2009 21st IEEE International Conference on Tools with Artificial Intelligence

A Multi-Objective Genetic Algorithm for Community Detection in Networks

Clara Pizzuti
Institute for High Performance Computing and Networking, ICAR-CNR
Via Pietro Bucci, 41C
87036 Rende (CS), Italy
{pizzuti}@icar.cnr.it

Abstract

A multiobjective genetic algorithm to uncover community structure in complex network is proposed. The algorithm optimizes two objective functions able to identify densely connected groups of nodes having sparse interconnections. The method generates a set of network divisions at different hierarchical levels in which solutions at deeper levels, consisting of a higher number of modules, are contained in solutions having a lower number of communities. The number of modules is automatically determined by the better tradeoff values of the objective functions. Experiments on synthetic and real life networks show the capability of the method to successfully detect the network structure.

1 Introduction

Complex networks constitute an efficacious formalism to represent the relationships among 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 of nodes having dense intra-connections, and sparse inter-connections. This intuitive definition of community can be formalized in different ways, depending on the criteria adopted to decide when a group of nodes is dense. However, in general, the detection of community structure in a network can be considered as a problem of clustering and, as such, it can be formally defined as an optimization problem [7]. This implies the choice of an appropriate objective function that, when optimized, determines the clustering that best fits the concept of density. In the last few years many different approaches have been proposed to uncover community structure in networks [1, 2, 11, 14, 15, 17, 19, 20, 22] (a recent review can be found in [8]). All these approaches define the concept of criterion function and try to find the clustering that optimizes this function. In particular, Girvan and Newman [9] used the concept of *modularity* (see the following for a formal definition) as criterion to stop the division of a network in sub-networks in their divisive hierarchical clustering algorithm, one of the most known community detection methods.

Community structure detection, however, is a problem that can naturally be formulated with two different objectives. The first is the maximization of internal links, the second is the minimization of external links. There is a tradeoff between these two objectives because when the clustering is constituted by the overall network the number of external links is null, thus it is minimized, however the cluster density in not high.

In this paper we propose a multiobjective approach, named *MOGA-Net*, to discover communities in networks by employing genetic algorithms. The method optimizes two objective functions introduced in [19] and [12] that revealed both efficacious in detecting modules in complex networks. The first objective function employs the concept of *community score* to measure the quality of the division in communities of a network. The higher the community score, the more dense the clustering obtained. The second defines the concept of *fitness* of the nodes belonging to a module and iteratively finds modules having the highest sum of node fitness, in the following referred as *community fitness*. When this sum reaches its maximum value, the number of external links in minimized. Both the objective functions have a positive real-valued parameter controlling the size of the communities. The higher the value of the parameter, the smaller the size of the communities found. *MOGA-Net* exploits the benefits of these two functions and obtains the communities present in the network by selectively exploring the search space, without the need to know in advance the exact number of groups. This number is automatically

1082-3409/09 $26.00 © 2009 IEEE
DOI 10.1109/ICTAI.2009.58

determined by the optimal compromise values of the objectives. An interesting result of the multiobjective approach is that it returns not a single partitioning of the network, but a set of solutions. Each of these solutions corresponds to a different tradeoff between the two objectives and thus to diverse partitioning of the network consisting of various number of clusters. This gives a great chance to analyze several clusterings at different hierarchical levels. Experiments on synthetic and real life networks show the capability of the multiobjective 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 formulates the community detection problem as a multiobjective optimization problem. Section 4 describes the method, the genetic representation adopted and the variation operators used. In section 5, finally, the results of the method on synthetic and real life networks are presented.

2 Community definition

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 [20] 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. Let $S \subset G$ the subgraph where node i belongs to, the degree of i with respect to S can be split as $k_i(S) = k_i^{in}(S) + k_i^{out}(S)$, where $k_i^{in}(S) = \sum_{j \in S} A_{ij}$ is the number of edges connecting i to the other nodes in S, and $k_i^{out}(S) = \sum_{j \notin S} A_{ij}$ 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 has been introduced in [19]. On the other hand, in [12], a criterion that minimizes the out-degree of a community is defined by adopting the definition of weak community described above. We now first recall the definitions of these measures, and then we show how they can be exploited in a multiobjective approach to find 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 $\mu_i = \frac{1}{|S|} k_i^{in}(S)$ where $| S |$ 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

$$\mathcal{CS} = \sum_{i=1}^{k} score(S_i)$$

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 has been formulated in [19] as the problem of maximizing $\mathcal{CS}$.

