$$C_i = \frac{|\{e_{jk}\}|}{k(k-1)}$$

  Where $|\{e_{jk}\}|$ satisfies that $v_j, v_k \in \Sigma_{v_i}$.

  Definition 2 *The clustering coefficient of a graph is defined as:*

$$C = \frac{1}{|V|} \sum_{i=0}^{|V|} C_i$$

Where $|V|$ is the number of vertices.

The fitness function takes the sub-graph defined by the community and calculates its clustering coefficient. It returns the inverse value, because the genetic algorithm tries to minimize the fitness function.

- **Hybrid Fitness Function (HF)**. This last fitness function combines Clustering Coefficient and Distance fitness strategies: it tries to find a set of communities satisfying both conditions previously defined. With this method we try to find strong and similar communities (members which are highly connected between them and have similar behaviour). The function defined is a simple weighted function: suppose that $F(x, y)$ is the fitness function, CC the clustering coefficient and d_{in} the distance between nodes, the value of HF fitness is:

$$F_i(CC, d_{in}) = w_1 * \frac{CC_i}{Max(\{CC_i\}_{i=1}^{K})}$$

$$+w_2 * \frac{d_{in_i}}{Max(\{d_{in_i}\}_{i=1}^{K})}$$

Where w_i are the weights given to each fitness: $w_i \in (0, 1)$. These values were experimentally obtained and setted to $w_1 = 0.1$ and $w_2 = 0.9$.

3.2. K-adaptive GCF Algorithm

In the previous algorithm, one of the possible improvements that can be performed is that the parameter K (the number of communities found) could change its value through the execution of the clustering process. To achieve this, the encoding and the fitness function have been modified to obtain a new algorithm version.

3.2.1. Encoding

In this new approach, the possible solutions can contain groups of communities, and not just an individual community. For this reason, the genotypes (chromosomes) are represented as a set of vectors of binary values. Each allele represents a community that is composed by a set of binary values, one for each node in the graph. This binary vectors are similar to the chromosomes of the previous encoding, the value 1 meaning that the node belongs to the

community and value 0 the opposite. The number of binary vectors (communities) that the chromosome (group of communities) has, corresponds to the value of K in the solution, see Figure 5.

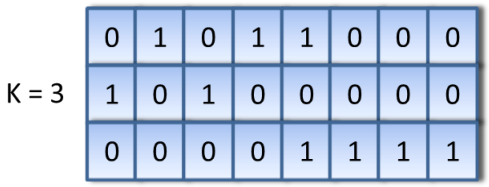

Figure 5: A Chromosome representing a group of communities of the graph. Each allele is a individual community where its binary vector represents the nodes of the graph and their belonging or not to the current community. In this example the solution contains 3 vectors representing three different communities, hence the K is equal to 3.

3.2.2. The K-Adaptive algorithm

The GCF Algorithm with adaptive K value works as follows:

1. A random population of community groups is generated.
2. The population evolves using a standard GA. The steps of the process are the same as was previously described in the the previous section for the K-fixed algorithm.
3. The chromosome that has the best fitness function value is selected as final solution.

Although the genetic algorithm has not been changed, the new codification has modified how the genetic operators (crossover and mutation) are applied. The new operators work as follows:

- **Crossover**. To apply the crossover operator, the algorithm chooses a random crossover point. Then every community preceding this point is copied from both parents to create a first new child, and every community succeeding this point is copied to create a second new child, as Figure 6 shows. The crossover point selects complete chromosomes, so whole communities are interchanged.

- **Mutation**. Once the crossover operator has finished, the mutation is executed. The algorithm chooses some values of the vectors that represent the communities at random, and change their values (with a predefined probability) from 1 to 0 or viceversa, see Figure 7.

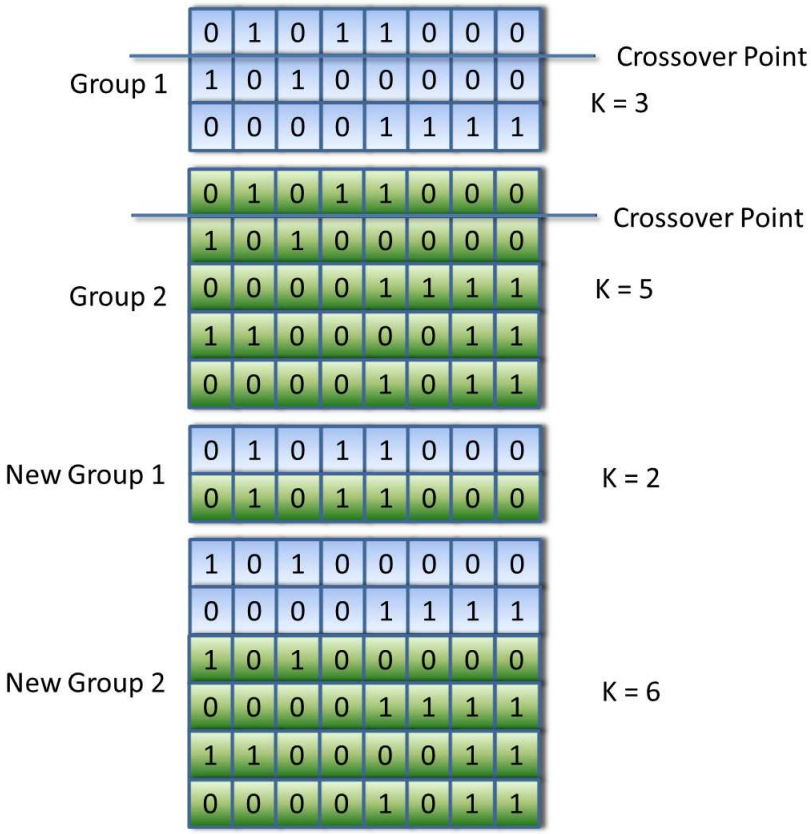

Figure 6: Crossover of two groups of communities whit different K. The new generated offspring maintains the maximum length of a community, it allows to generate new groups with a variable number (K) of communities.

3.2.3. The Centroid Fitness Function (CF)

The initial algorithm encoding (K-fixed) only allows to use metrics related to measures of a member belonging to their own community. Therefore, metrics such as the clustering coefficient, or the minimal distance between

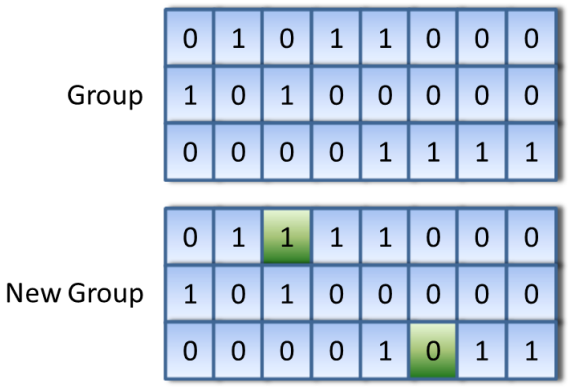

Figure 7: Mutation of a group of communities with K equal to 3. In this example, two nodes from two different communities have been modified.

nodes were used. However, the new encoding makes possible to include measures between groups of different communities.

