tions, 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 [24,3], 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 optimize a particular feature of the data. In our 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 [39,59,61]. In our approach, we apply the new clustering technique to data that can be represented as a graph, trying to find groups whose nodes share similar graph-based features.

The proposed technique is based on genetic algorithm methods for clustering and 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. [11].

We also based our algorithm on network analysis techniques [18]. These techniques are usually based on graph theory methods, because a graph is the most common and straightforward representation for a network. 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 [18]. We have used a modified clustering coefficient that can be applied in directed and weighted graphs [10,68] to experimentally study how it could be employed in a genetic-based clustering process.

Distance between nodes, clustering coefficient and the weighted clustering coefficient measures can be used to guide a genetic clustering algorithm with the goal of finding groups in a graph (or weighted graph) which minimize or maximize these measures. Although each of the measures can be used separately, our 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.

Finally, once a particular encoding and several fitness functions were designed, we applied the new algorithm 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 [27,49]. This data has been preprocessed and represented as a social network.

The rest of the paper is structured as follows. Section 2 describes the related work concerning clustering, genetic algorithm and community-finding algorithms. Section 3 presents some basic definitions referred to graph concepts that are later used to design our genetic algorithm. Section 4 shows the two genetic algorithms (with a fixed value of K, and the K-adaptive version) and the encoding designed, the fitness functions and other characteristics of the algorithms, like crossover and mutation operators. Section 5 provides a description of the dataset used, the experimental setup of the algorithms and a complete experimental evaluation of them. Finally, in Section 6 the conclusions and some future research lines of work are presented.

2. Related Work

This section starts with a general introduction to clustering techniques. After this brief introduction, we focus our attention on how genetic algorithms have been applied to clustering techniques. Later, we present an overview of graph clustering methods based on spectral clustering techniques, and some current applications to social networks that uses the clustering coefficient. Finally, we show some community finding algorithm methods paying close attention to social network analysis.

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 [42] 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) [19], 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 [46].

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 [9]. Some other improvements of K-means algorithm are related to deal the different kind of data representation, for example, mixed numerical data [5] and categorical data [57] . And there are also some studies comparing methods with different datasets, for example, Wang et al. [67] 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 subsection 2.2.

Machine learning techniques have also been improved through the k-means algorithm, for example, reinforcement learning algorithms[8,26]; or using topological features of the data set [25,26] which can also be helpful for data visualization.

As we mentioned before, in our approach we are working with overlapping clustering instead of partitional clustering (which is the case of the original K-means). In overlapping clustering there are two main approaches[32]: 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 [34]. One of the first approximations was fuzzy K-means [51], which can also benefit from combining with a genetic approach [7,41]. In our problem (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 a large number of different domains, mainly related to optimization problems [13,17,58]. 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 [16,62]. These algorithms have also been used for general data and information extraction [24]. The operators of the genetic algorithms can also be modified. Some examples of these modifications can be found in (Poli and Langdon, 2006)[54] 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, [38] software systems [21], file clustering [22] and task optimization [53], 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)[15], 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)[32] which provides a complete updated review in evolutionary algorithms for clustering.

There are several methods using evolutionary approaches from different perspectives, for example: (Aguilar, 2007)[4] modifies the fitness considering cluster asymmetry, coverage and specific information of the studied case; (Tseng and Yang, 2001)[63] use a compact spherical cluster structure and a heuristic strategy to find the optimal number of clusters; (Maulik and Bandyopadhyay, 2000)[43] use the clustering algorithm for metric optimization trying to improve the cluster centre positions; (Shi et al., 2011)[60] 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)[17] 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)[36] replace the crossover of the algorithm using K-means as a search operator, and (Wojciech and Kwedlo, 2011)[69] 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)[2]. There are also other complete studies for multi-objective clustering in (Handl et al., 2004)[30] and for Nearest Neighbour Networks in (Huttenhower et al., 2007)[33].