In [12] the concept of *community fitness* of a module S is defined as

$$\mathcal{P}(\mathcal{S}) = \sum_{i \in S} \frac{k_i^{in}(S)}{(k_i^{in}(S) + k_i^{out}(S))^{\alpha}}$$

where $k_i^{in}(S)$ and $k_i^{out}(S)$ are the internal and external degrees of the nodes belonging to the community S, and α is a positive real-valued parameter controlling the size of the communities. When $k_i^{out}(S) = 0 \ \ \forall i$, $\mathcal{P}(\mathcal{S})$ reaches its maximum value for a fixed α. The community fitness has been used by [12] to find communities one at a time. The authors introduced the concept of *node fitness* with respect to a community S as the variation of the *community fitness* of S with and without the node i, i. e.

$$\mathcal{P}_i(S) = \mathcal{P}(S \cup \{i\}) - \mathcal{P}(S - \{i\})$$

The method starts by picking a node at random, and considering it as a community S. Then a loop over all the neighbor nodes of S not included in S is performed in order to

choose the neighbor node to be added to S. The choice is done by computing the node fitness for each node, and augmenting S with the node having the highest value of fitness. At this point the fitness of each node is recomputed, and if a node turns out to have a negative fitness value it is removed from S. The process stops when all the not yet included neighboring nodes of the nodes in S 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.

In the next section we propose a multiobjective community detection approach that optimizes both these two objectives.

3 Multiobjective community detection

Many problems in different fields are naturally formulated with multiple objectives. In particular the division of a network in subgroups of nodes having dense intraconnections and sparse interconnections has two competing objectives. The first is to maximize the links among the nodes belonging to the same module, the second is to minimize the number of connections between the communities. Thus the problem of community detection can not adequately be represented as a single objective augmented with constraints to try to implicitly satisfy the other. A more suitable approach is to formalize this problem as a multiobjective clustering problem.
A multiobjective optimization problem $(\Omega, \mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_t)$ is defined as

$$\textit{min } \mathcal{F}_i(\mathrm{S}), \quad i = 1, \ldots, t \qquad \textit{subject to } \mathcal{S} \in \Omega$$

where $\Omega = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ is the set of feasible clusterings of a network, and $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_t\}$ is a set of t single criterion functions. Each $\mathcal{F}_i : \Omega \rightarrow \mathcal{R}$ is a different objective function that determines the feasibility of the clustering obtained. Since $\mathcal{F}$ is a vector of competing objectives that must be simultaneously optimized, there is not one unique solution to the problem, but a set of solutions are found through the use of Pareto optimality theory [6]. Given two solutions $\mathcal{S}_1$ and $\mathcal{S}_2 \in \Omega$, solution $\mathcal{S}_1$ is said to *dominate* solution $\mathcal{S}_2$, denoted as $\mathcal{S}_1 \prec \mathcal{S}_2$, if and only if

$$\forall i : \mathcal{F}_i(\mathcal{S}_1) \leq \mathcal{F}_i(\mathcal{S}_2) \ \wedge \ \exists\, i \ s.t. \ \mathcal{F}_i(\mathcal{S}_1) < \mathcal{F}_i(\mathcal{S}_2)$$

A dominated solution is is not interesting because an improvement can be attained in all the objectives. Instead, a *nondominated* solution is one in which an improvement in one objective requires a degradation of another. Multiobjective optimization aims to the generation and selection of nondominated solutions, these solutions are called *Pareto-optimal*. The goal is therefore to construct the Pareto optima. More formally, the set of Pareto-optimal solutions Π is defined as

$$\Pi = \{S \in \Omega : \quad \nexists S' \in \Omega \; with \; S' \prec S\}$$

The vector $\mathcal{F}$ maps the solution space into the objective function space. When the nondominated solutions are plotted in the objective space, they are called the *Pareto front*. Thus the Pareto front represents the better compromise solutions satisfying all the objectives as best as possible. It worth to note that the Pareto-optimal solutions, as outlined in [10], always include the optimal solutions of the clustering problems with a single objective to optimize. In the next section a description of our multiobjective algorithm is given.