We have designed a new fitness function, called Centroid Fitness (CF), that calculates the distance between the community centres belonging to a particular chromosome. This new measure is called d_{out} and it has been represented in Figure 8. In this case, large distances between centres could be desirable because it represents a bigger gap between classes or communities.

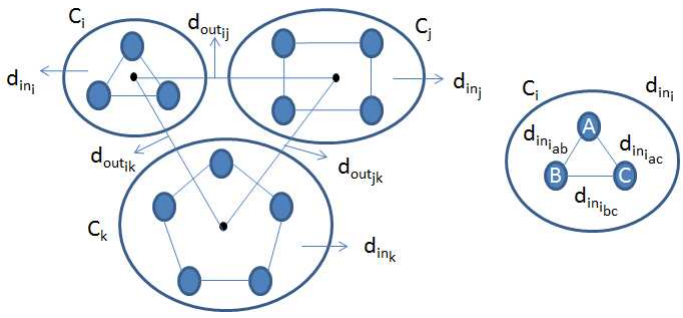

Figure 8: Sample network illustrating three communities and all of the distances that are calculated. The distance d_{in} represents the average distance calculated between the nodes which belong to a community, and the distance d_{out} represents the distance between community centres.

As a result of this new measure, that can be calculated for each indivi-

dual, a new fitness function which combines the Clustering Coefficient, the distance between nodes (d_{in}) and finally the distance between centres (d_{out}) can be designed. The idea of this new fitness is to find a set of communities that could satisfy all of the previously defined conditions. This new method tries to find groups of communities where each community is strong and similar, but also whose communities are the most different as possible between themselves.

The function defined is a simple weighted function: let $F(x, y)$ be the fitness function, CC the clustering coefficient, d_{in} the distance between nodes, and d_{out} the distance between centres, the value of the new fitness is calculated as follows:

$$\begin{aligned} F_i(CC, d_{in}, d_{out}) = w_1 * \frac{CC_i}{Max(\{CC_i\}_{i=1}^{K})} \\ + w_2 * \frac{d_{in_i}}{Max(\{d_{in_i}\}_{i=1}^{K})} \\ + w_3 * \frac{d_{out_i}}{Max(\{d_{out_i}\}_{i=1}^{K})} \end{aligned}$$

Where w_i are the weights given to each fitness: $w_i \in (0, 1)$. These values were set experimentally to $w_1 = 0.05$, $w_2 = 0.05$ and $w_3 = 0.9$.

4. Dataset Description

The Eurovision Song Contest has been studied using different clustering methods since the nineties [14, 15]. The main interest was to study and analyse alliances between countries, which had been reflected in form of communities or country clusters found. For this reason we have selected the dataset of this contest to carry out the experimental phase of our algorithm. The data used in this work has been extracted from Eurovision's official website [63].

4.1. The Dataset representation: The Eurovision voting system

Since 1975, the scoring system in the Eurovision Contest consists of the following rules. Each country distributes among other participants the following set of points: 1, 2, 3, 4, 5, 6, 7, 8, 10, 12. Countries give the highest score to the best song and the lowest to the less popular or less preferred. Once all the votes are added up, the final ranking is obtained. The country with the highest score wins the contest.

This data can be easily represented using a graph for each year of the contest. In this graph, the nodes will be countries and the points emitted can be used to weight the edges. The graph could be *directed* (the edges represent votes), or *undirected* (the edges only connect countries which have exchanged points in any direction). If we consider the latter, it is similar to setting edge weights uniformly to 1. According to this, the dataset of the votes emitted in a particular year could be represented as a graph, as is showed in Figure 9.

4.2. Study and Comparison of the Eurovision network in a random context

The first approximation that shows patterns can be obtained using a simple comparison between the Eurovision graph and a randomly generated graph with the same rules applied in the contest. Each participant country assigns ten set of points randomly among the remaining participant countries (generating an edge for every point cast). We call this graph representation *Random network*.

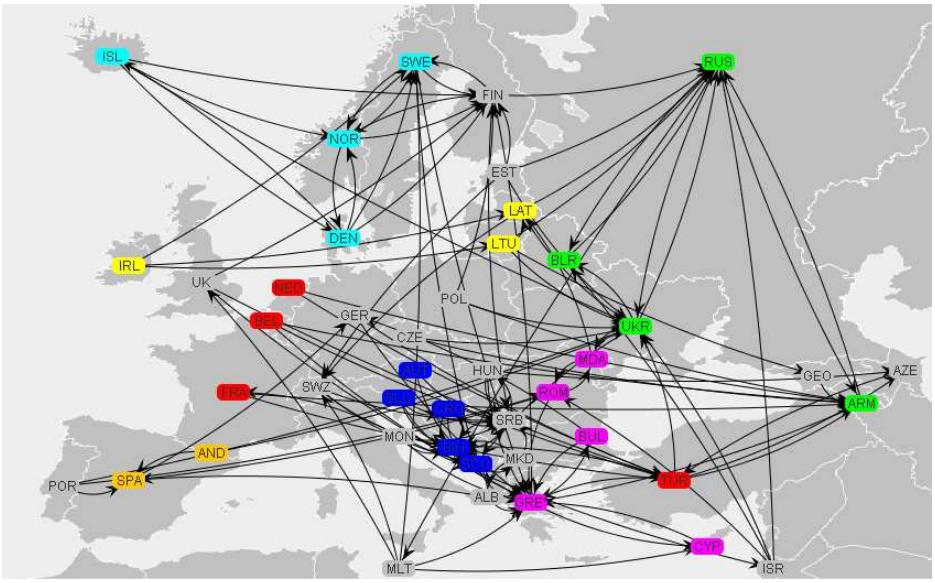

Figure 9: Eurovision graph example illustrating the votes emitted between the countries in 2009.

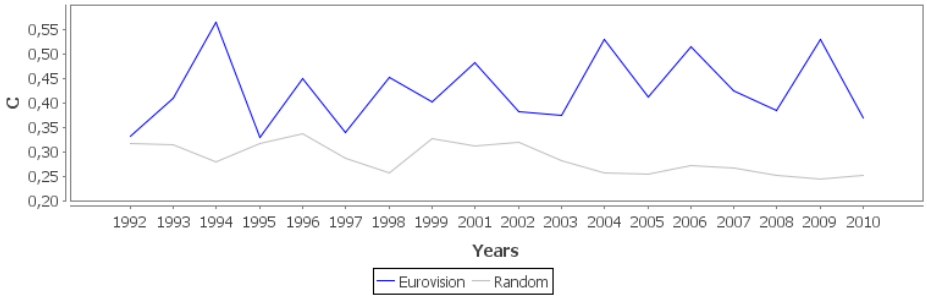

Figure 10: Clustering Coefficient comparison between the Eurovision network and a random graphs.

