Overlapped Community Detection in Complex Networks

Clara Pizzuti
ICAR-CNR
Via Pietro Bucci, 41C
87036 Rende (CS), Italy
pizzuti@icar.cnr.it

ABSTRACT

Extracting and understanding community structure in complex networks is one of the most intensively investigated problems in recent years. In this paper we propose a genetic based approach to discover overlapping communities. The algorithm optimizes a fitness function able to identify densely connected groups of nodes by employing it on the line graph corresponding to the graph modelling the network. The method generates a division of the network in a number of groups in an unsupervised way. This number is automatically determined by the optimal value of the fitness function. Experiments on synthetic and real life networks show the capability of the method to successfully detect the network structure.

Categories and Subject Descriptors

H.2.8 [**Database Managment**]: Database Applications — *Data Mining*; I.2.2 [**Artificial Intelligence**]: Automatic Programming; I.5.3 [**Computing Methodologies**]: Pattern Recognition—*Clustering*

General Terms

Algorithms

Keywords

Genetic Algorithms, Data Mining, Clustering, Complex Networks.

1. INTRODUCTION

Complex networks constitute an efficacious formalism to represent the relationships among the objects composing many real world systems. Collaboration networks, the Internet, the world-wide-web, biological networks, communication and transport networks, social networks are just some examples. Networks are modelled as graphs, where nodes represent the objects and edges represent the interactions among these objects. One of the main problems in the study of complex networks is the detection of *community structure*, i.e. the division of a network into groups (clusters or modules) of nodes having dense intra-connections, and sparse inter-connections. In the last few years many different approaches have been proposed to uncover community structure in networks [14, 22, 24, 5, 29, 2, 18] (a recent review can be found in [9]). However, as observed in [34], there are two main challenges in discovering communities. The first is that it is not known a priori the number of groups present in a given network. The second is that the communities may overlap, i.e. some nodes can belong to more than one cluster. The membership of an entity to many groups is very common in real world networks. For example, in a social network, a person may participate to many interest groups. Most of the known algorithms are not able to find overlapping communities. Only recently some methods capable to address this feature have been proposed [27, 25, 3, 34, 11, 12, 16]

In this paper we propose a new algorithm, named *GA-NET+*, to discover overlapped communities in networks by employing genetic algorithms. The method uses the concept of *community score* to measure the quality of the division in communities of a network, and tries to optimize this quantity by running the genetic algorithm on the *line graph* $L(G)$ of the graph G modelling the network. $L(G)$ represents the adjacency between the edges of G, thus it takes into account not only the links between a node and its direct connected neighbors, but also the higher-order interactions. A main advantage in using the line graph is that the partitioning of $L(G)$ obtained by *GA-NET+* corresponds to an overlapping graph division of G. The dense communities present in the network structure are obtained at the end of the algorithm by selectively exploring the search space, without the need to know in advance the exact number of groups. In fact, unlike many existing methods, the algorithm does not require the number of communities to find. This number is automatically determined by the optimal value of the *community score*. Experiments on synthetic and real life networks show the capability of the genetic approach to correctly detect communities with results comparable to the state-of-the-art approaches.

The paper is organized as follows. In the next section the concept of community is defined and the community detection problem is formalized. Section 3 describes the method, the genetic representation adopted and the variation operators used. In section 4 an overview of the main proposals in community detection algorithms is given. In section 5,