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 [29] 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 [18,23,45,68].

A complete roadmap to graph clustering can be found in (Schaeffer, 2007)[59] 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 exhaustive analysis of connectivity methods can be found in (Hartuv and Shamir, 2000)[31]), amongst others. This roadmap also provides an overview of computational complexity from a theoretical and experimental point of view of the studied methods.

From previously described graph clustering techniques, a recent and really powerful ones are those based on the spectral clustering. Next subsection, describes briefly the basic concepts of this technique.

2.3.1. *Spectral Clustering*

Spectral clustering methods are based on a straightforward interpretation of weighted undirected graphs as can be seen in [65,6,48,44]. The Spectral clustering approach is based on a similarity graph which can be formulated in three different types (equivalents[65]) of graphs: the ϵ-neighbourhood graph (all the components whose pairwise distance is smaller than ϵ are connected), k-nearest neighbour graphs (the vertex v_i is connected with vertex v_j if v_j is among the k-nearest neighbours of v_i) and the fully connected graph (all points with positive similarity are connected with each other). The main problem is how to compute the eigenvector and the eigenvalues of the Laplacian matrix of this similarity graph. Some works focus on this problem: (von Luxburg, 2007)[65] presents the problem, (Ng et al.,2001)[48] applies an approximation to a specific case, and (Nadler et al.), [44] applies Fokker-Planck

operators to get better results.

The classical algorithms can be found in (von Luxburg, 2007)[65], and a particular modification of them which obtains good clustering results, similar to human selection, can be found in (Ng et al.,2001)[48].

The theoretical analysis of this observed good behaviour is justified using the perturbation theory [65,44], random walks and graph cut [65]. The perturbation theory is also the main approximation for the proofs about convergence of the modification of Fokker-Planck operators and explains, through the eigengap, the behaviour of spectral clustering.

Some of the main problems of spectral clustering are related to the consistency of the two typical methods used in the analysis: normalized and un-normalized spectral clustering. A deep analysis about the theoretical effectiveness of normalized clustering over un-normalized can be found in (von Luxburg, 2008)[66].

Part of the present work is inspired by spectral clustering because we use clustering techniques which analyse similarity graphs. Nevertheless, in our case we are using different methods such as the clustering coefficient measures to find the subgraphs, even to use the Laplacian matrix extracted from the similarity graph.

2.3.2. *Clustering Coefficient (CC)*

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 [18,45,68].

The present work is close to the network approach and has been developed over different kinds of graphs. In our case, we are working with undirected, directed and weighted graphs, and we apply the graph structure to the clustering coefficient using this new value to find clusters in the network [45].

2.4. *Community Finding Approach*

The main application of the communities approach are social networks. The clustering problem is more complex when applied to find communities in networks (subgraph identifications). A community can be considered as a subset of individuals with relatively strong, direct, and intensive connections [23] between them. Some algorithms such as Edge Betweenness Centrality (EBC) [28] or Clique Percolation Method (CPM) [20] have been designed to solve this problem following a deterministic process. Both algorithms have been applied to community classification and detection in (Bello et al., 2011)[50]. EBC algorithm [28] is based on finding the edges of the network which connect communities and removing them to determine a good definition of these communities. CPM [20] 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 [52].

In the initial study of the problem [11], we adopt an evolutionary approach based on the K-means algorithm applied to community finding approach. However, in the process of community finding problems, K-means algorithm cannot be directly applied because it does not allow overlapping. But our representation for communities in form of overlapped subgraphs does not need membership probability for a node. And our fitness design allows to extend the algorithm to consider overlapped groups.

Other approximations related to the finding-community problem can be found in (Reichardt and Bornholdt, 2006)[56] where different statistical mechanics for community detection are used. (Pons and Latapy, 2005)[55] uses random walks to compute the communities. However, we decided to use genetic algorithms because we are interested in optimization methods for tuning up the definition of our 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)[47], and metrics that can be used to find the structure of a community