World Wide Web (2013) 16:545–565
DOI 10.1007/s11280-012-0174-4

Mesoscopic analysis of networks with genetic algorithms

Clara Pizzuti

Received: 15 July 2011 / Revised: 17 May 2012 /
Accepted: 22 May 2012 / Published online: 8 June 2012

Abstract The detection of communities is an important problem, intensively investigated in recent years, to uncover the complex interconnections hidden in networks. In this paper a genetic based approach to discover communities in networks is proposed. The algorithm optimizes a simple but efficacious fitness function able to identify densely connected groups of nodes with sparse connections between groups. The method is efficient because the variation operators are modified to take into consideration only the actual correlations among the nodes, thus sensibly reducing the search space of possible solutions. Experiments on synthetic and real life networks show the ability of the method to successfully detect the network structure.

Keywords genetic algorithms · data mining · clustering · community detection · networks

1 Introduction

The suitability of networks to represent many real world systems has given an impressive spur to the recent research area of complex networks. Collaboration networks, biological networks, communication and transport networks, the Internet, and the World-Wide-Web [25] are just some examples. Networks, in general, are constituted by a set of objects and by a set of interconnections among these objects. In social networks, for example, the objects are people and the connections represent social relations, such as common interests, friendship, religion, and so on. Members of networks and relationships between them can be modeled as a graph of nodes and edges. Each participant is denoted by a distinct node, and interactions are represented by edges connecting two objects. Complex networks can be analyzed

C. Pizzuti (✉)
Institute for High Performance Computing and Networking (ICAR),
Italian National Research Council (CNR), Via P. Bucci 41/C, 87036 Rende (CS), Italy
e-mail: pizzuti@icar.cnr.it

at different levels of granularity. The node level is the smallest scale to study. At this level the node degree can give valuable information on the role played by the objects participating in the network. More interestingly, the community or sub-graph level investigates the division of a network into groups (also called clusters or modules) having dense intra-connections, and sparse inter-connections, thus delivering a *mesoscopic* description of a network where the elements are the communities and not the nodes. This partitioning is typical to many networks, thus the study of *community structure* can give important information and useful insights to understand how the structure of ties affects individuals and their relationships. In fact, members of a community interact with each other, they share information, and can have a remarkable influence on the behavior of the other objects of the community.

The problem of community detection has been receiving a lot of attention in the last few years, and many different approaches have been proposed [1, 3, 4, 10, 17, 22, 23, 26, 29, 31–33, 37, 39].

In this paper an algorithm, named *GA-Net*, to discover communities in networks by employing *Genetic Algorithms (GAs)* [14] is proposed. The approach introduces the concept of *community score* to measure the quality of a network partitioning in communities, and tries to optimize this quantity by running the genetic algorithm. All 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. Specialized variation operators allow to reduce the space of the possible solutions thus improving the convergence of the algorithm. The method requires an input parameter that biases the search towards a different number of communities. The number of communities found is 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 state-of-the-art approaches.

The paper is organized as follows. In the next section an overview of the main proposals of community detection algorithms is given. Section 3 provides the necessary background to formalize the problem and defines the quality metric employed to detect communities. In Section 4 a description of the method along with the representation adopted and the variation operators used are provided. In Section 5 the results of the method on synthetic and real life data sets are presented. Section 6 discusses the advantages of using *GA-Net*. Finally, Section 7 concludes the paper.

2 Related work

Many different algorithms have been proposed to detect communities in complex networks [1, 3, 4, 7, 11, 13, 17, 22, 23, 26, 27, 29, 31–33, 35, 37, 39]. In the following we review some of the most known algorithms. Overviews of community identification methods in complex networks can be found in [6, 8, 10].

One of the most famous algorithm has been presented by Newman and Girvan in [11, 29]. The method is a divisive hierarchical clustering method based on an iterative removal of edges from the network. The edge removal splits the network in communities. An agglomerative, instead of a divisive, hierarchical algorithm that optimizes the concept of *modularity*, introduced in [29], is presented by Newman in [26]. The modularity is the fraction of edges inside communities minus the expected value of

the fraction of edges, if edges fall at random without regard to the community structure. Values approaching 1 indicate strong community structure. Thus the algorithm computes the modularity of all the clusters obtained by applying the hierarchical approach, and returns as result the clustering having the highest value of modularity. A faster version of the method, based on the same strategy, is described in [4].

Recently, some studies [9] have indicated that the optimization of modularity has a main disadvantage. It can fail in finding communities smaller than a fixed scale, even if these modules are well defined. The scale depends on the total size of the network and the interconnection degree of the modules. This resolution limit can constitute a weakness for all those methods whose objective to optimize is modularity.

Wakita and Tsurumi [37] improved the method of [4] by identifying the cause of inefficiency of this latter agglomerative method in the strategy adopted to merge communities. To this end they introduced three metrics that try to balance the size of the communities to be merged. The modularity criterion enriched with these metrics allows for a sensible improvement of the algorithm efficiency.

Radicchi et al. [32] proposed a divisive hierarchical algorithm to identify communities based on the concept of *edge-clustering coefficient*, defined in analogy with the node clustering coefficient.[1] The edge-clustering coefficient is the number of triangles an edge participates, divided by the number of triangles it might belong to, given the degree of the adjacent nodes. Their algorithm works like that of Newman and Girvan, but it is faster. The main difference is that instead of choosing to remove the edge with the highest edge betweenness, the removed edges are those having the smallest value of edge-clustering coefficient. However, a quantitative measure for the evaluation of the dendrograms generated by the hierarchical approach is not defined. Thus the choice of a solution with respect to another must rely on the intuitive concept of community that a user has.

Pons and Latapy [31] introduced an agglomerative hierarchical algorithm to compute the community structure of a network. The algorithm starts from a partition of the graph in which each node is a community, and then merges the two adjacent communities (i.e. having at least a common edge) that minimize the mean of the square distances between each vertex and its community. The distances between communities are recomputed and the previous step is repeated until all the nodes belong to the same community. In order to decide the best partitioning to choose, the modularity criterion of Girvan and Newmann is adopted.

Blondel et al. [3] presented a method that partitions large networks based on the modularity optimization. The algorithm consists of two phases that are repeated iteratively until no further improvement can be obtained. At the beginning each node of the network is considered a community. Then, for each node i, all its neighbors j are considered and the gain in modularity of removing i from its community and adding it to the j community is computed. The node is placed in the community for which the gain is positive and maximum. If no community has positive gain, i remains in its original group. This first phase is repeated until no node move can improve the

[1] The clustering coefficient has been defined by [38]. 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.

modularity. The second phase builds a network where the communities obtained are considered as the new nodes and a link between two communities a, b exists if there is an edge between a node belonging to a and a node belonging to b. The network can be weighted, in such a case the weight of the edge between a and b is the sum of the weights of the links between nodes of the corresponding communities. At this point the method can be reiterated until no more changes can be done to improve modularity. The method is very accurate, however, it is unable to detect modules at a particular scale.

Approaches to community detection based on Genetic Algorithms can be found in [7, 13, 22, 35]. In [35] the authors present a genetic algorithm that uses as fitness function the network modularity proposed by Newmann and Girvan. An individual is constituted by N genes, where N is the number of objects. The ith gene corresponds to the ith node, and its value is the community identifier of node i. They use a non standard one-way crossover operation in which, given two individuals A and B, a community identifier j is chosen at random, and the identifier j of the nodes $j_1, \ldots j_h$ of A is transferred to the same nodes of B.