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

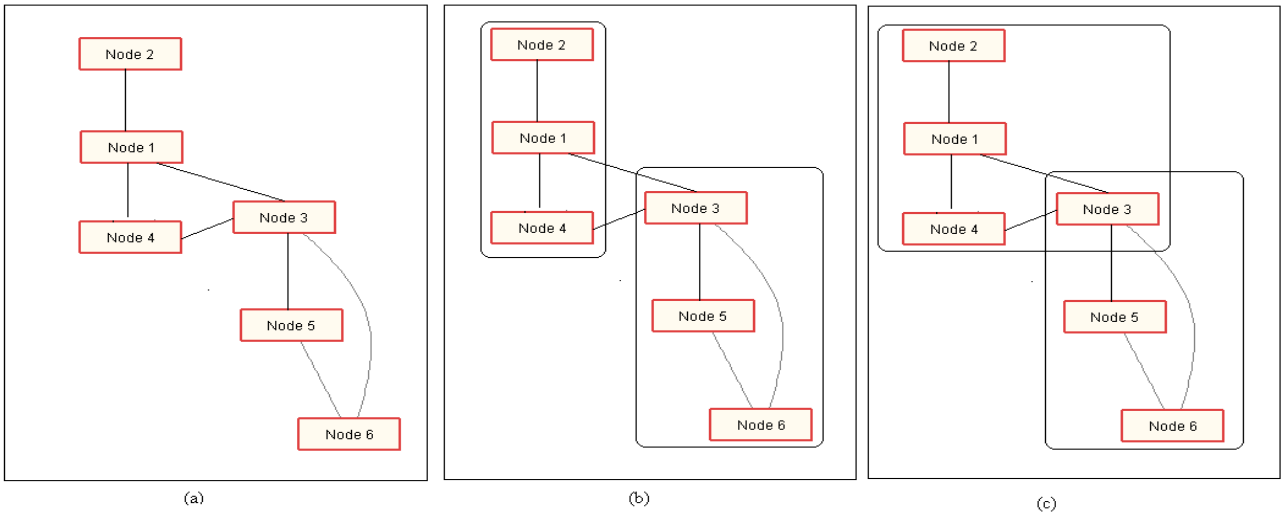

Figure 1: (a) A simple graph with six nodes; (b) a partition of the graph in two communities; (c) a division of the graph in two overlapping communities.

finally, the results of the method on synthetic and real life data sets are presented.

2. COMMUNITY DEFINITION AND DETECTION

A network $\mathcal{N}$ can be modelled as a graph $G = (V, E)$ where V is a set of objects, called nodes or vertices, and E is a set of links, called edges, that connect two elements of V. A community (also called cluster or module) in a network is a group of vertices (i.e. a sub-graph) having a high density of edges within them, and a lower density of edges between groups. This definition of community is rather vague and there is no general agreement on the concept of density. A more formal definition has been introduced in [29] by considering the degree k_i of a generic node i, defined as $k_i = \sum_j A_{ij}$, where A is the adjacency matrix of G. A is such that an entry at position (i, j) is 1 if there is an edge from node i to node j, 0 otherwise. Given a subgraph $S \subset G$, where node i belongs to, its degree with respect to S can be split as

$$k_i(S) = k_i^{in}(S) + k_i^{out}(S)$$

$k_i^{in}(S) = \sum_{j \in S} A_{ij}$ is the number of edges connecting i to the other nodes in S. $k_i^{out}(S) = \sum_{j \notin S} A_{ij}$ is the number of edges connecting i to the rest of the network. A subgraph S is a community in a strong sense if

$$k_i^{in}(S) > k_i^{out}(S), \ \forall i \in S$$

A subgraph S is a community in a weak sense if

$$\sum_{i \in S} k_i^{in}(S) > \sum_{i \in S} k_i^{out}(S)$$

Thus, in a strong community, each node has more connections within the community than with the rest of the graph. In a weak community the sum of the degrees within the subgraph is larger than the sum of degrees towards the rest of the network.

A quality measure of a community S that maximizes the in-degree of the nodes belonging to S and that implicitly minimizes their out-degree has been introduced in [28]. We now recall the definition of this measure, and then we show how it can be exploited to find overlapping communities. In the following, without loss of generality, the graph modelling a network is assumed to be undirected.

Let μ_i denote the fraction of edges connecting node i to the other nodes in S. More formally

$$\mu_i = \tfrac{1}{|S|} k_i^{in}(S)$$

where $\mid S \mid$ is the cardinality of S.

The *power mean of S of order r*, denoted as $\mathbf{M}(S)$ is defined as

$$\mathbf{M}(S) = \frac{\sum_{i \in S} (\mu_i)^r}{|S|}$$

Notice that, in the computation of $\mathbf{M}(S)$, since $0 \leq \mu \leq 1$, the exponent r increases the weight of nodes having many connections with other nodes belonging to the same module, and diminishes the weight of those nodes having few connections inside S.

The *volume* v_S of a community S is defined as the number of edges connecting vertices inside S, i.e the number of 1 entries in the adjacency sub-matrix of A corresponding to S, $v_S = \sum_{i,j \in S} A_{ij}$.

The *score* of S is defined as $score(S) = \mathbf{M}(S) \times v_S$. Thus the score takes into account both the fraction of interconnections among the nodes (through the power mean) and the number of interconnections contained in the module S (through the volume). The *community score* of a clustering $\{S_1, \ldots S_k\}$ of a network is defined as

$$\mathcal{CS} = \sum_{i}^{k} score(S_i)$$

The *community score* gives a global measure of the network division in communities by summing up the local score of

each module found. The problem of community identification can then be formulated as the problem of maximizing $\mathcal{CS}$.

Genetic algorithms have been used in [28] to partition a network in communities by optimizing the *community score*. The method uses the locus-based adjacency representation proposed in [26] and employed by [13, 21] for multiobjective clustering. In this graph-based representation an individual of the population consists of H genes $g_1, \ldots, g_H$, where H is the number of vertices, and each gene can assume allele values j in the range $\{1, \ldots, H\}$. Genes and alleles represent nodes of the graph $G = (V, E)$ modelling a network $\mathcal{N}$, and a value j assigned to the ith gene is interpreted as a link between the nodes i and j of V. This means that in the clustering solution found i and j will be in the same cluster. A decoding step, however, is necessary to identify all the components of the corresponding graph. The nodes participating to the same component are assigned to one cluster.