4 Algorithm Description

In this section we give a description of the multiobjective algorithm *MOGA-Net*, the representation adopted for partitioning the network, and the variation operators used. In the last few years many efforts have been devoted to the application of evolutionary computation to the development of multiobjective optimization algorithms. Evolutionary algorithms, in fact, proved very successful to solve multiobjective optimization problems because of the population-based nature of the approach that allows the generation of several elements of the Pareto set in a single run [5, 3].

The Multiobjective Genetic Algorithm (MOGA) we used is the *Nondominated Sorting Genetic Algorithm (NSGA-II)* proposed by Srinivas and Deb in [21] and implemented in the *Genetic Algorithm and Direct Search Toolbox* of MATLAB. NSGA-II builds a population of competing individuals and ranks them on the basis of nondominance (for a detailed description of the approach see [5]). In order to employ NSGA-II, *MOGA-Net* has been adapted with a customized population type that suitably represents a partitioning of a network and endowed with two complementary objectives. In the following the objective functions selected, the genetic encoding adopted and the modified variation operators used to work with this encoding are described.

Objective Functions: As described above, we are interested in identifying a partitioning $\{S_1, \ldots S_k\}$ that maximizes the number of connections inside each community and minimizes the number of links between the modules. The first objective is fulfilled by the *community score*. The first objective function is thus $\mathcal{CS} = \sum_{i=1}^{k} score(S_i)$. The second objective is carried out by the *community fitness* by summing up the fitnesses of all the S_i modules. The parameter α, that tunes the size of the communities, has been set to 1 because, as the authors observed, in most cases the partitioning found for this value are relevant. The second objective is thus $\sum_{i=1}^{k} \mathcal{P}(S_i)$.

Genetic representation: Our clustering algorithm uses the locus-based adjacency representation proposed in [18]

and employed by [10] for multiobjective clustering. 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 value 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. 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. As observed in [10], the decoding step can be done in linear time. 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. Figure 1 shows a network partition and the corresponding encoded genotype.

Initialization: Our initialization process takes in account the effective connections of the nodes in the network. A random individual is generated. However, if in the ith position there is an allele value j, but the edge (i, j) does not exist, the individual is $repaired$, i.e j is substituted with one of the neighbors of i. Repaired individuals are called $safe$ because they avoid uninteresting divisions containing unconnected nodes. $Safe$ individuals improve the convergence of the method because the space of the possible solutions is restricted.

Uniform Crossover: We used uniform crossover because 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 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.

Mutation: The mutation operator that randomly change 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.

Given a network $\mathcal{N}$ and the graph G modelling it, *MOGA-Net* starts with a population initialized at random and $repaired$ to produce $safe$ individuals. Every individual generates a graph structure in which each component is a connected subgraph of G. For a fixed number of generations the multiobjective genetic algorithm evaluates the objective values, assigns a rank to each individual according to Pareto dominance and sorts them. Then a new population is generated by applying the specialized variation operators described above. At the end of the procedure, *MOGA-Net* returns a set of solutions, i.e. all those contained in the Pareto front. Each of these solutions corresponds to a different tradeoff between the two objectives and thus to diverse partitioning of the network consisting of various number of clusters. This gives a great chance to analyze several clusterings at different hierarchical levels. In fact, as experimental results will show, the Pareto optimal solutions exhibit a hierarchical structure in which solutions with a higher number of communities are contained in solutions having a lower number of modules. However a criterion should be established to automatically select one solution with respect to another. To this end, in the next section, we suggest to adopt the concept of $modularity$, introduced by Girvan and Newman[17] to assess the quality of the partitioning obtained and to select, among the solutions found, that having the highest value of modularity.

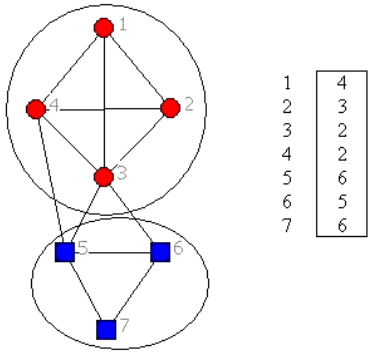

Figure 1. A network of 7 nodes partitioned in two communities $\{1, 2, 3, 4\}$ and $\{5, 6, 7\}$, and the corresponding locus-based representation.