The *random network* model assumes that a given country does not favours or penalize other countries and all songs have equal musical quality. Therefore a country X will give points randomly to another ten countries. If, for example, N countries are considered then the probability that a country X votes to a country Y is given by P = 10/(N-1). Usually, in social networks, two vertices with corresponding edges to a third vertex have a higher probability of being connected to each other. Hence, our hypothesis is that may be possible to observe the same effect in the Eurovision network. Therefore, to

study this effect it is reasonable to analyse the clustering coefficient defined in the previous subsection where Clustering Coefficient Fitness Function was described.

Figure 10 shows the clustering coefficients calculated for years ranging from 1992 to 2010. When the two different graphs are compared, Eurovision and Random network, a greater CC in the Eurovision graphs can be shown. It means that the distribution of edges in the graph is not random, or otherwise, there is an "intention of vote" between countries. Therefore, we could conclude that communities, or alliances between countries, could exist.

5. Experimental Results

5.1. Preliminary analysis of fitness functions

In the previous section a data analysis using the clustering coefficient was performed (Figure 10). This analysis confirms the existence of clusters or communities in the Eurovision graph representation. The 2009 year dataset shows the greatest difference in the clustering coefficient, meaning this year contains a large set of different communities. Hence, this year has been selected to perform an initial study for all of previous fitness functions designed.

This preliminary study has been divided in two parts, one for each version of the algorithm. To compare these two algorithm versions, the following measures (which have been previously defined in the fitness functions description) are considered:

- d_{in}: It provides information about the node similarity within clusters.
- CC: It provides information about the inner connections of the clusters or the k-cliques.
- d_{out}: It provides information about the distances between centroids.

Table 1 shows the experimental set up. In this table, we can see the parameters of the K-fixed and K-adaptive versions of the algorithm that have been experimentally obtained. $\mu+\lambda$ is the selection criteria used in both genetic algorithms, where λ is the number of offspring (population size), and μ is the number of the best parents that survive from the current generation to the next.

5.1.1. Fitness function analysis for K-fixed algorithm

Firstly, using *K-fixed* algorithm, the previous described measures (d_{in}, CC and d_{out}) have been calculated to compare the results obtained by each fitness function. The obtained values of these measures for each fitness functions with the 2009 dataset, are shown in Table 2.

K is a parameter of the genetic algorithm that sets the number of communities. Table 2 presents the communities obtained using K equal to 6. This value was experimentally obtained simulating different executions of the algorithm for values of K ranging from 2 to 10. Once a complete study over the

Algorithm	K-fixed	K-Adaptive
Mutation probability	0.2	0.03
Generations	2500	500
Population size	3000	1000
Selection criteria $(\mu + \lambda)$	$3000 + 300$	$1000 + 100$
K value	6	-

Table 1: Genetic Parameters of GCF Algorithm

Fitness	**Communities**	d_{in}	**CC**
MDF	**Lithuania Latvia**	**10,91**	**0**
MDF	**Sweden Denmark**	**11,04**	**0**
MDF	**Sweden Hungary**	**11,31**	**0**
MDF	**Cyprus Moldova**	**11,40**	**0**
MDF	**Israel Netherlands**	**11,66**	**0**
MDF	**Albania Germany**	**11,83**	**0**
MC^2F	**Sweden Bosnia-Herzegovina Moldova Russia Finland Ukraine Iceland Turkey Germany**	**20,57**	**1**
MC^2F	**France Sweden Moldova Russia Finland Iceland Germany Azerbaijan UnitedKingdom**	**21,20**	**1**
MC^2F	**France Sweden Moldova Finland Romania Iceland Germany Azerbaijan UnitedKingdom**	**21,78**	**1**
MC^2F	**France Estonia Sweden Finland Iceland Germany UnitedKingdom**	**20,93**	**1**
MC^2F	**Sweden Moldova Russia Finland Ukraine Iceland**	**20,55**	**1**
MC^2F	**Estonia Sweden Bosnia-Herzegovina Finland Azerbaijan Iceland Turkey Germany**	**21,89**	**1**
HF	**Estonia Sweden Finland Iceland**	**18,03**	**1.0**
HF	**Sweden Moldova Russia Finland Ukraine Iceland**	**19,52**	**1.0**
HF	**Norway Sweden Denmark Iceland**	**18,77**	**0.92**
HF	**Moldova Russia Ukraine Poland**	**16,40**	**0.75**
HF	**Armenia Russia Lithuania Ukraine**	**16,56**	**0.75**
HF	**France Germany United-Kingdom**	**19,93**	**1.0**

Table 2: Communities found using K = 6. The distances between centres (d_{out}) obtained by fitness are: (a) MDF = 14.65, (b) MC^2F = 5.40 and (c) HF = 11.26.

available data was made, the optimal number of communities, with minimal overlapping, was found with K equal to 6. An analysis of the results obtained attending to each fitness is the following:

- **Minimal Distance Fitness Function(MDF)**. The first fitness function takes the minimum values of d_{in} distance. But it can be noticed that the number of members contained in these communities is dramatically small, as can be seen in Table 2. The d_{in} distance values obtained are lower, meaning that the communities found have similar features, but all of these groups only have two nodes.
- **Clustering Coefficient Fitness Function (**$\mathrm{MC^2F}$**)**. The resulting communities are shown in Table 2 identified by the fitness and we can see that many of them present high overlapping among members. The distance between centres (d_{out}), it has decreased dramatically from *14.65* (obtained by the previous fitness function) to *5.40*. Therefore, the communities found are very similar to each other, and present a higher overlapping. Considering the d_{in} distance, it is increased. The goal of finding larger groups has been achieved, but now these groups present too much overlapping to be considered as stable communities. So the final goal of the algorithm has not been really achieved.
- **Hybrid Fitness Function (HF)**. Finally, in this last fitness function, the combining of the two previous functions enables to discover groups of nodes which have similar features and whose members are connected between them. In Table 2 we can see that the distance between centres, d_{out}, has been greatly improved. Now this value is closer to the value obtained by the first fitness function (*11.26*). The d_{in} distance, and the clustering coefficient take intermediate values. And in addition, the given communities found have an appropriate size and a reduced overlapping.

Finally, in terms of distance measures, the results have been greatly improved using the hybrid fitness (HF). The distance between centres (d_{out}) increases dramatically from the $\mathrm{MC^2F}$ function to the hybrid one. Therefore, the communities found are far from each other and they can be better differentiated. The distance d_{in} obtains lower values, meaning that the found communities have more similar members. And finally, the clustering coefficient takes similar, and very high values, in all the cases. Based on these experimental results we can conclude that globally the hybrid approach performs better.