Gog et al. [13] proposed a collaborative evolutionary algorithm that uses also the modularity as fitness function to optimize. The main novelty of this approach is that each individual is endowed with the knowledge about the best potential solution already obtained during the search process, and the value of its best ancestor. The sharing of this information helps the method to find significative community structure. Both the two above methods could fail to uncover community structure when the network contains modules satisfying the conditions of the limit resolution property stated in [9].

A different approach is described in [7] where a random walk distance measure between graphs is integrated in a genetic algorithm to cluster networks. The representation used is the k-medoids, where each cluster center is represented by one of the nodes of the network. The fitness function tries to minimize the sum of all the pair-wise distances between nodes. The main limitation of this approach is that the number k of clusters must be known in advance.

An agglomerative clustering method based on Genetic Algorithms has been proposed by Lipczak et al. [22]. In this approach each individual represents a single community, instead of the whole clustering solution. Two fitness functions are considered. The former considers the normalized cut, i.e. it assumes that a graph is divided into two disjoint sets A and B, and defines the score of this division as the fraction of all the connections between A and B with respect to the number of connections involving A and B separately. The other fitness function is essentially the modularity of Girvan and Newman. The authors compared their approach with $UPGMA$ [34], a well known hierarchical method, and showed the good performance of their approach. A main difference of this approach with respect to the other GA-based methods is the representation used. In fact Lipczak et al. proposed to represent each cluster with a chromosome, thus a solution is represented by the whole population. The motivation of this choice, as stated from the authors, was to reduce the size of an individual and the fitness computational cost. This kind of representation implies that the method, in order to obtain a partitioning of the network in k clusters, needs to use a population of k individuals. Thus the method must be executed for an increasing number of clusters, and thus a population of increasing size, to find the best result. Another drawback comes from the variable length of the individuals. In

order to perform crossover, a mapping to the fixed-length representation of the two individuals involved in the crossover operation is needed. The mapping of a parent adds null genes in places of genes present in the other parent. This strategy partially destroys the objective of reducing the size of individuals.

Recently, the problem of community detection has been tackled by means of *particle swarm optimization (PSO)* [40]. In this approach a fixed number of particles are deployed onto the search space and move according to their velocity vector. Each particle has size equal to the number of nodes of the network and represents a partitioning. At each iteration, the fitness of particles is computed, and that having the best fitness is stored as the current best solution. The fitness function adopted is the modularity. The particles then update their position and velocity vector, and repeat the same steps until the stop condition is not reached.

3 Community detection problem

A network $\mathcal{N}$ can be modeled as a graph $G = (V, E)$ where V is a set of $n =\mid V \mid$ objects, called nodes or vertices, and E is a set of $m =\mid E \mid$ links, called edges, that connect two elements of V. In the following, without loss of generality, the graph modeling a network is assumed to be undirected. A community 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. In [8] it is observed that a formal definition of community does not exist because this definition often depends on the application domain. In this paper we assume the intuitive definition given by Radicchi et al. [32] of weak community. A weak community is interpreted as a set of nodes having the total number of intra-connections higher than the number of inter-connections among different communities. The partitioning of the graph G, modeling a network $\mathcal{N}$, in k weak communities $\{S_1, \ldots, S_k\}$, can be transformed into that of partitioning the adjacency matrix A of G in k sub-matrices, such that the sum of densities of the sub-matrices is maximized.

A naive density measure for a sub-matrix of n rows/columns is the number of ones (i.e. interactions) it contains. The higher the number of ones, the more connected the n nodes. However, counting the number of interactions does not give any information about the interconnections among the nodes. A quality measure of a community S that maximizes the in-degree of the nodes belonging to S can be defined as follows.

$$score(S) = \frac{\sum_{i \in S} \left(\frac{1}{|S|} \sum_{j \in S} A_{ij}\right)^r}{|S|} \times \sum_{i,j \in S} A_{ij}$$

where $\mid S \mid$ is the cardinality of S, $\frac{1}{|S|}\sum_{j \in S} A_{ij}$ is the fraction of edges connecting node i to the other nodes in S, and $\sum_{i,j \in S} A_{ij}$ is the double of 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.

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

In order to better explain the meaning of community score, let S be a group of nodes having n_s nodes and m_s edges, i.e. $m_S = \{(u, v) \mid u \in S, v \in S\}$. Note that $\sum_{i,j \in S} A_{ij} = 2m_S$. When $r = 1$,

$$score(S) = \frac{\sum_{i,j \in S} A_{ij}}{|S|^2} \times \sum_{i,j \in S} A_{ij} = \frac{2m_S}{n_S^2} \times 2m_S = \left(\frac{2m_S}{n_S}\right)^2$$

Thus the score of a community measures the density of the edges with respect to the number of nodes. This implies that, if the community S has a high density of edges, and it is contained in another community $\overline{S}$ of lower density, the score of S can be higher than that of $\overline{S}$, and the larger community could be split in many smaller communities. Figure 1 shows the scores of communities constituted by an increasing number of nodes $n_S = 8, 16, 32, 64, 128, 256$ when the number of edges augments from 2 to the maximum number of possible edges $n_S \times (n_S - 1)/2$. The figure points out that smaller and highly dense clusters can reach a score higher than larger, but less dense, groups of nodes. For example, consider the score of an 8-nodes community of maximum density equal to 1, i.e. a clique of 8 nodes. Its score, which is 0.875, is higher than the score of a community of 16 nodes having edge density less than 0.95. In the latter case, in fact, the score would be ≤ 0.8461. Thus the 8-clique is preferred over the 16-nodes cluster. This behavior is emphasized when $r > 1$ and damped when $r < 1$, thus r controls the size of a community S. In fact, since the quantity $\frac{1}{|S|} \sum_{j \in S} A_{ij} \leq 1$, the higher the value of r, the lower the value of *score(S)* and, consequently, the lower the value of $\mathcal{CS}$. Thus, increasing r biases $\mathcal{CS}$ towards

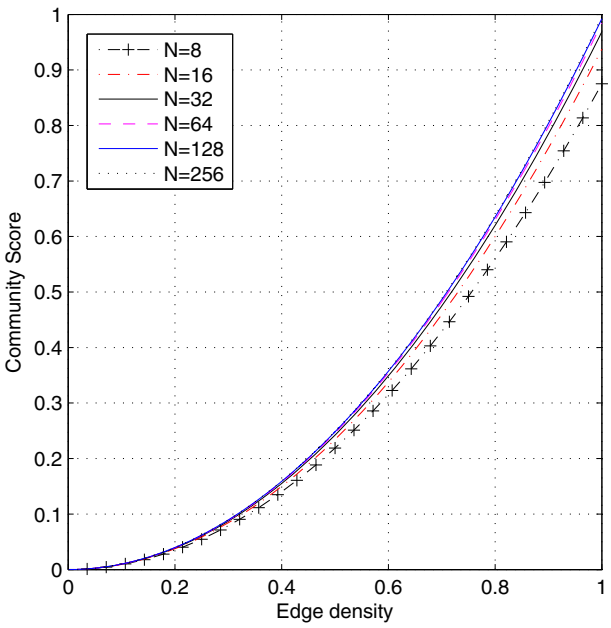

Figure 1 Scores of communities with $n_S = 8, 16, 32, 64, 128, 256$ and increasing number of edges.

matrices containing a low number of zeroes but of lower volume, and communities of smaller size are found. Its value can be set on the base of the resolution level desired. In the experimental result section we show that varying the value of r allows for an analysis of the network at different hierarchical levels.