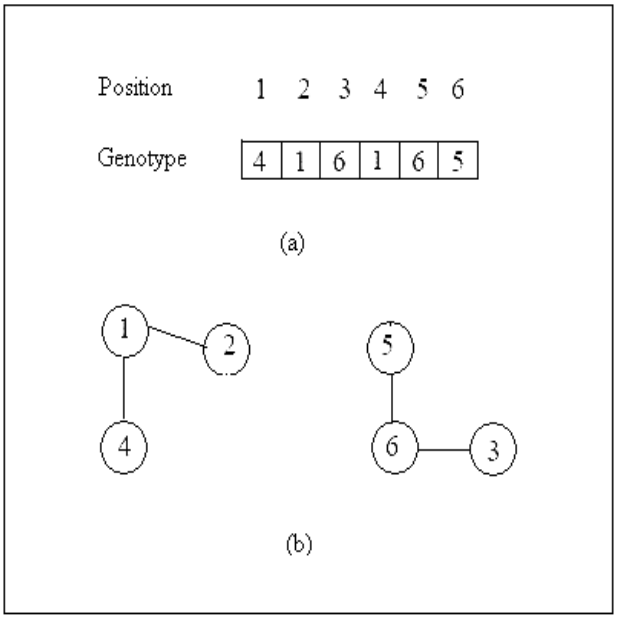

Figure 2: (a) The locus-based representation of a genotype relative to the graph of figure 1; (b) the graph based structure of the genotype.

Consider the simple graph shown in figure 1(a). It consists of six nodes and seven edges. The partition in two communities $\{1, 2, 4\}$ and $\{3, 5, 6\}$ is displayed in figure 1(b). The locus-based representation of the genotype corresponding to this solution can be seen in figure 2(a), and the decoded graph of the individual in the population corresponding to this genotype is shown in figure 2(b). A main advantage of this representation is that the number k of clusters is automatically determined by the number of components contained in an individual and determined by the decoding step. A drawback, however, is that each node can be connected to only one other node. This means that it is not possible to represent the participation of a vertex to multiple clusters.

For the graph of figure 1(a), a more natural division in two communities should include node 3 in both, as shown in figure 1(c). However, the locus-based adjacency representation, does not allow for multiple links among nodes. Thus, the graph can be partitioned, for example, like in figure 1(b).

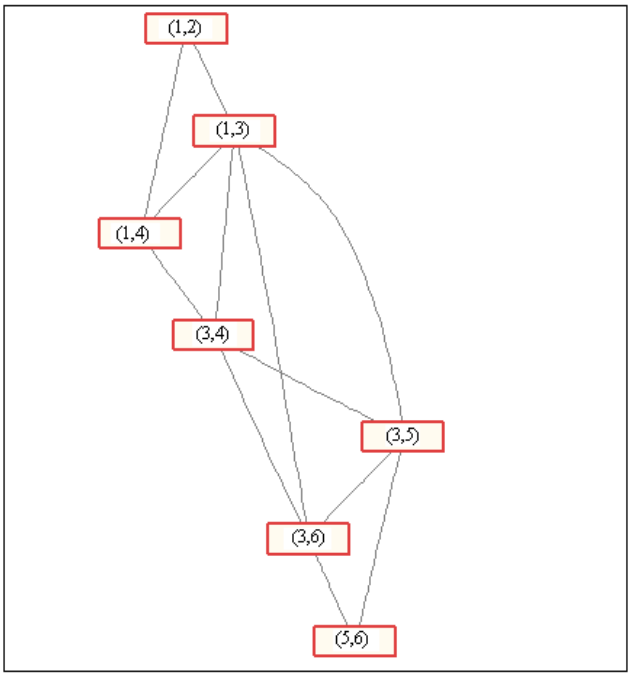

Figure 3: The line graph corresponding to the graph of figure 1.

In this paper an approach that allows to overcome this disadvantage and, at the same time, to exploit the benefits of the locus-based representation is proposed. Given a graph $G = (V, E)$ we propose to apply the genetic algorithm to the *line graph* of G. The *line graph* $L(G)$ of an undirected graph G is another graph $L(G)$ such that 1) each vertex of $L(G)$ represents an edge of G, and 2) two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G. Thus a line graph represents the adjacency between edges of G. The line graph of the graph contained in figure 1(a) is shown in figure 3. Notice that it contains seven nodes (one for each edge in G). Two nodes in $L(G)$ are connected if they have a node in common in G. Thus, for example, there is an edge between the nodes in $L(G)$ labelled (1,2) and (1,4) because they share node 1 of G.