5 Experimental Results

In this section we study the effectiveness of our approach on a synthetic data set. Then we compare the results obtained by *MOGA-Net* with the Girvan and Newman's algorithm (`http://cs.unm.edu/~aaron/research/fastmodularity.htm`), in the following referred as *GN*, on some real-worlds networks for which the partitioning in communities is known. In both cases we show that our multiobjective genetic algorithm successfully

detects the network structure and is competitive with that of Girvan and Newman. The *MOGA-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 300, the number of generations 30.

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. We adopted an external measure, the *Normalized Mutual Information*, to estimate the similarity between the true partitions and the detected ones, and an internal one, the *modularity* introduced by Girvan and Newman. The *Normalized Mutual Information* is a similarity measure proved to be reliable by Danon et al. [4]. 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 $I(A,B)$ is defined as :

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

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 $A = B$, $I(A,B) = 1$. If A and B are completely different, $I(A,B) = 0$.

The $modularity$ of Newman and Girvan [17] is a well known quality function to evaluate the goodness of a partition. Let k be the number of modules found inside a network, the $modularity$ is defined as

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

where 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. The first term of each summand of the modularity Q is the fraction of edges inside a community, 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.

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. [13], which is an extension of the classical benchmark proposed by Girvan and Newan in [9]. 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 of links with the nodes of its community, and $1-\gamma$ with the other nodes of the network. 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 10 different networks for values of γ ranging from 0.1 to 0.5, and used the *Normalized Mutual Information* to measure the similarity between the true partitions and the detected ones.

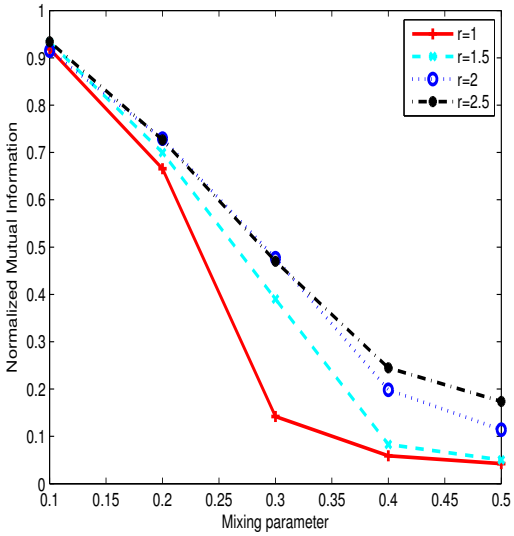

Figure 2. Normalized mutual information obtained by *MOGA-NET* on the synthetic network for different values of the exponent r.

Figure 2 shows the normalized mutual information, averaged over the 10 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, independently the value of r, *MOGA-Net* is able to recover the 90% and 70% of community structure when the fuzziness modules is low (until $\gamma \leq 0.2$). 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, being the communities mixed each other.

Real-life data set. We now show the application of *MOGA-Net* on four real-world networks, the *Zachary's Karate Club*, the *Bottlenose Dolphins*, the