4 Genetic representation and operators

Genetic Algorithms [14] are a class of adaptive general-purpose search techniques inspired by natural evolution. They have been proposed by Holland [16] in the early 1970s as computer programs that simulate the evolution process in nature. In the last few years genetic algorithms revealed competitive alternative methods to traditional optimization and search techniques and they have been applied to many problems in diverse research and application areas such neural nets evolution, planning and scheduling, machine learning and pattern recognition. A standard Genetic Algorithm (GA) evolves a constant-size population of elements (called *chromosomes*) by using the genetic operator of *reproduction*, *crossover* and *mutation*. Each chromosome represents a candidate solution to a given problem and it is associated with a *fitness value* that reflects how good it is, with respect to the other solutions in the population. Generally, a chromosome is encoded as a string of bits from a binary alphabet. The reproduction operator copies elements of the current population into the next generation with a probability proportionate to their fitness (this strategy is also called roulette wheel selection scheme). The crossover operator generates two new chromosomes by crossing two elements of the population selected proportionate to their fitness. The mutation operator randomly alters the bits of the strings.

In the following we give a description of the algorithm *GA-Net*, the representation adopted for partitioning the network, and the genetic operators used.

Genetic representation Our clustering algorithm uses the locus-based adjacency representation proposed in [30]. In this graph-based representation an individual of the population consists of N genes $g_1, \ldots, g_N$ and each gene can assume allele values j in the range $\{1, \ldots, N\}$. 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. Suppose to have the network showed in Figure 2a. It consists of eleven nodes numbered from 1 to 11. The network can be partitioned in the three groups visualized by different colors and shapes of the nodes. Out of the many possible genotypes, that showed in Figure 2b, corresponds to the graph division given in Figure 2c. It is worth to note that the locus-based representation naturally fits with the problem of community detection since its decoding automatically identifies the number k of connected components, i.e. of communities. The nodes participating in the same component are assigned to one cluster. Furthermore, with respect to other approaches, such as [13, 35], that adopt a chromosome of length N storing the identifier of the community which nodes belong to, it has a complexity of the search space that reduces from N^N of the cluster based representation, to $\prod_{i=1}^{N} k_i$ where k_i is the degree of node i. Since often networks are sparse, the solution space is narrower, thus the locus-based representation can sensibly improve the efficiency of the genetic approach.

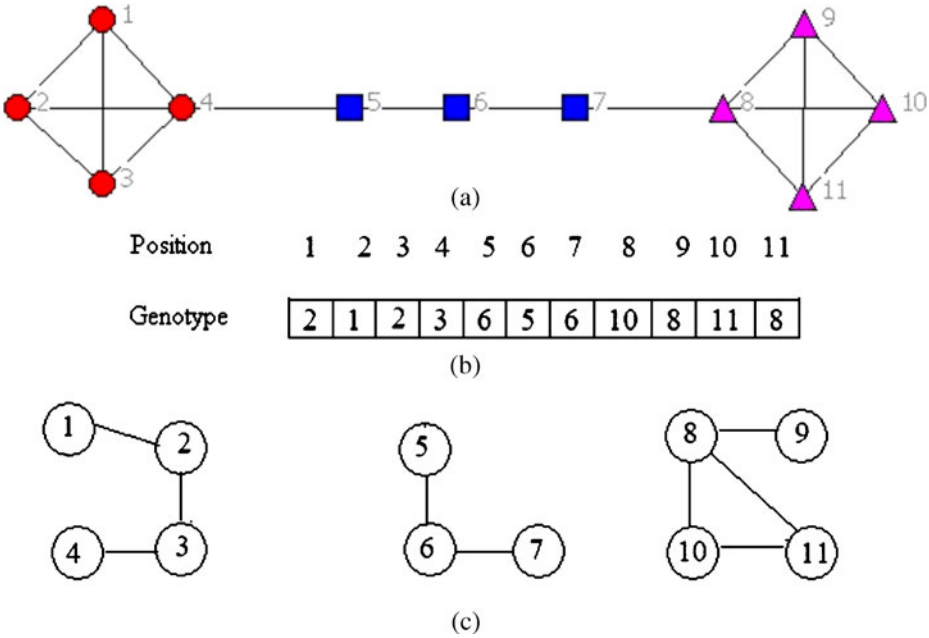

Figure 2 (**a**) A network modelled as a graph; (**b**) the locus-based representation of a genotype; (**c**) the graph-based structure of the genotype.

Objective function We are interested in identifying a partitioning that optimizes the *community score* because this guarantees highly intra-connected and sparsely inter-connected communities. The objective function is thus

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

Initialization The initialization process assigns to each each node i one of its neighbors j. This guarantees a division of the network in connected groups of nodes.

Uniform crossover and mutation The kind of crossover operator adopted is uniform crossover. Given two parents, a random binary vector is created. Uniform crossover then selects the genes where the vector is a 0 from the first parent, and the genes where the vector is a 1 from the second parent, and combines the genes to form the child. The main motivation of using uniform crossover is that it guarantees the maintenance of the effective connections of the nodes in the network in the child individual. In fact, because of the biased initialization, each individual in the population is such that if a gene i contains a value j, then the edge (i, j) exists. Since the child at each position i contains a value j coming from one of the two parents, then the edge (i, j) exists. Figure 3 shows an example of crossover. Two parents, individuals A and B, and their graph-based representations are reported. Uniform crossover of A and B gives the child C. The mutation operator, analogously to the initialization process, randomly assigns to each node i one of its neighbors.

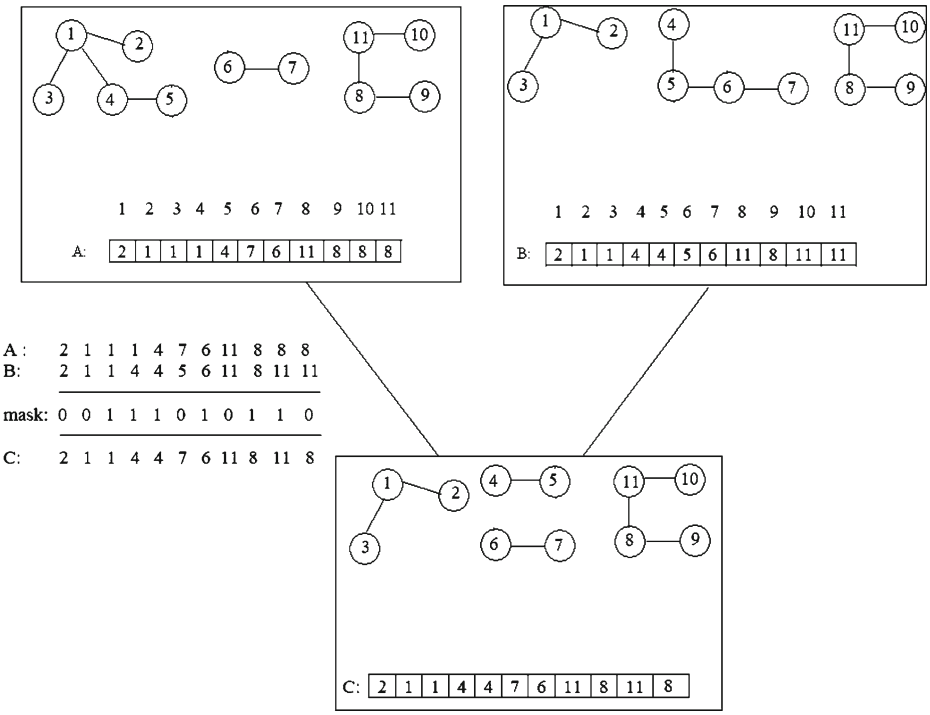

Figure 3 Uniform crossover of two individuals A and B, their genotype, their graph-based representation, and the child generated C.