The line graph is often used in graph theory and has a number of advantages. First, it can recover the original network thus maintaining all the information content. Second, it takes into account not only the direct neighbors of a node. Third, it is more highly structured of the original graph. In fact, its has been verified that the line graph has a higher clustering coefficient[1] of the original graph [27]. Further-

[1]The clustering coefficient has been defined by Watt in [32]. Given a node i, let n_i be the number of links connecting the k_i neighbors of i to each other. The clustering coefficient of i is $C_i = 2n_i/k_i(k_i - 1)$. n_i represents the number of triangles passing through i, and $k_i(k_i - 1)/2$ the number of possible triangles that could pass through node i. The clustering coefficient a graph is the average of the clustering coefficients of the nodes it contains.

more, the line graph clustering approach produces an overlapping graph partitioning of the original interaction graph, thus allowing nodes to be present in multiple communities. The approach of using the line graph to obtain overlapping modules is not new. Pereira et al. [27] adopted it to find overlapping modules in protein-protein interaction networks. However, the combination of the line graph with genetic algorithms has not been previously explored.

In the next section a detailed description of the algorithm is given.

3. ALGORITHM DESCRIPTION

In this section we give a description of the algorithm *GA-NET+*, and the variation operators used.

Given a network $\mathcal{N}$ and the graph $G = (V, E)$ modelling it, *GA-NET+* performs the following steps;

1. Compute the line graph $L(G)$ associated with G
2. create an initial population of random individuals whose length equals the number $L =\mid E \mid$ of edges of G
3. while termination condition is not satisfied, perform the following sub-steps
   (a) translate each individual $A = \{g_1, \ldots, g_L\}$ of the population in the corresponding individual $\overline{A} = \{g_1, \ldots, g_H\}$ of the original graph G
   (b) evaluate the fitness of the translated individuals
   (c) create a new population of individuals by applying the variation operators

The algorithm starts by generating a population initialized at random with individuals representing a partition in sub-graphs of the line graph $L(G)$ and *repaired* to produce $safe$ individuals, that is individuals generating connected sub-graphs of $L(G)$. This is realized by checking that an effective link exists between a gene at position i and the allele value j. This value is maintained only if the edge (i, j) exists. Otherwise, j is substituted with one of the neighbors of i. This guided initialization biases the algorithm towards a decomposition of the network in connected groups of nodes. An individual generating this kind of partitioning is called $safe$ because it avoids uninteresting divisions containing unconnected nodes. $Safe$ individuals improve the convergence of the method because the space of the possible solutions is restricted.

After that the fitness must be evaluated. As described in the previous section, we are interested in identifying a clustering that optimizes the *community score* because this guarantees highly intra-connected and sparsely inter-connected communities. The objective function is thus $\mathcal{CS} = \sum_i^k Q(S_i)$.

However, the fitness must be evaluated on the original graph G, instead of the line graph $L(G)$. Thus a translation from the individual A, representing a partitioning $\{C_1, \ldots C_k\}$ of $L(G)$, to the individual $\overline{A}$, representing an overlapping division $\{S_1, \ldots S_h\}$ of G, is necessary before fitness evaluation.

Regarding the variation operators, we used uniform crossover because it guarantees the maintenance of the effective connections of the nodes of the network in the child individual. In fact, because of the biased initialization, each individual in the population is safe, that is it has the property, that if a gene i contains a value j, then the edge (i, j) exists. Thus, given two safe parents, a random binary vector is created. Uniform crossover then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. The child at each position i contains a value j coming from one of the two parents. Thus the edge (i, j) exists. This implies that from two safe parents a safe child is generated.

The mutation operator that randomly changes the value j of a i-th gene causes a useless exploration of the search space, because of the same above observations on node connections. Thus the possible values an allele can assume are restricted to the neighbors of gene i. This *repaired* mutation guarantees the generation of a safe mutated child in which each node is linked only with one of its neighbors.

Before presenting the experiments, in the next section an overview of the main approaches to community detection is given.

4. RELATED WORK

Many different algorithms, coming from different fields such as physics, statistics, data mining, have been proposed to detect communities in complex networks [10, 14, 22, 24, 5, 29, 23, 2, 18]. These approaches, the most famous of which being that of Newman and Girvan [10, 24], divide a network in separated clusters of nodes, where each node can belong to only one group. Most of the real world networks, however, are constituted by overlapped communities of nodes. Thus, more recently, a growing interest in developing methods that allow overlapping among the discovered communities is rising [27, 25, 3, 11, 34, 12, 16, 15].

In the following a review of some of the proposals that detect overlapping communities is given.

One of the first approach has been proposed in protein-protein interaction domain and it is due to Pereira et al. [27]. They transform the interaction graph into the corresponding line graph, in which edges represent nodes and nodes represent edges, and then apply a known clustering algorithm on the line graph. The validity of the method has been established by the biological significance of the modules obtained.