5.1.2. Comparison of fitness functions for K-fixed and K-adaptive algorithms

The next experiments were executed using the *K-adaptive* algorithm. Figure 11 shows the experimental results, comparing both versions of the algorithm. Fitness functions labelled with an asterisk represent the results for K-adaptive version of the algorithm (the Centroid Fitness (CF) function only could be calculated for K-adaptive algorithm because only the new encoding designed makes possible to include measures between groups of different communities).

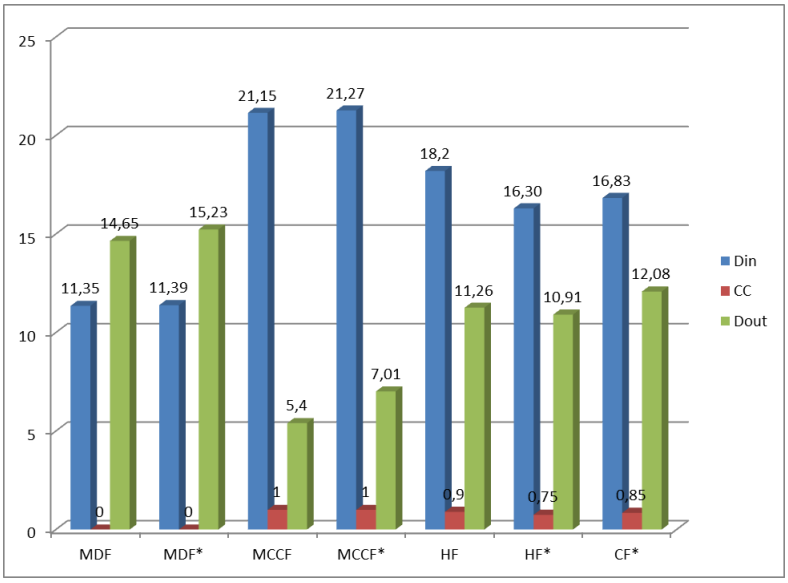

Figure 11: Values of the clustering coefficient and the distances d_{in} and d_{out} obtained using the designed fitness functions with both versions of the algorithm. The fitness functions labelled with an asterisk show the values for the K-adaptive algorithm.

As we can seen in the previous figure, the first two fitness functions, (MDF and MDF^*), take the minimum (d_{in}) distance and the maximal d_{out} distance, but the value of the CC is 0 in both cases. It means that the members of the communities are not connected between them.

In the next two functions (MC^2F, MC^2F^*) the opposite situation is encountered. The maximum possible value of CC is reached, but the distance measures get dramatically worse.

Both approaches have been combined in new hybrid fitness functions (HF, HF^*) that try to find new communities with better values for all the considered measures. Figure 11 shows the distance between centres (d_{out}) and the distance between nodes (d_{in}), take values lying between the first and second functions. Finally, the clustering coefficients (0,9 and 0,75 respectively) are closer to the values obtained by the second fitness functions, that obtain the maximum possible value (1).

The last fitness considered, the Centroid Fitness function (CF^*), obtains similar results for CC and d_{in} values and improves the d_{out} distance. This expected result came from the own definition of this function, that uses the distance between centroids to determine how to build the community.

Finally, all the experimental results from these fitnesses are compared for both versions of the algorithm. It can be noticed that the K-adaptive algorithm obtains similar or better results than the K-fixed algorithm in all the cases. Therefore, the CF function has been selected to experimentally test our community finding approach against other community finding algorithms.

5.2. Experimental Evaluation of GCF algorithms

5.2.1. Comparison between algorithms

In this section, we will compare the different results that we have obtained using CPM and EBC algorithms against the results of the new algorithms designed. The periods which we have been considered as most representative were:

- 1992-1996: Jury-based voting system was used exclusively.
- 2004-2008: Televoting was used exclusively, as well as having a semi-finals round.

As we can see in these results, the d_{in} measure is minimized by both genetic algorithms (K-fixed and K-adaptive), however the first version of the algorithm (K-fixed) obtains better results, see Figures 12 and 13. The new approach (both GFC algorithms) obtains similar results, and a big gap is presented between these genetic algorithms and EBC or CPM. It means that the community members found with GCF algorithms have more similar features than with EBC or CPM.

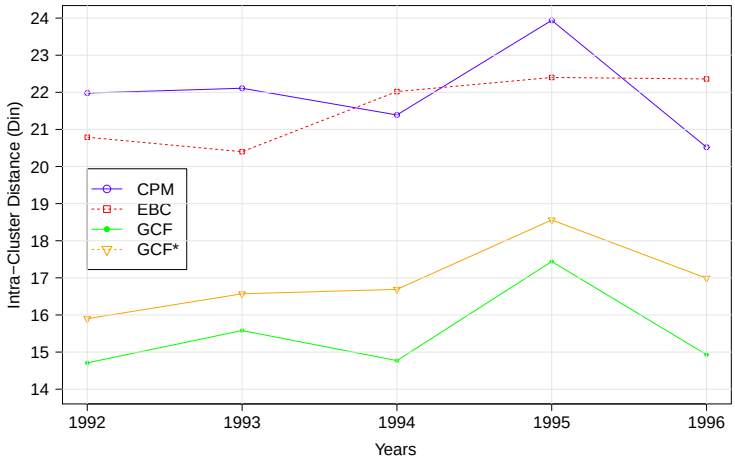

Figure 12: d_{in} comparison of the Eurovision Contest Song from 1992 to 1996 using CPM, EBC and the new GCF algorithms.

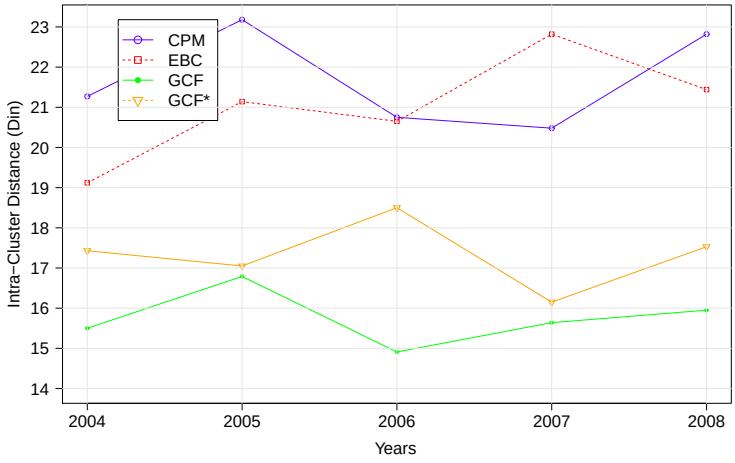

Figure 13: d_{in} comparison of the Eurovision Contest Song from 2004 to 2008 using CPM, EBC and the new GCF approaches.