The algorithm works as follows. Given a network $\mathcal{N}$ and the graph G modeling it, *GA-Net* starts with a population initialized at random but such that each node is linked with one of its neighbors. Every individual generates a graph structure in which each component is a connected subgraph of G. For a fixed number of generations the genetic algorithm computes the fitness function of each individual and applies the specialized variation operators described above to produce the new population. The individual having the best community score is returned as solution.

5 Experimental results

In this section we study the effectiveness of our approach on a synthetic data set. Then we test the results obtained by *GA-Net* on some real-worlds networks for which the partitioning in communities is known and compare it with the methods of [4] (referred as CNM), [3] (referred as BGLL), [31] (referred as PL). Furthermore the results obtained by Xiaodong et al. in [40] with their particle swarm optimization approach (referred as PSO) are also reported. Finally *GA-Net* and BGLL are compared on some real-life networks for which the network division is not known.

In all the 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 4.3 R2010a, using the Genetic Algorithms and Direct

Search Toolbox 2. In order to set parameter values, a trial and error procedure has been employed and then the parameter values giving good results for the benchmark data sets have been selected. Thus we set crossover rate to 0.8, mutation rate to 0.2, elite reproduction 10% of the population size, roulette selection function. The population size was 100, the number of generations 100. For all the data sets, the statistical significance of the results produced by *GA-Net* has been checked by performing a t-test at the 5% significance level. The p-values returned are, on average, below 0.05E-10, thus the significance level is very high since the probability that a community computed by *GA-Net* could be obtained by chance is very low.

5.1 Evaluation metrics

The quality of the partitioning obtained can be evaluated by using *validity indices*. The validity indices can be internal, i.e. they rely on the connections and separation between the groups, or external, through the use of additional data to assess the clustering outcomes. In this paper, an external measure, the *Normalized Mutual Information* (NMI), has been adopted to estimate the similarity between the true partitions and the detected ones, and an internal one, the *modularity* introduced by Girvan and Newman, to measure the density of the links inside a community with respect to the links between communities.

The *Normalized Mutual Information* is a similarity measure proved to be reliable by [5]. Given two partitions A and B of a network in communities, let C be the confusion matrix whose element C_{ij} is the number of nodes of community i of the partition A that are also in the community j of the partition B. The normalized mutual information $NMI(A, B)$ is defined as :

$$NMI(A, B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} C_{ij} log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{c_A} C_{i.} log(C_{i.}/N) + \sum_{j=1}^{c_B} C_{.j} log(C_{.j}/N)}$$

where c_A (c_B) is the number of groups in the partition A (B), $C_{i.}$ ($C_{.j}$) is the sum of the elements of C in row i (column j), and N is the number of nodes. If $A = B$, $NMI(A, B) = 1$. If A and B are completely different, $NMI(A, B) = 0$.

The *modularity* of [29] is a well known quality function to evaluate the goodness of a partition. The idea underlying the modularity is that a random graph has not a clustering structure, thus the edge density of a cluster should be higher than the expected density of a subgraph whose nodes are connected at random. This expected edge density depends on a chosen *null model*. Modularity can be written in the following way:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - P_{ij})\delta(C_i, C_j)$$

where A is the adjacency matrix of the graph, m is the number of edges of the graph, and P_{ij} is the expected number of edges between nodes i and j in the null model. δ is the Kronecker function and yields one if i and j are in the same community, zero otherwise. When it is assumed that the random graph has the same degree

distribution of the original graph, $P_{ij} = \frac{k_i k_j}{2m}$, where k_i and k_j are the degrees of nodes i and j respectively. Thus the modularity expression becomes:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

Since only the pairs of vertices belonging to the same cluster contribute to the sum, the modularity can be rewritten as

$$Q = \sum_{s=1}^{k} \left[\frac{l_s}{m} - \left(\frac{d_s}{2m} \right)^2 \right]$$

where k is the number of modules found inside a network, l_s is the total number of edges joining vertices inside the module s, and d_s is the sum of the degrees of the nodes of s. Thus the first term of each summand is the fraction of edges inside a community, and the second one is the expected value of the fraction of edges that would be in the network if edges fall at random without regard to the community structure. Values approaching 1 indicate strong community structure.

5.2 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 [19], which is an extension of the classical benchmark proposed by [11]. The network consists of 128 nodes divided into four communities of 32 nodes each. Every node has an average degree of 16 and shares a fraction γ with the other nodes of the network, and $1 - \gamma$ of links with the nodes of its community. γ is called the mixing parameter. When $\gamma < 0.5$ 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 100 different networks for values of γ ranging from 0.1 to 0.5, and computed the *Normalized Mutual Information* to measure the similarity between the true partitions and the detected ones, and the modularity to evaluate the goodness of the partitioning obtained.

Figures 4 and 5 show the normalized mutual information and the modularity, averaged over the 100 runs, for different values of the exponent r when the mixing parameter γ increases from 0.1 to 0.5. The figure points out that, when the fuzziness of modules is low (until $\gamma \leq 0.2$), independently of the r value, *GA-Net* is able to recover almost 90% of community structure and obtains good modularity values. However, when the mixing parameter increases, higher values of r help in the retrieval of the true community structure. Notice that for $\gamma = 0.5$, each node has half of the links inside its community and the other half with the rest of the network thus it is very difficult to identify the hidden groups, because the communities are mixed each other. Tables 1 and 2 reports the average values, over the 100 runs, of the normalized mutual information and modularity, respectively, along with the standard deviation. The tables point out the very low values of the standard deviation. This means that the differences among the clusterings found over the 100 runs are negligible.

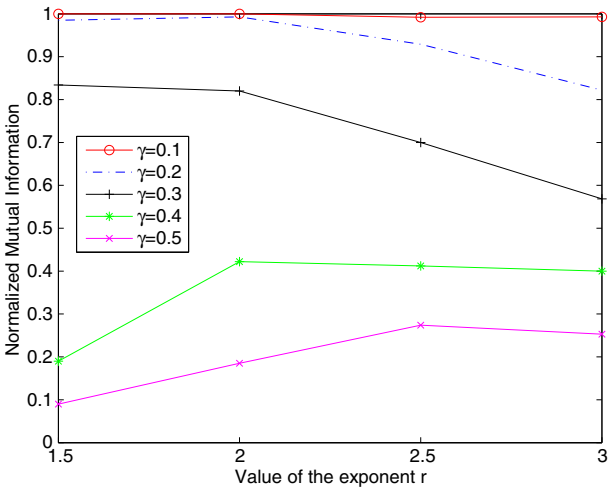

Figure 4 Normalized mutual information values obtained by *GA-Net* on the synthetic network for different values of the exponent r when the mixing parameter γ varies from 0.1 to 0.5.