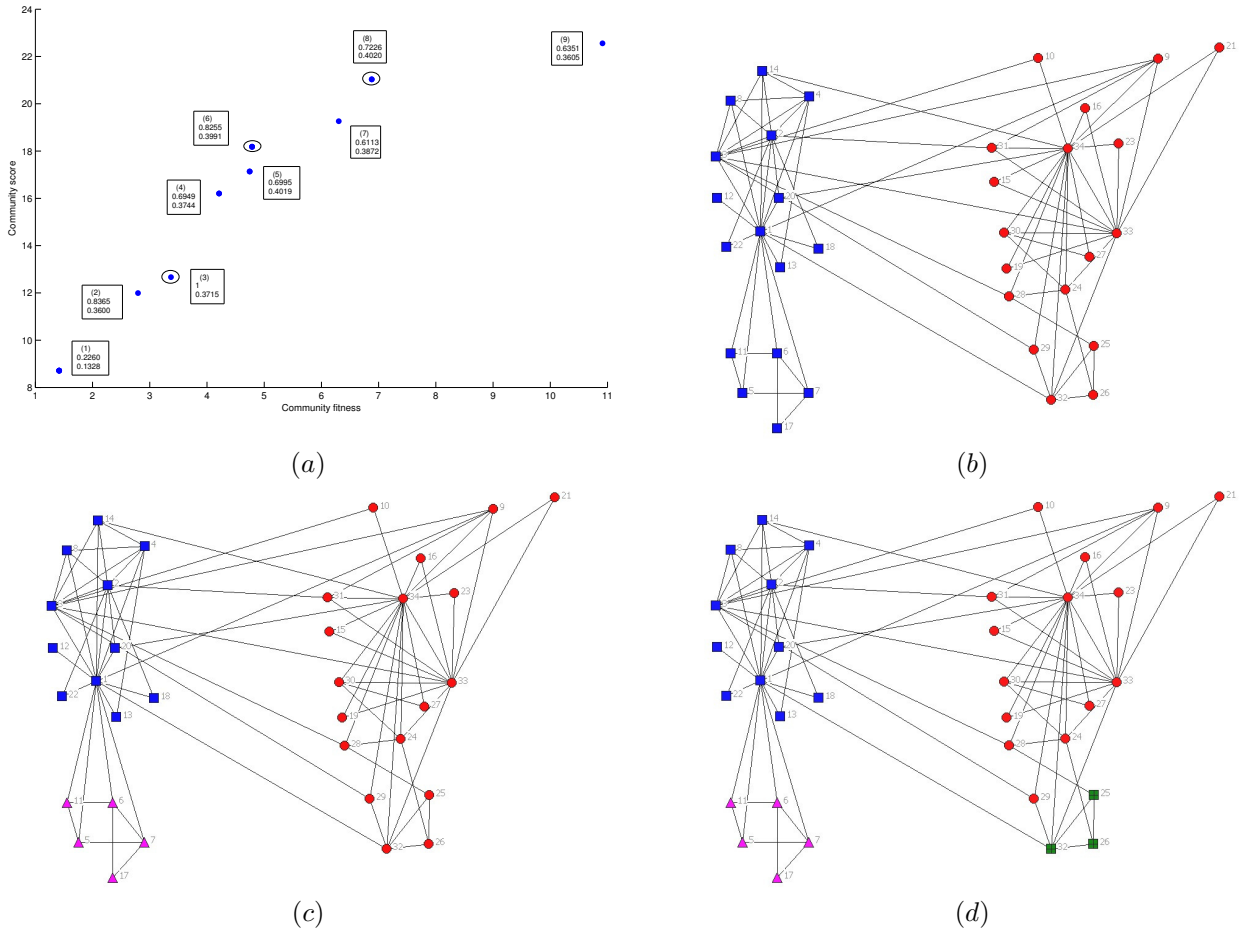

Figure 3. (a) Pareto front of one run. (b) Network corresponding to the exact solution (node number (3) on the Pareto front). (c) Network corresponding to solution (6). (d) Network corresponding to solution (8).

American College Football, and the *Krebs' books on American politics*, well studied in the literature (see `http://www-personal.umich.edu/~mejn/netdata/`), and compare our results with those obtained by Girvan and Newman. The Zackary's Karate Club network was generated by Zachary, 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 network of 62 bottlenose dolphins living in Doubtful Sound, New Zealand, was compiled by Lusseau 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 American College Football network [9] 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. The last example is the network of political books compiled by V. Krebs. The nodes represent 105 books on American politics brought from Amazon.com, and edges join pairs of books frequently purchased by the same buyer (unpublished http://www.orgnet.com/). Books were divided by Newman [16] according to their political alignment (conservative or liberal), except for a small number of books (13) having no

Table 1. Best NMI results obtained by our method and Girvan and Newman's algorithm for the real-life data sets.

	avg best NMI	std best NMI	avg Mod	std Mod	GN NMI
Zackary's Karate Club	1	0	0.371	0	0.692
Bottlenose Dolphins	1	0	0.373	0	0.573
American Coll. Football	0.795	0.016	0.497	0.027	0.762
Krebs' books	0.597	0.014	0.470	0.021	0.530

Table 2. Best modularity results obtained by our method and Girvan and Newman's algorithm for the real-life data sets.

	avg best Mod	std best Mod	avg NMI	std NMI	GN Mod
Zackary's Karate Club	0.415	0.07e-16	0.602	0.011e-15	0.380
Bottlenose Dolphins	0.505	0.009	0.506	0.046	0.495
American Coll. Football	0.515	0.016	0.775	0.023	0.577
Krebs' books	0.518	0.004	0.536	0.025	0.502

clear affiliation.