The Figures 14 and 15 show the CC measure results, and as we can observe, its value is maximized by both genetic algorithms. In this case the new genetic algorithm approximation using an adaptive K value obtains the best results, followed by the first version of the algorithm with fixed K. The EBC

and CPM algorithms obtain the worst CC results, meaning there are fewer connections between nodes within communities.

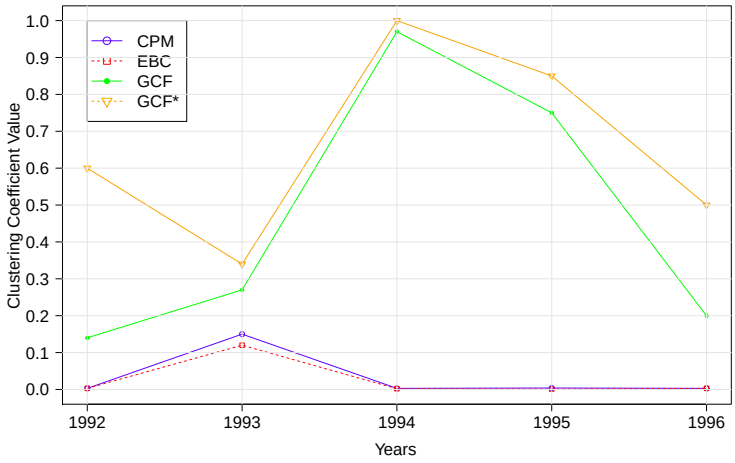

Figure 14: *CC* comparison of the Eurovision Contest Song from 1992 to 1996 using CPM, EBC and the new GCF algorithms.

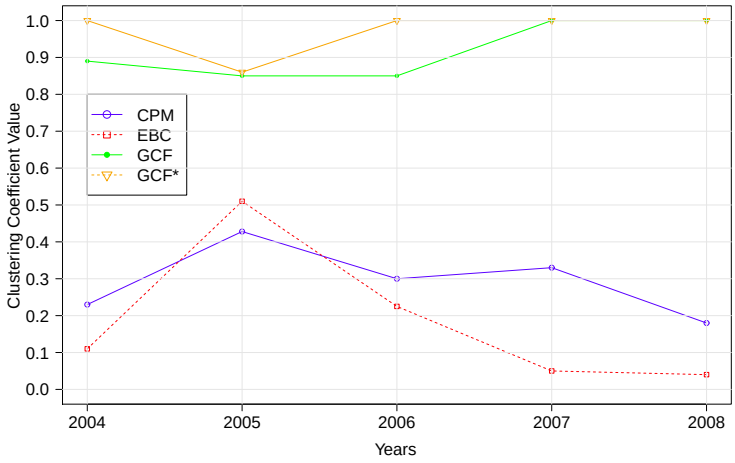

Figure 15: *CC* comparison of the Eurovision Contest Song from 2004 to 2008 using CPM, EBC and the new GCF algorithms.

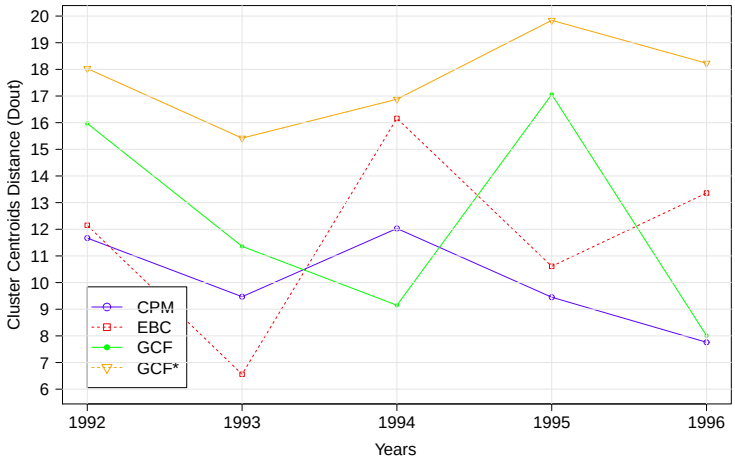

Figure 16: d_{out} comparison of the Eurovision Contest Song from 1992 to 1996 using CPM, EBC and the new GCF algorithms.

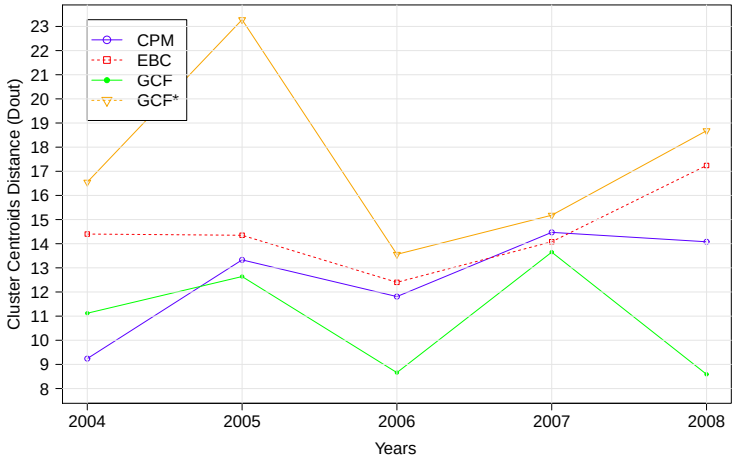

Figure 17: d_{out} comparison of the Eurovision Contest Song from 2004 to 2008 using CPM, EBC and the new GCF approaches.

Regarding the d_{out} measure, see Figures 16 and 17, where we can observe that it is maximized by both genetic algorithms. As in the previous case, the new genetic algorithm approximation obtains the best results. It was one of the original goals of the algorithm modification. The difference

observed in cluster centroid distance between the results obtained by the first genetic approach and those thrown by EBC and CPM algorithms is not far too noticeable. Nonetheless, the adaptive GCF version always improves that value.

5.2.2. Community Interpretation

In this subsection we compare the results of the communities founded by the new K-adaptive algorithm, giving them a human interpretation. The 2006 year result contains the greatest number of finding communities (K equals to 5). Other years have obtained values from 2 to 4. This year contains the largest set of different communities, therefore it has been selected to perform this community interpretation.

The next three figures plot the communities found in a geographical context, where a high correlation between neighbouring countries and their membership to like communities can already be appreciated. A example of the neighbour effect is the subset conformed by Norway, Sweden and Finland, that we can see in all the maps these three countries belong to the same community.