5.3 Real-life networks with known community division

We now show the application of *GA-Net* on four real-world networks, well studied in the literature: *The Zackary's Karate Club network* [41], *Bottlenose Dolphins* [24], *Krebs' books on American politics* [27], and *The American College Football network* [11], and compare our results with the algorithms of [3, 4, 31]. Furthermore, we report the modularity results obtained by the PSO approach, published in [40], on three out of the 4 real-life networks. The number of real-life data sets is low because of the

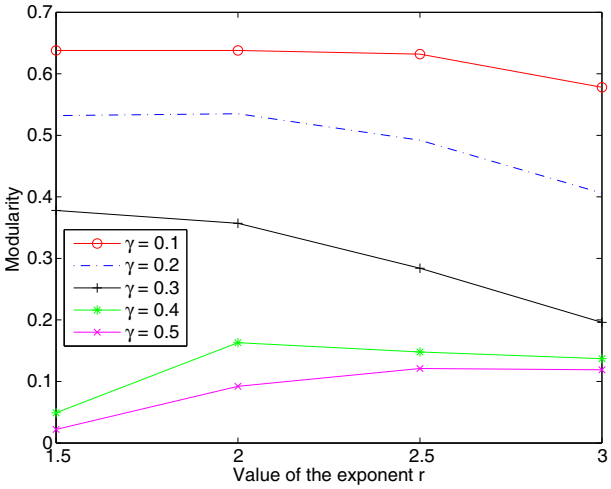

Figure 5 Modularity values obtained by *GA-Net* on the synthetic network for different values of the exponent r when the mixing parameter γ varies from 0.1 to 0.5.

Table 1 Normalized mutual information and corresponding standard deviation obtained by *GA-Net* on the synthetic data sets

γ	$r = 1.5$		$r = 2$		$r = 2.5$		$r = 3$	
	NMI	stddev	NMI	stddev	NMI	stddev	NMI	stddev
0.1	1	0	1	0	0.992	0.016	0.933	0.041
0.2	0.985	0.025	0.993	0.031	0.929	0.023	0.822	0.047
0.3	0.834	0.052	0.82	0.071	0.700	0.096	0.5686	0.122
0.4	0.1755	0.066	0.422	0.032	0.412	0.025	0.386	0.089
0.5	0.074	0.072	0.185	0.040	0.2748	0.051	0.253	0.026

unavailability in the literature of networks for which the true community division is known.

For each network we run *GA-Net* for values of r equals to 0.3, 0.5, 1, 1.5, 2, and computed the average normalized mutual information and modularity, besides the best values of NMI and modularity over 100 runs. The other contestant methods produce a unique result, that optimizing the modularity value.

Table 3 shows the good performance of *GA-Net* with respect to the others approaches. On the Karate club network *GA-Net* obtains the highest normalized mutual information of 0.826 for r=0.3 and 0.5, and a best modularity value of 0.419 for r=1,1.5, 2. As regards Bottlenose Dolphins the best NMI value of 0.888 is returned by *GA-Net* with r=0.3, though GA-MOD obtains a modularity value of 0.519. On the Krebs' book network *GA-Net* finds best values of NMI and modularity of 0.590 and 0.525, respectively, for r=0.3. Finally, on the American College Football data set, for r=2, *GA-Net* obtains a best NMI value of 0.924 and best modularity value of 0.6005 with respect to 0.926 and 0.601 of Blondel et al. The modularity values obtained by the particle swarm optimization approach on the three first networks, instead are rather poor, thus establishing the superiority of genetic algorithms. It is worth to note that the optimization of modularity does not necessarily corresponds to maximization of the normalized mutual information. In fact, as pointed out by [15], the optimal partition returned by the best modularity value may not coincide with the partition that correctly identifies the intuitive community division. These observations corroborate the belief that the input r parameter is not a limitation, but rather a means to study community structure. In the next section some suggestions on the choice of this parameter are provided.

Table 2 Modularity and corresponding standard deviation obtained by *GA-Net* on the synthetic data sets

γ	$r = 1.5$		$r = 2$		$r = 2.5$		$r = 3$	
	Mod	stddev	Mod	stddev	Mod	stddev	Mod	stddev
0.1	0.638	0.004	0.638	0.004	0.632	0.016	0.578	0.037
0.2	0.532	0.028	0.535	0.019	0.492	0.013	0.406	0.035
0.3	0.378	0.044	0.357	0.056	0.284	0.037	0.196	0.060
0.4	0.049	0.036	0.163	0.018	0.148	0.022	0.137	0.030
0.5	0.022	0.025	0.092	0.010	0.121	0.003	0.119	0.003

Table 3 Best NMI results obtained by *GA-Net* and the other algorithms for the real-life data sets

		GA-Net					CNM	BGLL	PL	PSO
		0.3	0.5	1	1.5	2				
Karate	avg NMI	0.826	0.719	0.694	0.667	0.648				
	avg MOD	0.399	0.414	0. 413	0.409	0.400				
	best NMI	0.826	0.826	0.707	0.707	0.707	0.692	0.707	0.562	
	best MOD	0.399	0.419	0.419	0.419	0.415	0.380	0.415	0.394	0.231
Dolphins	avg NMI	0.888	0.502	0.409	0.409	0.401				
	avg MOD	0.379	0.482	0.454	0.457	0.429				
	best NMI	0.888	0.593	0.462	0.454	0.467	0.573	0.450	0.675	
	best MOD	0.379	0.509	0.486	0.493	0.491	0.495	0.495	0.517	0.331
Krebs	avg NMI	0.564	0.489	0.434	0.423	0.406				
	avg MOD	0.524	0.510	0.489	0.457	0.428				
	best NMI	0.590	0.518	0.456	0.470	0.448	0.530	0.442	0.543	
	best MOD	0.525	0.516	0.499	0.484	0.477	0.502	0.515	0.515	0.412
Football	avg NMI	0.167	0.820	0.851	0.820	0.904				
	avg MOD	0.175	0.389	0.548	0.510	0.575				
	best NMI	0.491	0.879	0.881	0.883	0.924	0.762	0.926	0.879	
	best MOD	0.378	0.588	0.584	0.565	0.6005	0.577	0.601	0.602	