The Clique Percolation Method of Palla et al. [7, 25] implemented in CFinder [1], finds k-clique percolation clusters, i.e. groups of nodes that can be reached via chains of k-cliques and the link in these cliques. The idea behind this approach is that a cluster can be interpreted as the union of small fully connected subgraphs that share nodes. A k-clique is a complete subgraph constituted by k nodes such that there is an edge for each pair of nodes. Two k-cliques are said adjacent if they have k-1 common nodes. A k-clique-community is then defined as the union of all the k-cliques that can be reached through adjacent k-cliques. The algorithm extracts all the maximal complete subgraphs, i.e. the maximal cliques. Then a clique-clique overlap matrix is built in which each entry contains the number of common nodes between the two corresponding cliques, and each diagonal entry is the clique size. The k-cliques-communities can be found by deleting every entry off the diagonal having a value less than k-1, and every diagonal entry less than k. The remaining separate components will be the k-cliques-communities. The parameter k has to be provided in input. Increasing k shrinks community size because nodes must belong to at least a clique of size k.

Lancichinetti et al. [16, 15] propose an algorithm to find communities one at a time. The method starts by picking a node X at random, and considering it as a community C. Then a loop over all the neighbors nodes of C is performed in order to choose the neighbor node to be added to C. The choice is done by computing a fitness function for each node, and augmenting C with the node having the highest value of the fitness. At this point the fitness of each node is recomputed, and if a node turns out to have a negative fitness, it is removed from C. The process stops when all the C nodes have a negative fitness. Once a community has been obtained, a new node is picked and the process restarts until all the nodes have been assigned to at least one group. Overlapping can be obtained since a node can be considered many times during the process. The fitness function adopted is defined as follows. Let C be a module, then

$$f_C = \frac{k_{in}^C}{(k_{in}^C + k_{out}^C)^\alpha}$$

where k_{in}^C and k_{out}^C are the total internal and external degrees of the nodes of C. α is a positive real-valued parameter controlling the size of the community. The role of α is analogous to our power mean parameter r. Higher values of both return denser communities, but the size diminishes. In the next section we show that our genetic algorithm approach is very competitive with respect to this one and that of Palla et. al. [25]

Regarding approaches to community detection based on Genetic Algorithms, only few proposals can be found in the literature [30, 31, 8]. None of them, however, contemplate the case of overlapping communities.

5. EXPERIMENTAL RESULTS

In this section the effectiveness of the approach on a synthetic data set is studied. Then the results obtained by *GA-NET+* are compared with those reported by Lancichinetti et al. in [15] on some real-worlds networks for which the partitioning in communities is known. In both cases we show that our genetic algorithm successfully detects the network structure and is competitive with the other approaches. The *GA-NET+* algorithm has been written in MATLAB, using the Genetic Algorithms and Direct Search Toolbox 2. The experiments have been performed on a Pentium 4 machine, 1800MHz, 1GB RAM. We employed standard parameters for the genetic algorithm, crossover rate 0.8, mutation rate 0.2, elite reproduction 10% of the population size, roulette selection function. The population size was 50, the number of generations 30.

Synthetic data set. In order to check the ability of our approach to successfully detect the community structure of a network, we use the benchmark proposed by Lancichinetti et al. [17], which is an extension of the classical benchmark proposed by Girvan and Newan in [10]. The network consists of 512 nodes divided into four communities of 128 nodes each. Every node has an average degree of 16 and shares a fraction α of links with the other nodes of its community, and $1-\alpha$ with the other nodes of the network. α is called the mixing parameter. When $\alpha \leq 0.5$ the neighbors of a node inside its group are more than the neighbors belonging to the other three groups, thus a good algorithm should discover them. We generated 10 different networks for values of α ranging from 0.2 to 0.5, and used the *Normalized Mutual Information* to measure the similarity between the true partitions and the detected ones.

The *Normalized Mutual Information* is a similarity measure coming from Information Theory [20] proved to be reliable by Danon et al. [6]. The original formulation, however, does not contemplate the possibility of having communities sharing nodes. In Lancichinetti et al. [17] an extension to deal with overlapping modules is presented. In the following we summarize the extension introduced in [17]. Given two divisions A and B of a network in communities, with respectively $\mid A \mid$ and $\mid B \mid$ clusters, to measure the distance between two clusterings A and B, it is necessary to measure the amount of information needed to recover A, once B is known. The normalized mutual information $N(A,B)$ is defined as :

$$N(A,B) = 1 - \frac{1}{2}(H(A|B)_{norm} + H(B|A)_{norm})$$

where

$$H(A|B)_{norm} = \frac{1}{\mid A \mid} \sum_{k=1}^{|A|} \frac{H(A_k|B)}{-p_A log(p_A)}$$

with p_A the fraction of nodes contained in the clustering A. Note that $-p_A log(p_A)$ is the entropy of A. $H(A_k|B)$ is the conditional entropy of a module $A_k \in A$ with respect to the clustering B, and it is computed as follows:

$$H(A_k|B) = \begin{cases} min_{l \in \{1,2,\ldots,|B|\}} H(A_k|B_l) \\ \quad if\ p_{11} + p_{00} > p_{10} + p_{01}\ (see\ below) \\ \\ -p_{A_k} log(p_{A_k}) \quad otherwise \end{cases}$$