For each network, the algorithm was executed 10 times. At each run, the solutions having the best value of NMI and the best value of modularity have been selected. For each of them the corresponding modularity and NMI values, respectively, have been computed. The average values over these 10 runs are reported in tables 1 and 2. Table 1 reports the average of the best NMI (avg best NMI) and its standard deviation (std best NMI), the average modularity value (avg Mod) corresponding to the solutions having the best NMI and its standard deviation (std Mod), the Normalized Mutual Information value of the solution found by *GN* (GN NMI). Table 2 reports the average of the best modularity value (avg best Mod) and its standard deviation (std best Mod), the average NMI value (avg NMI) corresponding to the solutions having the best modularity and its standard deviation (std NMI), the modularity value of the solution found by *GN* (GN Mod).

The tables clearly shows the very good performance of *MOGA-Net* with respect to Girvan and Newman's approach. In fact, on the Zackary's Karate Club *MOGA-Net* found the exact solution for all the 10 runs with a modularity value of 0.371, while the *GN* method obtained an NMI value of 0.692 and a modularity of 0.380. The solution found by Girvan and Newman splits a cluster in two and misplaces a node of the other cluster (node 9). On the other hand table 2 shows that the average of the best modularity values obtained by *MOGA-Net* is 0.415. For all the solutions corresponding to these best modularity values, *MOGA-Net* never misplaces any node, though it splits the two clusters in smaller ones. Figure 3 displays the Pareto front in one out of the 10 runs, and the networks (3), (6), and (8) corresponding to the best value of NMI (solution (3)) and the best two values of modularity ((6) and (8)). Note that the solutions of the Pareto front have a hierarchical structure. Network (8), displayed in figure 3(d), consists of four modules obtained by the split of the two main groups in two subgroups respectively. This division has the highest value of modularity found (0.4020). Network (6), shown in figure 3(c), contains three communities, obtained by splitting the community on the left of the figure in two subgraphs. Network (3), visualized in figure 3(b), corresponds to the true partitioning of the Zackary's Karate Club in two groups. Also on the Dolphins network *MOGA-Net* found the exact solution for all the 10 runs with a modularity value of 0.373. The solution found by Girvan and Newman splits a cluster into four clusters of size 22, 2, 15, and 23, with two nodes misplaced. Table 2 shows that the average of the best modularity values obtained by *MOGA-Net* is 0.505, while that of *GN* is 0.495. On the American College Football network, *MOGA-Net* obtained an average best normalized mutual information of 0.795 with a modularity of 0.497, while the NMI of *GN* was 0.762, as reported in table 1. The best average modularity obtained by *MOGA-Net* was 0.515 while *GN* obtained the sightly higher value of 0.577. Finally, on the Krebs' network *MOGA-Net* and *GN* obtained an NMI value of 0.597 and 0.530 respectively. As regards the modularity *MOGA-Net* obtained 0.518, while Girvan and Newman had a value of 0.502.

The results obtained show capability of the multiobjective genetic algorithm to effectively deal with community identification in networks. More importantly, the non domi-

nated solutions contained in the Pareto front are meaningful and allow the analysis of the community structure at different hierarchical levels, each corresponding to a different number of clusters. The choice of one model with respect to another can be automatically done by taking the partitioning with the highest modularity value, or it can be delegated to a domain expert.

6 Conclusions

The paper presented a multiobjective genetic algorithm for detecting communities in complex networks. The approach has been shown to correctly detect communities and to be competitive with state-of-the-art methods. The algorithm has the advantage, with respect to the single objective approaches, to provide a set of solution at different hierarchical levels by giving the opportunity to analyze the network structure at different resolution levels.

References

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

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

[3] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.

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

[5] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd, Chichester, England, 2001.

[6] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition, 2005.

[7] A. Ferligoj and V. Batagelj. Direct multicriterion clustering. *Journal of Classification*, 9:43–61, 1992.

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

[9] 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.

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

[11] 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.

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

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

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

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

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

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

[18] 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.

[19] 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.

[20] 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.

[21] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolitionary Computation*, 2(3):221–248, 1994.

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