5.4 Study of the *r* parameter

As pointed out, the *r* parameter allows for an analysis of the community structure at different hierarchical levels, each corresponding to a different number of clusters. The choice of the value to use can be done by a user on the base of the resolution level desired. A more systematic approach could be that of considering the concept of *stability* of a partitioning of a network, as introduced in [2] and employed in [20]. A partition of a network is considered *stable* if it can be destroyed only by sensibly changing the parameter *r* for which it was obtained. Since varying *r* different community structures are found with different modularity values, the plot of the modularity value with respect to *r* can present plateaus, the length of the plateau can give a criterion to choose the better value of *r*. In order to show the feasibility of this approach, *GA-Net* has been executed on the *Zackary's Karate Club* network for values of the exponent *r* ranging from 0.1 to 2. Figure 6 shows the change in average modularity value for increasing *r*, while Figure 7 reports the number of clusters found with respect to the *r* values. Figure 6 points out a plateau for $0.5 \leq r \leq 0.9$, which correspond to the network division in 4 clusters depicted in Figure 8c. Actually this is the best division found with respect to the modularity value, but if it does not correspond to the true division of the Karate Club in two groups, displayed in Figure 8a. When r=0.3 or 0.4 *GA-Net* finds the three communities showed in Figure 8b. The smaller one, constituted by the nodes 5, 6, 7, 11, 17 is a subgroup of the community on the left. By increasing *r* above 0.9 the modularity value diminishes and a higher number of groups are produced. For example, the community on the right of Figure 8d is split in three sub-groups for r=1. Thus studying the stability of a partitioning can provide an effective criterion in the choice of the *r* parameter value to use.

5.5 Real-life networks with unknown community division

The normalized mutual information and modularity employed to compare *GA-Net* with the other approaches, though the most popular, have some limitations. In fact,

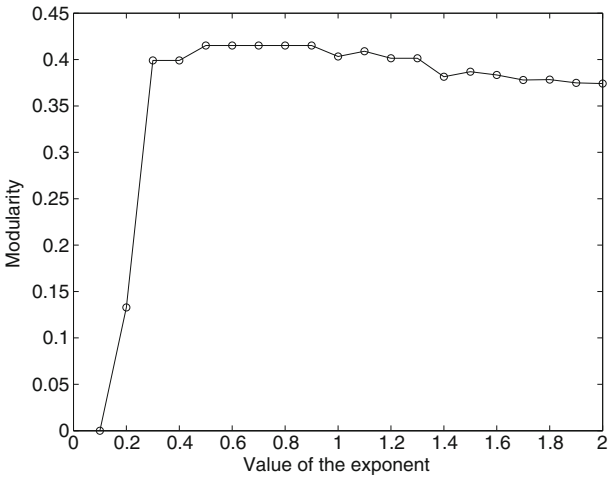

Figure 6 Change in average modularity for different values of the exponent *r*.

the NMI is applicable only with synthetic networks for which the network partition is known. On the other hand, the assessment of a method with a criterion that coincides with the fitness function it optimizes, could bias the validation phase. Recently, Leskovec et al. [21] have compared a range of community detection methods by introducing different measures. They observe that the concept of good cluster relies on two criteria. The first is the number of edges between the members of the cluster, the second is the number of edges between the members of the cluster and the rest of the network. Thus they group quality indices in two categories: multi-criterion scores, that combine both criteria, and single criterion scores, that are based on only one criterion. Modularity is a single criterion score. In the following we report some of multi-criterion indices, defined to capture the notion of cluster quality, and generalize them to evaluate network structures with different number of communities. In

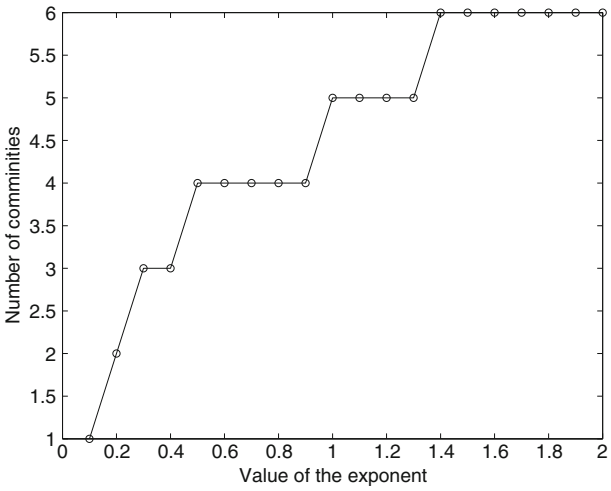

Figure 7 Change in the number of communities found for different values of the exponent *r*.

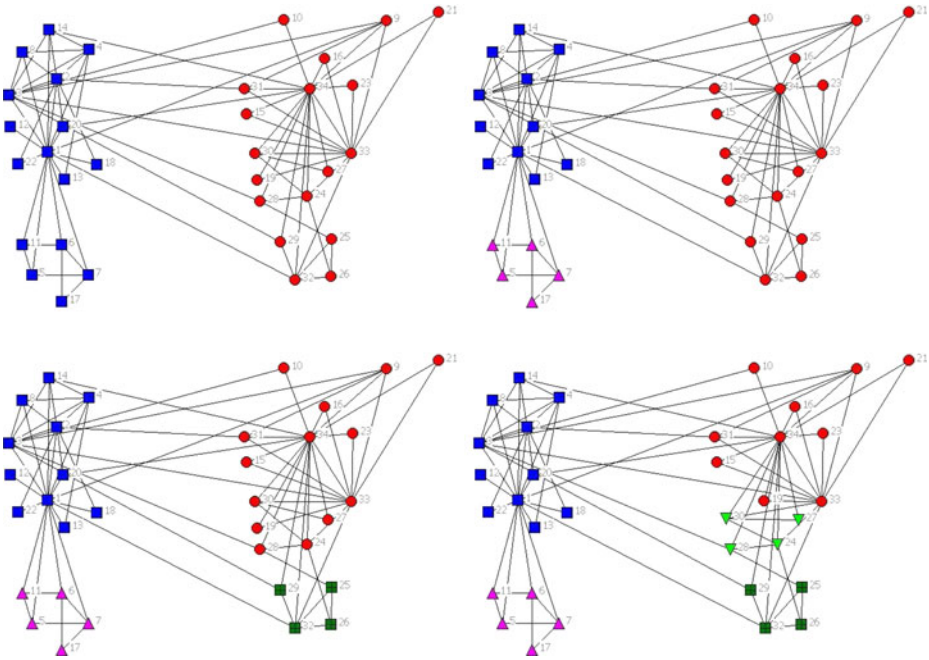

Figure 8 (**a**) True partition of the Karate Club. (**b**) Network partition with r=0.3 (**c**) Network partition with r=0.5 (**d**) Network partition with r=1.

particular we compare our approach and that of Blondel et al. [3] with respect to modularity and the multi-criterion scores. The network considered are the adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens [28], the network of Jazz musicians [12], and the Metabolic network C. Elegans [18].

Let $G = (V, E)$ the graph modeling a network with $n =\mid V \mid$ nodes and $m =\mid E \mid$ edges. Let S be a cluster of nodes having n_s nodes and m_s edges, and $c_s = \{(u, v) \mid u \in S, v \notin S\}$ the number of edges on the boundary of S. Let $\{S_1, \ldots, S_k\}$ be the partition of G in k clusters. The following metrics, reported from [21], that catch the concept of quality of a community structure are defined.