$H(A_k|B_l)$ is the amount of information needed to infer the module A_k, given a certain module B_l. The constraint is necessary in order to avoid to choose a cluster B_l in B similar to the complementary of A_k, instead of A_k. $H(A_k|B_l)$ is calculated as:

$$H(A_k|B_l) = (-p_{11}log(p_{11}) - p_{10}log(p_{10}) - p_{01}log(p_{01})$$

$$-p_{00}log(p_{00})) - (-p_{B_l}log(p_{B_l}))$$

where p_{11} is the fraction of nodes shared by the two clusters A_k and B_l, p_{10} is the fraction of nodes belonging to A_k but not to B_l, p_{01} is the fraction of nodes belonging to B_l but not to A_k, and p_{00} is the fraction of nodes contained in neither A_k nor B_l. $H(B|A)_{norm}$ is computed in an analogous way.

When $N(A,B) = 1$ it means that the two clusterings are identical. Since the benchmark networks we use to validate how well our approach recover the original structure are such that each node is labelled with the class number of only one community, it is not possible to obtain a value of the normalized mutual information equal to 1 for the results obtained by *GA-NET+*. However, the higher the value of the normalized mutual information obtained, the better the solution found.

Figure 4 shows the normalized mutual information, averaged over the 10 runs, for different values of the exponent r when the mixing parameter α increases from 0.2 to 0.5. The figure points out that, for low values of r, *GA-NET+* is able to recover almost 70% of community structure only

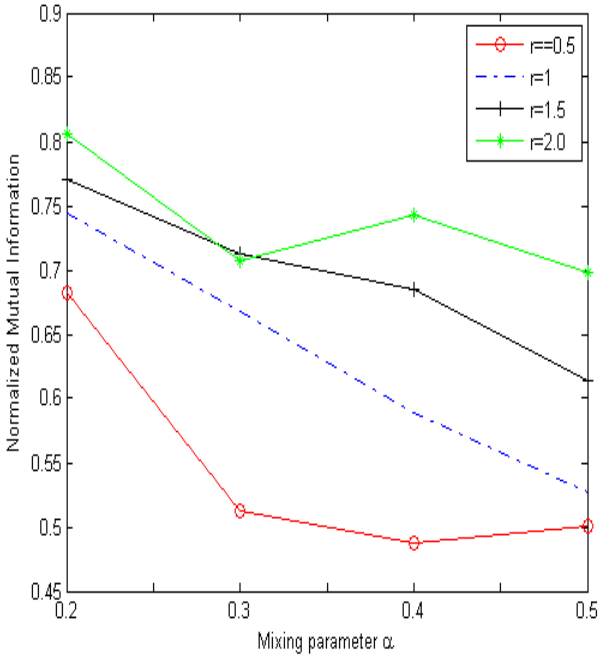

Figure 4: Normalized mutual information obtained by *GA-NET+* on the synthetic network for different values of the exponent r.

when the fuzziness modules is low ($\alpha = 0.2$). When $r = 2$, instead, the algorithm is able to recover the true community structure in almost more than 70% of cases even for $\alpha = 0.5$, i.e. each node has half of the links inside its community and the other half with the rest of the network. This result is very interesting because a high mixing parameter increases the network fuzziness, thus it is rather difficult to identify the hidden groups, being the communities mixed with each other.

Real-life data set. We now show the application of *GA-NET+* on three real-world networks, the *Zachary's Karate Club* , the *Bottlenose Dolphins*, and *American College Football*, well studied in the literature, and compare our results with those obtained by Lancichinetti et al. in [15] and Palla et al. [25], reported in [15].

The *Zackary's Karate Club* network was generated by Zachary [33], who studied the friendship of 34 members of a karate club over a period of two years. During this period, because of disagreements, the club divided in two groups almost of the same size. The social network of 62 bottlenose dolphins living in Doubtful Sound, New Zealand, was compiled by Lusseau [19] from seven years of dolphins behavior. A tie between two dolphins was established by their statistically significant frequent association. The network split naturally into two large groups, the number of ties being 159. The last example is the American College Football network [10] which comes from the United States college football. The network represents the schedule of Division I games during the 2000 season. Nodes in the graph represent teams and edges represent the regular season games between the two teams they connect. The teams are divided in conferences. The teams on average played 4 inter-conference matches and 7 intra-conference matches, thus teams tend to play between members of the same conference. The network consists of 115 nodes and 616 edges grouped in 12 teams.

For each network, we run *GA-NET+* 10 times and computed the average normalized mutual information over these 10 runs. As regards the values of the normalized mutual information of the other two methods, we took the results reported in [15], where the authors compare their method with that of [25]. Figure 5 clearly shows the very good performance of *GA-NET+* with respect to both the other two approaches.