The CPM community results (see Figure 18) show that there are big communities with great overlapping, where overlapped countries are in bold. In this map, several country sub-groups that are neighbours or have similar cultural roots can be appreciated:

- *Baltic States*: Lithuania, Latvia and Estonia.
- *Nordic Countries*: Norway, Sweden, Finland, and Denmark.
- *Balkan Countries*: Macedonia, Albania, Serbia, Bosnia-Herzegovina, Croatia and Slovenia.
- *Old Soviet Union*: Russia, Belarus, Ukraine and Armenia.

After analyzing all these communities, it is clear that exists partnerships among neighboring countries and their historical and cultural roots.

The EBC results are less fit in the geographical context (see Figure 19), but also displayed neighbour sub-groups as follows:

- *Nordic Countries*: Norway, Sweden and Denmark.

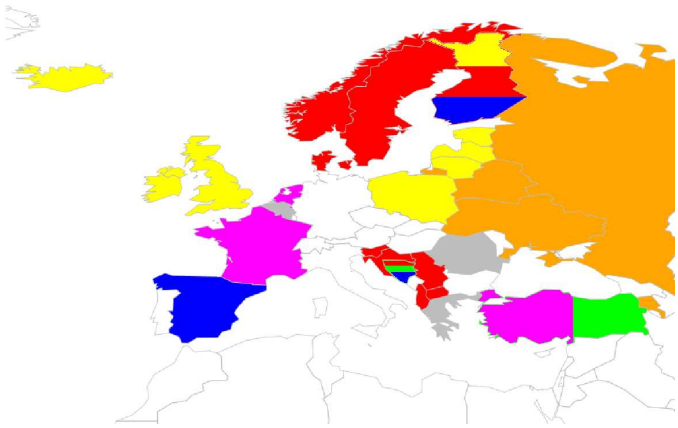

Figure 18: CPM Cluster Results of 2006. The communities are: [Spain, **Bosnia and Herzegovina** and **Finland**], [France, Netherlands and **Turkey**], [Iceland, Ireland, United Kingdom, Poland, Lithuania, Latvia, Estonia and **Finland**], [Norway, Sweden, **Finland**, Macedonia, Albania, Serbia, **Bosnia and Herzegovina**, Croatia and Slovenia and Denmark], [Belgium, Romania and Greece], [**Turkey**, **Bosnia and Herzegovina**] and [Russia, Belarus, Ukraine, Armenia]

- *Balkan Countries*: Slovenia, Croatia, Bosnia-Herzegovina, Serbia, Macedonia and Romania.
- *North-West Countries*: pain, France, Belgium, Germany and Netherlands.

EBC algorithm does not allow overlapping, but it also generates big communities whose d_{in} and CC measures take the worst values.

On the other hand, the resulting communities obtained through the adaptive algorithm are smaller, see Figure 20. This is expected if we consider that our algorithm tries to find communities whose members are highly connected between them, and also have similar characteristics. We can see that the new algorithm find communities where also the neighbourhood effect occurs:

- *Baltic States*: Lithuania and Latvia.
- *Nordic Countries*: Norway, Sweden and Finland.
- *Balkan Countries(1)*: Macedonia and Bosnia-Herzegovina.
- *Balkan Countries(2)*: Greece and Romania.

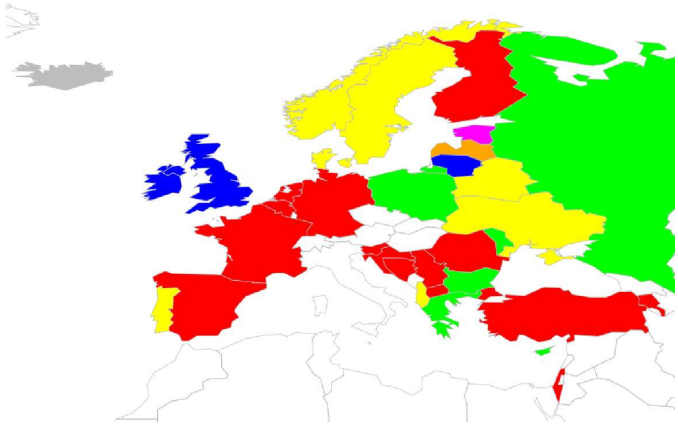

Figure 19: EBC Cluster Results of 2006. The communities are: [Ireland, United Kingdom, Lithuania], [Estonia], [Sweden, Norway, Portugal, Ukraine, Belarus, Denmark, Albania], [Spain, France, Belgium, Germany, Netherlands, Slovenia, Croatia, Bosnia and Herzegovina, Serbia, Macedonia, Romania, Turkey], [Iceland] and [Poland, Russia, Greece, Moldavia, Bulgaria] and [Latvia]

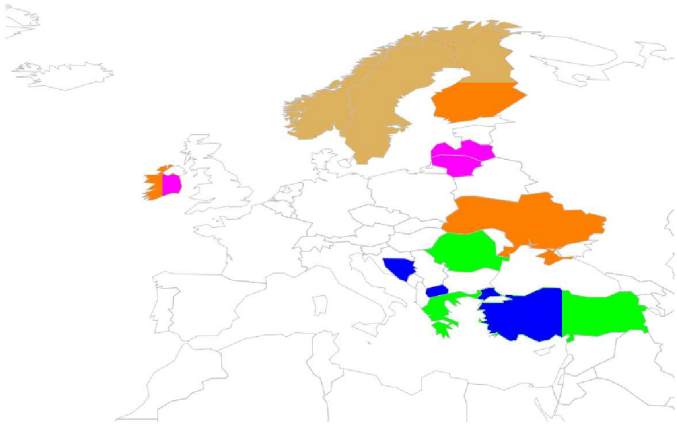

Figure 20: New GCF Cluster Results of 2006. The communities are: [**Ireland**, **Finland**, Ukraine], [**Turkey**, Macedonia, Bosnia and Herzegovina], [Lithuania, Latvia and **Ireland**], [Sweden, Norway and **Finland**], [Greece, **Turkey** and Romania]

Once the members of the communities are analysed an important issue appears immediately: most of the communities, or a subset of them, are contained in the communities found in the CPM algorithm. This is the case of

the community formed by Norway, Sweden and Finland or the formed by Ireland, Lithuania and Latvia, for example. It means that the new algorithm has tuned up the original community definition of the classical algorithms, finding communities which have an appropriate size, reduced overlapping and closer distances between clusters.

Finally, we can conclude that stable communities could be identified using community-finding algorithms in a social-based network. However, related to Eurovision dataset, the found communities tend to be formed by countries that share a common cultural history, boarders and even language roots.

6. Conclusions

To create an overlapped graph clustering algorithm we have focused our research in genetic algorithms. We have developed an algorithm where the number of clusters is adaptive instead of predefined. To guide the algorithm we have defined several fitness functions. In our solution the fitness functions have been inspired by complex network analysis specially focused in the clustering coefficient measure. The fitness functions also consider the quality of the clusters minimizing the distance between the elements which belong to a cluster, and maximizing the cluster centroid distance.