Conductance — it measures the fraction of edges pointing outside a community

$$Co = \left(\sum_{s=1}^{k} \frac{c_s}{2m_s + c_s} \right) \Big/ k$$

Expansion — it measures the number of edges per nodes that point outside the community

$$Ex = \left(\sum_{s=1}^{k} \frac{c_s}{n_s} \right) \Big/ k$$

Internal Density — it measures the internal edges density of a community

$$ID = \left(\sum_{s=1}^{k} 1 - \frac{2m_s}{n_s(n_s - 1)} \right) \Big/ k$$

Cut Ratio — it measures the fraction of all possible edges leaving the community

$$CR = \left(\sum_{s=1}^{k} \frac{c_s}{n_s(n - n_s)} \right) \Big/ k$$

The lower the values of these scores, the better the quality of the community structure obtained.

Table 4 reports the validity indices computed for *GA-Net* with different values of the *r* parameter, and BGLL. From the table it can be observed that while BGLL obtains higher values of modularity and conductance for all the networks considered, *GA-Net* performs better on Internal Density and Cut Ratio for all the networks, and on Expansion for Jazz and Adjnoun networks. These results suggest that the community score adopted by *GA-Net* finds smaller and highly dense groups of nodes having few edges towards the remaining network. These clusters substantially differs from those obtained by optimizing the modularity function, that, as already said, finds groups of nodes having a density higher than that expected in a random graph.

Table 4 Best scores obtained by *GA-Net* and BGLL algorithm for real-life data sets

	GA-Net						BGLL				
		Mod	Co	Ex	ID	CR	Mod	Co	Ex	ID	CR
Jazz	r=0.8(avg)	0.2785	0.5019	4.9646	0.3865	0.0292	0.4431	0.3186	9.0101	0.5946	0.0582
	(best)	0.2879	0.3888	2.5290	0.2502	0.0168					
nodes	r=1.2 (avg)	0.2924	0.4917	4.9760	0.3653	0.0293					
198	best	0.4093	0.3621	2.2106	0.2522	0.0150					
edges	r=1.6 (avg)	0.3623	0.5335	8.6224	0.3957	0.0510					
2742	(best)	0.4131	0.3904	7.0741	0.3138	0.0393					
	r=2 (avg)	0.3505	0.5998	9.3967	0.3860	0.05416					
	best	0.3912	0.5225	6.2522	0.2646	0.0384					
C. Elegans	r=0.8(avg)	0.3003	0.5623	3.6616	0.3021	0.008	0.4357	0.3783	3.1646	0.7888	0.0079
	(best)	0.3428	0.5362	3.1235	0.2612	0.0071					
nodes	r=1.2 (avg)	0. 2995	0.5964	4.3041	0.3244	0.0097					
453	best	0.3168	0.5735	4.03723	0.3072	0.0091					
edges	r=1.6 (avg)	0.2830	0.6041	4.2778	0.2913	0.0096					
4596	best	0.2950	0.5646	3.7961	0.2677	0.0086					
	r=2 (avg)	0.2862	0.6014	4.3107	0.2927	0.0097					
	best	0.3125	0.5487	3.9572	0.2522	0.0089					
Adjnoun	r=0.8(avg)	0.1266	0.5754	2.6017	5.1774	0.0264	0.2906	0.5420	4.0378	0.7409	0.0422
	(best)	0.1626	0.5376	2.3614	0.4182	0.02420					
nodes	r=1.2 (avg)	0.1845	0.6555	3.738	0.4801	0.0362					
112	best	0.2362	0.6322	3.0849	0.3911	0.0303					
edges	r=1.6 (avg)	0. 2129	0.6913	4.7081	0.4899	0.0447					
425	best	0.2357	0.6608	4.0937	0.4350	0.0392					
	r=2 (avg)	0.2097	0.7024	4.8492	0.4460	0.04583					
	best	0.2244	0.6915	4.4666	0.4023	0.0424					

6 Discussion

Community detection in complex network has captured a lot of interest in the last few years, and the introduction by Newman and Girvan [29] of the quantitative measure of modularity to assess the quality of a partitioning in communities has stimulated and advanced the research to uncover community structure. Recently, however, it has been proved that the optimization of modularity has a resolution limit that depends on the total size of the network and the interconnections of the modules. In [9] it is showed that modularity has an intrinsic scale such that modules below this scale, even if tightly connected, cannot be found. This limit implies the important drawback that, searching for partitioning of maximum modularity, may lead to solutions in which important structures at small scales are not discovered.

All the methods presented in the previous section, except *GA-Net*, suffer from this problem. Suppose to have the network depicted in Figure 9 composed by 4 cliques, two identical cliques of 10 nodes, and two identical cliques of 5 nodes. Neither of $BGLL$, PL, and CNM are capable of distinguishing the two small cliques. They return a partitioning in which these two small cliques are merged with a maximum modularity value of 0.5471. It is worth noticing that Blondel et al. [3] state that their approach seems to elude the limit resolution thanks to the multilevel approach of their method. However, as the above example shows, they only partially circumvent the problem. *GA-Net*, instead, perfectly discriminates the two small cliques obtaining a modularity value of 0.5356, for values of $r \geq 0.8$, and merges them for lower values of r. This means that the search for communities that maximizes the *community score* does not suffer of scale problems and has the main advantage of allowing the analysis of the network at different granularity levels. A user can thus decide at which hierarchical depth explore the structure of the network or adopt the strategy described in the previous section to obtain the most thorough information about its modular organization. Furthermore, it is worth noting that the other scores introduced in the previous section pointed out that our approach can outperform methods optimizing modularity when different metrics are adopted to evaluate the division of a network in communities.

Finally we want to point out that one of the main criticisms in using genetic algorithms, compared with traditional optimization algorithms, is the high execution time required to generate a solution. The major limitation of evolutionary algorithms is, in fact, the repeated fitness function evaluation that, for complex problems could often be prohibitive. The problem is exacerbated when large populations of individuals are used and an high number of generations are executed to obtain an

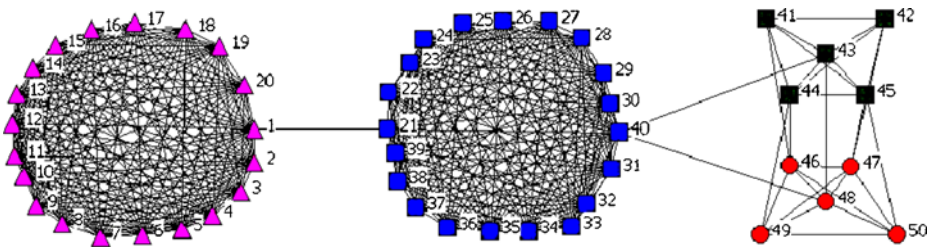

Figure 9 Network showing the resolution limit of modularity.