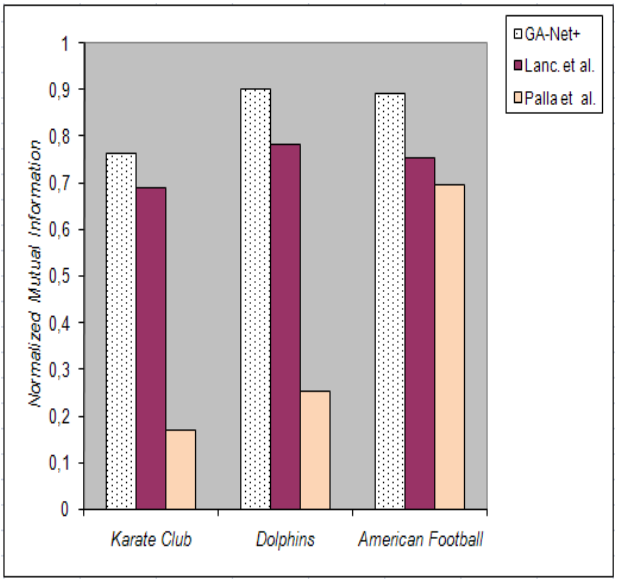

Figure 5: Comparison of GA-NET+, Lancichinetti et al., and Palla et al. relative to the extended Normalized Mutual Information for Karate club, Dolphins, and American College Football networks.

In fact, over 10 runs, *GA-NET+* obtained an average normalized mutual information of 0.7635, 0.90071, 0.8913 on the *Zachary's Karate Club*, the *Bottlenose Dolphins*, and *American College Football* networks, respectively. On the other hand Lancichinetti et al. obtained 0.690, 0.781, and 0.754, while Palla et al. 0.170, 0.254, and 0.697, respectively.

To conclude, figure 6 displays the network division generated by Zackary in two distinct groups, identified by circles and triangles of different colors, and four, out of the eleven overlapped groups obtained by *GA-NET+*. The figure has been reproduced by using the *NetDraw* software [4]. The figure points out that the sub-graphs sharing nodes are significant. Consider, for example, the module containing nodes $\{1, 5, 6, 7, 11, 17\}$. Nodes $\{5, 6, 7, 11, 17\}$ are strictly connected to each other and four out of five of them are linked to only node 1. Thus the participation of node 1 in this group is meaningful. On the other hand, node 1 is a central node in the network because its degree is much higher than the others (in the literature these kind of nodes are often called "hub"). This naturally candidates it to belong to many different groups. Analogously, in the community composed by the nodes $\{24, 25, 26, 28, 32\}$, node 32 can belong to more than one community. It is worth noting that node 10, classified by Zackary in the community on the right,

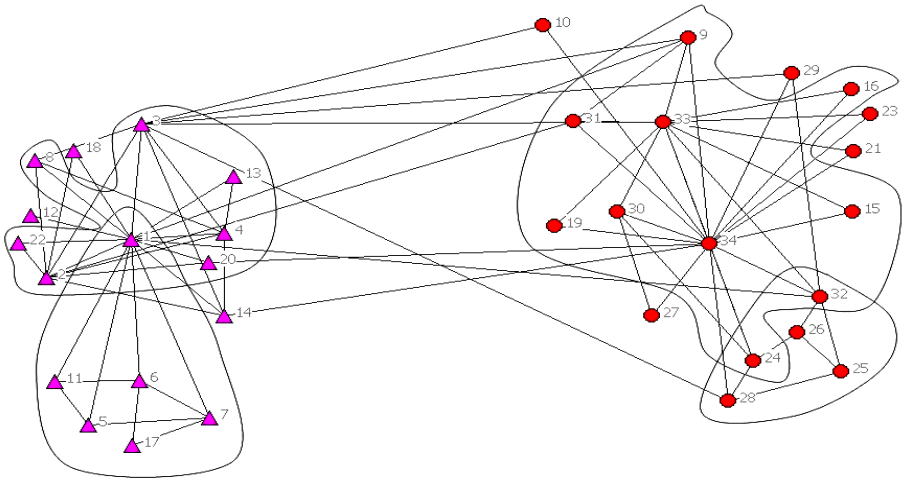

Figure 6: Overlapped Communities found by the genetic based method *GA-NET+*

has only one link with both the two Zackary's communities. *GA-NET+* assigned node 10 to two groups. One group is formed with nodes all belonging to the community on the right, the other group includes also nodes from the group on the left, namely nodes 1, 3, and 14. This choice is plausible because the module found contains also node 9, connected to both nodes 1 and 3, and node 34, linked to node 14.

The results obtained show the capability of genetic algorithms to effectively deal with community identification in networks.