Our experimental findings show that, using this new approach, it is able to reach better results than classical community finding algorithms such as CPM or EBC. Comparing both algorithm versions against CPM and EBC we discover that the communities defined by the genetic algorithms are smaller than the communities found by CPM and EBC. It is important to observe that some communities generated by the genetic algorithms are almost contained in the communities generated by CPM. It means that the genetic algorithm has tuned up the original community definition of the classical algorithm. So, we can concluded that this new algorithm (both versions) finds communities that have an appropriate size, reduced overlapping and closer distances between clusters.

Finally some improvements can be made in the the algorithm. Our future work will be focused on complex network evolution. We are interested in dynamical network behaviour. Also, for the Eurovision dataset, other features such as geographical distances or historical behaviours could be included in future fitness functions to study the behaviour of the GCF algorithm.

7. Contributions

The main contribution of this Master Thesis is related to the definition of a new soft clustering approach. It is based on a genetic algorithm where a new encoding is designed to allow two main goals. First, the automatic adaptation of the number of communities that can be detected (K). Second, the definition of several fitness functions that guide the searching process using some measures extracted from graph theory.

Distance between nodes, distance between centres and clustering coeffi-

cient measures have been used to guide a genetic clustering algorithm with the goal of finding groups in a graph which minimize or maximize these measures. Although each of the measures can be used separately, the new genetic algorithm approach combines them using a hybrid function which gives different weights to each measure. This combination generates some problems specially when it is necessary to decide which measure is more relevant than the others. That is the reason why some experimental tests have been carried out to obtain the final weight for each measure, that were used in the hybrid fitness function.

Finally, once a particular encoding and several fitness functions have been designed, the new algorithm have been applied to the Eurovision Contest Song dataset. This well-known contest provides interesting data which has been deeply studied and analysed from different perspectives (social, political, economical and historical, among others) over the last decades. This data has been preprocessed and represented as a social network, and later used to study the behaviour of our new approach.

8. Published Works

The published works related to this Master Thesis are the following:

- **Gema Bello, Hector D. Menendez and David Camacho**. Using the Clustering Coefficient to guide a Genetic-based Community Finding Algorithm. *In Proceedings of the 12th International Conference on Data Extraction and Automated Learning (IDEAL 2011)*. 2011, Lecture Notes in Computer Science (LNCS),Vol. 6936, pp. 160-169.
- **Gema Bello, Raul Cajias, David Camacho**. Study on the Impact of Crowd-Based Voting Schemes in the Eurovision European Contest. *In Proceedings of 1st International Conference on Web Intelligence, Mining and Semantics (WIMS'11)*. ACM press, DOI: 10.1145/1988688.1988718.

Referencias

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

[2] James C. Bezdek, James Keller, Raghu Krisnapuram, and Nikhil Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing (The Handbooks of Fuzzy Sets)*, Springer, 1 edition, Mar. 2005.

[3] J. B. Macqueen, "Some methods of classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[4] G. N. Lance and W. T. Williams, "A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems," *The Computer Journal*, vol. 9, no. 4, pp. 373–380, Feb. 1967.

[5] M. Oussalah and Samia Nefti, "On the use of divergence distance in fuzzy clustering," *Fuzzy Optimization and Decision Making*, vol. 7, pp. 147–167, June 2008.

[6] Imre Derényi, Gergely Palla, and Tamás Vicsek, "Clique Percolation in Random Networks," *Physical Review Letters*, vol. 94, no. 16, pp. 160202–1 – 160202–4, Apr 2005.

[7] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, June 2005.

[8] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, June 2002.

[9] W.B. Langdon and R. Poli, "Evolving problems to learn about particle swarm and other optimisers," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, sept. 2005, vol. 1, pp. 81 –88 Vol.1.

[10] Alex A. Freitas, "A review of evolutionary algorithms for data mining," in *In: Soft Computing for Knowledge Discovery and Data Mining*, 2007, pp. 61–93.

[11] Daniel T. Larose, *Discovering Knowledge in Data*, John Wiley and Sons, 2005.

[12] Satu Elisa Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[13] M. Dehmer, *Structural Analysis of Complex Networks*, Birkhauser Boston, 2010.

[14] Derek Gatherer, "Comparison of eurovision song contest simulation with actual results reveals shifting patterns of collusive voting alliances," *Journal of Artificial Societies and Social Simulation*, vol. 9, no. 2, pp. 1, 2006.

[15] Alberto Ochoa Ortíz, Angel E. Muñoz Zavala, and Arturo Hernández Aguirre, "A hybrid system using pso and data mining for determining the ranking of a new participant in eurovision," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, New York, NY, USA, 2008, GECCO '08, pp. 1713–1714, ACM.

[16] Nino Boccara, *Modeling Complex Systems*, Springer, 1 edition, 2003.

[17] Daniel Fenn, Omer Suleman, Janet Efstathiou, and Neil Johnson, "How does europe make its mind up? connections, cliques, and compatibility between countries in the eurovision song contest," *Physica A: Statistical Mechanics and its Applications*, vol. 360, no. 2, pp. 576–598, February 2005.

[18] Marie Phillips., "It's time to make our minds up on europe.," *The Observer*, , no. Friday 12, March 2004.

[19] EBU, "http://www.ebu.ch/," October 2010.

[20] 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.

[21] G. Nathiya, S. C. Punitha, and M. Punithavalli, "An analytical study on behavior of clusters using k means, em and k* means algorithm," *CoRR*, vol. abs/1004.1743, 2010.

[22] Wesam Barbakh and Colin Fyfe, "Online clustering algorithms," *International Journal of Neural Systems*, vol. 18, no. 3, pp. 185–194, 2008.

[23] Amir Ahmad and Lipika Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data and Knowledge Engineering*, vol. 63, no. 2, pp. 503 – 527, 2007.

[24] Dharmendra K Roy and Lokesh K sharma, "Genetic kmeans clustering algorithm for mixed numeric and categorial data sets," *International Journal of Artificial intelligence and Applications(IJAIA)*, vol. 1, no. 2, pp. 23 – 28, 2010.

[25] Lin Wang, Minchu Jiang, Yinghua Lu, Minfu Sun, and Frank Noe, "A comparative study of clustering methods for molecular data.," *International Journal of Neural Systems*, vol. 17, no. 6, pp. 447 – 458, 2007.

[26] Wesam Barbakh and Colin Fyfe, "Clustering with reinforcement learning," in *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, Hujun Yin, Peter Tino, Emilio Corchado, Will Byrne, and Xin Yao, Eds., vol. 4881 of *Lecture Notes in Computer Science*, pp. 507–516. Springer Berlin / Heidelberg, 2007.