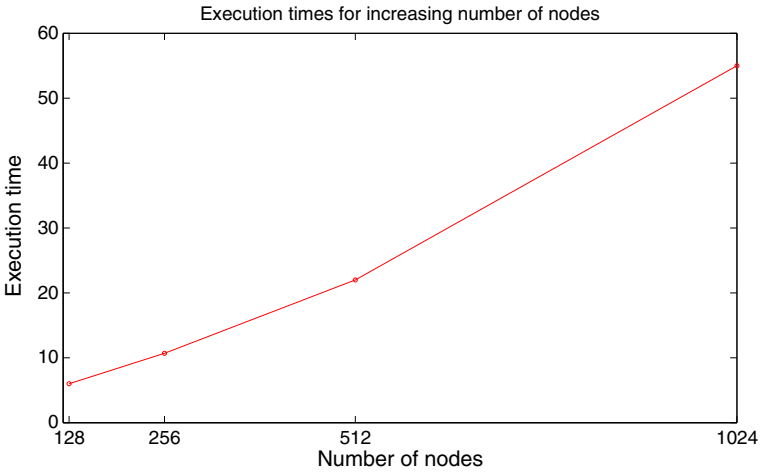

Figure 10 Execution times in seconds of *GA-Net* when the number of nodes increases from 128 to 1024.

optimal approximated solution. In our approach fitness evaluation is rather simple and can be computed in linear time, thus the main problem comes from the network size. Figure 10 shows how the execution time (in seconds) increases when the number of nodes augments from 128 to 1024. The figure indicates that the running time increases linearly with the size of the input, thus large sized networks could be used if more powerful machines are available. Moreover, Genetic Algorithms are naturally suited to be implemented on parallel architectures [36], and an implementation of *GA-Net* on a parallel machine can be easily realized.

7 Conclusions

The paper presented a genetic algorithm for detecting communities in networks. The approach introduced the concept of community score, and searches for an optimal partitioning of the network by maximizing the community score. All 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. The concept of community score, though simple, revealed very efficacious. More importantly, it enables to disclose the hierarchical organization of a network. Experiments on synthetic and real life networks showed the ability of the genetic approach to correctly detect communities with results comparable to state of the art approaches. It is worth to note that the real-life data sets presented in the paper to evaluate the method are rather small respect to the very large networks available nowadays. It is known that Genetic Algorithms can require high execution times when large populations of individuals are used. On the other hand, they are naturally suited to be implemented on parallel architectures. In order to deal with very large networks and make the approach proposed competitive with the state of the art methods that detect communities, we are planning to realize an implementation of *GA-Net* on a parallel machine.

References

1. Arenas, A., Diaz-Guilera, A.: Synchronization and modularity in complex networks. Eur. Phys. J. ST **143**, 19–25 (2007)
2. Arenas, A., Fernández, A., Gómez, S.: Analysis of the structure of complex networks at different resolution levels. arXiv:physics/0703218v2 (2008)
3. Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exp. **P10008** (2008)
4. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. **E70**, 066111 (2004)
5. Danon, L., Díaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech. **P09008** (2005)
6. Danon, L., Duch, J., Arenas, A., Díaz-Guilera, A.: Community structure identification. Large Scale Structure and Dynamics of Complex Networks: From Information Technology to Finance and Natural Science, pp. 93–113. World Scientific (2007)
7. Firat, A., Chatterjee, S., Yilmaz, M.: Genetic clustering of social networks using random walk. Comput. Stat. Data Anal. **51**(12), 6285–6294 (2007)
8. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
9. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. Proc. Natl. Acad. Sci. U.S.A. **104**(1), 36–41 (2007)
10. Fortunato, S., Castellano, C.: Community structure in graphs. arXiv:0712.2716v1 [physics.soc-ph] (2007)
11. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. U.S.A. **99**, 7821–7826 (2002)
12. , Gleiser, P.M., Danon, L.: Community structure in Jazz. Adv. Complex Systems **6**(4), 565–573 (2003)
13. Gog, A., Dumitrescu, D., Hirsbrunner, B.: Community detection in complex networks using collaborative evolutionary algorithms. In: 9th European Conference on Artificial Life (ECAL'07), pp. 886–894 (2007)
14. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing (1989)
15. Good, B.H., de Montjoye, Y., Clauset, A.: The performance of modularity maximization in practical contexts. Phys. Rev. E **81**(4), 046106 (2010)
16. Holland, J.H.: Adaptation in Natural and Artificial Systems. Univ. of Michigan Press, Ann Harbor Mich. (1975)
17. Hopcroft, J.E., Khan, O., Kulis, B., Selman, B.: Natural communities in large linked networks. In: Proc. International Conference on Knowledge Discovery and Data Mining (KDD'03), pp. 541–546 (2003)
18. Jeong, H., Tombor, B., Albert, R., Oltvai, Z., Barabási, A.-L.: The large-scale organization of metabolic networks. Nature **470**, 651–655 (2000)
19. Lancichinetti, A., Fortunato, S., Radicchi, F.: New benchmark in community detection. arXiv:0805.4770v2 [physics.soc-ph] (2008)
20. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure of complex networks. New J. Phys. **11**(033015) (2009)
21. Leskovec, J., Lang, K., Mahoney, M.W.: Empirical comparison of algorithms for network community detection. In: Proc. Int. World Wide Web Conference (WWW 2010), pp. 631–640 (2010)
22. Lipczak, M., Milios, E.: Agglomerative genetic algorithm for clustering in social networks. In: Proc. Genetic and Evolutionary Computation Conference (GECCO'09), pp. 1243–1250 (2003)
23. Lozano, S., Duch, J., Arenas, A.: Analysis of large social datasets by community detection. Eur. Phys. J. ST **143**, 257–259 (2007)
24. Lusseau, D.: The emergent properties of dolphin social network. In: Biology Letters, Proc. R. Soc. London B (suppl.) (2003)
25. Musial, K., Kazienko, P.: Social networks on the internet. World Wide Web J. doi:10.1007/s11280-011-0155-z (2012)
26. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. **E69**, 066133 (2004)
27. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. U.S.A. **103**, 8577–8582 (2006)

28. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**, 036104 (2006)
29. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. **E69**, 026113 (2004)
30. Park, Y.J., Song, M.S.: A genetic algorithm for clustering problems. In: Proc. of 3rd Annual Conference on Genetic Algorithms, pp. 2–9 (1989)
31. Pons, P., Latapy, M.: Computing communities in large networks using random walks. J. Graph Algorithms Appl. **10**(2), 191–218 (2006)
32. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. U.S.A. **101**(9), 2658–2663 (2004)
33. Schuetz, P., Caflish, A.: Multistep greedy algorithm identifies community structure in real-world and computer-generated networks. Phys. Rev. E **78**(026112) (2008)
34. Sneath, P.H.A., Sokal, R.R.: Numerical Taxonomy: The Principles and Practice of Numerical Classification. W. H. Freeman (1973)
35. Tasgin, M., Bingol, A.: Communities detection in complex networks using genetic algorithms. In: Proc. of the European Conference on Complex Systems (ECSS'06) (2006)
36. Tomassini, M.: Parallel and distributed evolutionary algorithms: a review. In: Chichester et al. (eds) Evolutionary Algorithms in Engineering and Computer Science, J. Wiley and Sons (1999)
37. Wakita, K., Tsurumi, T.: Finding community structure in mega-scale social networks. arXiv:cs/0702048v1 (2007)
38. Watt, D.J.: Small Worlds. Princeton University Press (1999)
39. Wei, F., Quian, W., Wang, C., Zhou, A.: Detecting overlapped communities in networks. World Wide Web J. **12**, 235–261 (2009)
40. Xiaodong, D., Cunrui, W., Xiangdong, L., Yanping, L.: Web community detection model using particle swarm optimization. In: Proc. of the IEEE Congress on Evolutionary Computation (CEC 2008), pp. 1074–1079 (2009)
41. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**, 452–473 (1977)