6. CONCLUSIONS

The paper presented a genetic algorithm for detecting overlapping communities in complex networks. The approach makes use of the line graph to extract all the dense communities present in the network by selectively exploring the search space, without the need to know in advance the exact number of groups. Experiments on synthetic and real life networks showed the capability of the genetic approach to correctly detect communities with comparable results with state-of-the-art approaches. Future research will aim at applying multi-objective optimization to improve quality results.

7. REFERENCES

[1] Balázs Adamcsek, Gergeley Palla, Ille J. Farkas, Imre Derényi, and Tamás Vicsek. Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.

[2] A. Arenas and A. Diaz-Guilera. Synchronization and modularity in complex networks. *European Physical Journal ST*, 143:19–25, 2007.

[3] Jeffrey Baumes, Mark K. Goldberg, Mukkai S. Krishnamoorthy, Malik Magdon-Ismail, and Nathan Preston. Efficient identification of overlapping communities. In *IEEE International Conference on Intelligence and Security Informatics, ISI'05*, pages 27–36, 2005.

[4] S. P. Borgatti. Netdraw 1.0 : Network visualization software. harvard: Analytic technologies. 2002.

[5] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[6] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics*, P09008, 2005.

[7] I. Derenyi, Gergely Palla, and Tamás Vicsek. Clique percolation in random networks. *Physical Review Letters*, 94(16):160202, 2005.

[8] Aykut Firat, Sagit Chatterjee, and Mustafa Yilmaz. Genetic clustering of social networks using random walk. *Computational Statistics and Data Analysis*, 51:6285–6294, 2007.

[9] Santo Fortunato and Claudio Castellano. Community structure in graphs. *arXiv:0712.2716v1 [physics.soc-ph]*, 2007.

[10] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proc. National. Academy of Science. USA 99*, pages 7821–7826, 2002.

[11] Steve Gregory. An algorithm to find overlapping communities structure in networks. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'07)*, pages 91–102, 2007.

[12] Steve Gregory. A fast algorithm to find overlapping communities in networks. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'08)*, pages 408–423, 2008.

[13] Julia Handle and Joshua Knowles. An evolutionary approach to multiobjective clustering. *IEEE transactions on Evolutionary Computation*, 11(1):56–76, 2007.

[14] John E. Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Natural communities in large linked networks. In *Proc. International Conference on Knowledge*

Discovery and Data Mining (KDD'03), pages 541–546, 2003.

[15] Andrea Lancichinetti, Santo Fortunato, and János Kertés. Community spectroscopy in complex networks. *technical report*, 2008.

[16] Andrea Lancichinetti, Santo Fortunato, and Janos Kertész. Detecting the overlapping and hierarchical community structure of complex networks. *arXiv:0802.1281v1 [physics.soc-ph]*, 2008.

[17] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. New benchmark in community detection. *arXiv:0805.4770v2 [physics.soc-ph]*, 2008.

[18] S. Lozano, J. Duch, and A. Arenas. Analysis of large social datasets by community detection. *European Physical Journal ST*, 143:257–259, 2007.

[19] D. Lusseau. The emergent properties of dolphin social network. *Biology Letters, Proc. R. Soc. London B (suppl.)*, 2003.

[20] D. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, UK, 2002.

[21] N. Makate, M. Miki, T. Hiroyasu, and T. Senda. Multiobjective clustering with automatic k-determination fot large-scale dara. In *Proc. of the Int. Genetic and Evolutionary Computation Conference (GECCO'07)*, pages 861–868, 2007.

[22] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[23] M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA 103*, pages 8577–8582, 2006.

[24] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[25] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[26] Y.J. Park and M.S. Song. A genetic algorithm for clustering problems. In *Proc. of 3rd Annual Conference on Genetic Algorithms*, pages 2–9, 1989.

[27] J. B. Pereira, A.J. Enright, and C.A. Ouzounis. Detection of functional modules from protein interaction networks. *Proteins: Structure, Fuctions, and Bioinformatics*, (20):49–57, 2004.

[28] Clara Pizzuti. GA-NET: a genetic algorithm for community detection in social networks. In *Proc. of the 10th Intenational Conference on Parallel Problem Solving from Nature (PPSN 2008)*, pages 1081–1090, 2008.

[29] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proc. Natl. Acad.Sci. USA (PNAS'04)*, 101(9):2658–2663, 2004.

[30] Mursel Tasgin and Aluk Bingol. Communities detection in complex networks using genetic algorithms. In *Proc. of the European Conference on Complex Systems (ECSS'06)*, 2006.

[31] Mursel Tasgin, Amac Herdagdelen, and Aluk Bingol. Communities detection in complex networks using genetic algorithms. *oai:arXiv.org:0711.0491v1 [physics.soc-ph]*, 2007.

[32] D. J. Watt. *Small worlds*. Princeton University Press, 1999.

[33] W.W Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[34] Shihua Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Phisica A*, 374:483–490, 2007.