[27] Colin Fyfe and Wesam Barbakh, "Immediate reward reinforcement learning for clustering and topology preserving mappings," in *Similarity-Based Clustering*, Michael Biehl, Barbara Hammer, Michel Verleysen, and Thomas Villmann, Eds., vol. 5400 of *Lecture Notes in Computer Science*, pp. 35–51. Springer Berlin - Heidelberg, 2009.

[28] Colin Fyfe, "Topographic maps for clustering and data visualization," in *Computational Intelligence: A Compendium*, John Fulcher and L. Jain, Eds., vol. 115 of *Studies in Computational Intelligence*, pp. 111–153. Springer Berlin - Heidelberg, 2008.

[29] Jose Antonio Iglesias, Plamen Angelov, Agapito Ledezma, and Araceli Sanchis, "Human activity recognition based on evolving fuzzy systems.," *International Journal of Neural Systems*, vol. 20, no. 5, pp. 355 – 364, 2010.

[30] Sanghamitra Bandyopadhyay, "Genetic algorithms for clustering and fuzzy clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 6, pp. 524–531, 2011.

[31] Jianzhuang Liu and Weixin Xie, "A genetics-based approach to fuzzy clustering," in *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings*

of 1995 IEEE International Conference on, 1995, vol. 4, pp. 2233–2240 vol.4.

[32] Coley, *An Introduction to Genetic Algorithms for scientists and engineers*, World Scientific Publishing, 1999.

[33] M. Srinivas and L.M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 24, no. 4, pp. 656 –667, apr 1994.

[34] Riccardo Poli and William B. Langdon, "Backward-chaining evolutionary algorithms," *Artificial Intelligence*, vol. 170, no. 11, pp. 953 – 982, 2006.

[35] D. Doval, S. Mancoridis, and B. S. Mitchell, "Automatic Clustering of Software Systems using a Genetic Algorithm," in *IEEE Proceedings of the 1999 Int. Conf. on Software Tools and Engineering Practice (STEP'99)*, 1999, pp. 73–91.

[36] V. Fernandez, R. G. Martinez, R. Gonzalez, and L. Rodriguez, "Genetic algorithms applied to clustering," in *In Proceedings of the Winter Simulation Conference*, 1997, pp. 1307–1314.

[37] P. Pokorný and P. Dostál, "Cluster analysis and genetic algorithms," in *In: Management, Economics and Business Development in the New European Conditions*, 2008, pp. 1–9.

[38] Rowena M. Cole, "Clustering with Genetic Algorithms," M.S. thesis, Nedlands 6907, Australia, 1998.

[39] Jose Aguilar, "Resolution of the clustering problem using genetic algorithms," *International Journal of Computers*, vol. 1, no. 4, pp. 237 – 244, 2007.

[40] Lin Yu Tseng and Shiueng Bien Yang, "A genetic approach to the automatic clustering problem," *Pattern Recognition*, vol. 34, no. 2, pp. 415 – 424, 2001.

[41] U Maulik, "Genetic algorithm-based clustering technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455–1465, 2000.

[42] Liang-Dong Shi, Ying-Huan Shi, Yang Gao, Lin Shang, and Yu-BinN Yang, "Xcsc:: A novel approach to clustering with extended classifier system.," *International Journal of Neural Systems*, vol. 21, no. 1, pp. 79 – 93, 2011.

[43] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 1, pp. 218 –237, jan. 2008.

[44] K. Krishna and M. N. Murty, "Genetic K-means Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 29, no. 3, pp. 433–439, 1999.

[45] Wojciech and Kwedlo, "A clustering method combining differential evolution with the k-means algorithm," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1613 – 1621, 2011.

[46] K. Adamska, "Cluster analysis of genetic algorithms results," *Inteligencia Artificial, Revista Iberoamericana de IA*, vol. 9, no. 28, pp. 25–32, 2005.

[47] Julia Handl, Julia H, and Joshua Knowles, "Evolutionary multiobjective clustering," in *In Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature*. 2004, pp. 1081–1091, Springer.

[48] Curtis Huttenhower, Avi Flamholz, Jessica Landis, Sauhard Sahi, Chad Myers, Kellen Olszewski, Matthew Hibbs, Nathan Siemers, Olga Troyanskaya, and Hilary Coller, "Nearest Neighbor Networks: clustering expression data based on gene neighborhoods," *BMC Bioinformatics*, vol. 8, no. 1, pp. 250, 2007.

[49] Alexander N. Gorban and Andrei Zinovyev, "Principal manifolds and graphs in practice: From molecular biology to dynamical systems.," *International Journal of Neural Systems*, vol. 20, no. 3, pp. 219 – 232, 2010.

[50] Santo Fortunato, Vito Latora, and Massimo Marchiori, "Method to find community structures based on information centrality," *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, vol. 70, no. 5, pp. 056104, 2004.

[51] Mariá Cristina Vasconcelos Nascimento and André C. P. L. F. Carvalho, "A graph clustering algorithm based on a clustering coefficient for weighted graphs," *J. Braz. Comp. Soc.*, vol. 17, no. 1, pp. 19–29, 2011.

[52] Duncan J Watts, *Small worlds: The dynamics of networks between order and randomness*, Princeton University Press, Princeton, NJ, 1999.

[53] Erez Hartuv and Ron Shamir, "A clustering algorithm based on graph connectivity," *Information Processing Letters*, vol. 76, no. 4–6, pp. 175–181, 2000.

[54] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection.," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 74, no. 1 Pt 2, July 2006.

[55] P. Pons and M. Latapy, "Computing communities in large networks using random walks (long version)," *ArXiv Physics e-prints*, Dec. 2005.

[56] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review*, vol. 69, no. 026113, 2004.

[57] Aaron Clauset, M. E. J. Newman, , and Cristopher Moore, "Finding community structure in very large networks," *Physical Review E*, pp. 1– 6, 2004.

[58] Marek Lipczak and Evangelos Milios, "Agglomerative genetic algorithm for clustering in social networks," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, New York, NY, USA, 2009, GECCO '09, pp. 1243–1250, ACM.

[59] Keehyung Kim, RI (Bob) McKay, and Byung-Ro Moon, "Multiobjective evolutionary algorithms for dynamic social network clustering," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, New York, NY, USA, 2010, GECCO '10, pp. 1179–1186, ACM.

[60] Gad Yair, "Unite unite europe'the political and cultural structures of europe as reflected in the eurovision song contest," *Social Networks*, vol. 17, no. 2, pp. 147–161, 1995.

[61] Gad Yair and Daniel Maman, "The persistent structure of hegemony in the eurovision song contest," *Acta Sociologica*, vol. 39, no. 3, pp. 309–325, 1996.

[62] V Ginsburgh and A Noury, "The eurovision song contest. is voting political or cultural?," *European Journal of Political Economy*, vol. 24, no. 1, pp. 41–52, 2008.

[63] "Eurovision song contest," 2011, http://www.eurovision.tv.