A Multiobjective Genetic Algorithm to Find Communities in Complex Networks

Clara Pizzuti

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 inter-connections. 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 that the algorithm successfully detects the network structure and it is competitive with state-of-the-art approaches.**

Index Terms—**Complex networks, multiobjective clustering, multiobjective evolutionary algorithms.**

I. 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 modeled as graphs, where nodes represent the objects and edges represent the interactions among these objects.

An important problem in the study of complex networks is the detection of community structure [25], also referred to as *clustering* [21], i.e., the division of a network into groups of nodes, called *communities* or *clusters* or *modules*, having dense intra-connections, and sparse inter-connections. This problem, as pointed out in [21], is meaningful only if the graph modeling the network is *sparse*, i.e., the number of edges is much less than the possible number of edges, otherwise it becomes similar to data clustering [31]. Clustering on graphs differs from data clustering since clusters in graphs are based on edge density, while in data clustering they are groups of points close with respect to a distance or similarity measure. The concept of community in a network, however, is not rigorously defined since its definition is influenced by the application domain of interest. Thus, the intuitive notion that the number of edges inside the same community should be much higher than the number of edges connecting to the remaining nodes of the graph, constitutes a general advice for community definition. This intuitive definition pursues two different objectives: maximizing the internal links and minimizing the external links.

Multiobjective optimization is a problem solving technique that successfully finds a set of solutions when multiple and conflicting objectives must be optimized. These solutions are obtained through the use of Pareto optimality theory [15] and constitute global optimum solutions satisfying all the objectives as best as possible. Evolutionary algorithms to solve multiobjective optimization problems revealed successful because of their population-based nature which allows the simultaneous production of multiple optima and a good approximation of the Pareto front [5].

Community detection, thus, could be formulated as a multiobjective optimization problem and the framework of Pareto optimality can provide a set of solutions corresponding to the best compromise among the objectives to optimize. In fact, there is a tradeoff between the two above-mentioned objectives because when the community structure 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 the last few years, many approaches have been proposed to employ multiobjective techniques for data clustering. Most of these proposals cluster objects in metric spaces [14], [17], [18], [28], [38], [39], [49], [51], though a method for partitioning graphs has been presented in [8] and a graph clustering algorithm of web user sessions is described in [12].

In this paper, a multiobjective approach, named multiobjective genetic algorithms for networks (*MOGA-Net*), to discover communities in networks by employing genetic algorithms is proposed. The method optimizes two objective functions introduced in [32] and [44] 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 to as community fitness. When this sum reaches its maximum value, the number of external links is 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*

Manuscript received February 17, 2010; revised June 8, 2010, October 12, 2010, and January 17, 2011; accepted March 30, 2011. Date of current version May 24, 2012.

The author is with the Institute of High Performance Computing and Networking, National Research Council of Italy, Rende, Cosenza 87036, Italy (e-mail: pizzuti@icar.cnr.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2011.2161090

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 determined by the optimal compromise values of the two 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 partitionings of the network consisting of various numbers of clusters. Experiments on synthetic and real life networks showed that the set of Pareto optimal solutions uncovers the hierarchical organization of the network, where solutions with a higher number of clusters are included in solution having a lower number of communities. This peculiarity of the multiobjective approach gives a great chance to analyze the network at different hierarchical levels and study communities with different modular levels.

This paper is organized as follows. In the next section, the concept of community is defined and the community detection problem is formalized. Section III describes the main approaches to community detection. Section IV formulates the community detection problem as a multiobjective optimization problem. Section V describes the method, the genetic representation adopted, and the variation operators used. In Section VI, the results of the method on synthetic and real life networks and a comparison with some of the state-of-the-art approaches are reported. Section VII, finally, discusses the advantages of the multiobjective approach and concludes this paper.

II. Community Definition

A network $\mathcal{N}$ can be modeled 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 [48] 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 $S \subset G$ be the subgraph where node i belongs to, the degree of i with respect to S can be split as

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

where

$$k_i^{\text{in}}(S) = \sum_{j \in S} A_{ij}$$

is the number of edges connecting i to the other nodes in S, and

$$k_i^{\text{out}}(S) = \sum_{j \notin S} A_{ij}$$

is the number of edges connecting i to the rest of the network. A subgraph S is a community in a strong sense if

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

A subgraph S is a community in a weak sense if

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

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 toward the rest of the network. In the following, we adopt the concept of weak community, thus a community is interpreted as a set of nodes having a total number of intra-connections higher than the number of inter-connections among different clusters.

III. Related Work

Many different algorithms, coming from different fields such as physics, statistics, data mining, and evolutionary computation have been proposed to detect communities in complex networks. The approaches adopted can broadly be classified into three different types: divisive hierarchical methods, agglomerative hierarchical methods [31], and optimization methods. The divisive hierarchical methods start from the complete network, detect the edges that connect different communities, and remove them. Examples of these approaches can be found in [3], [25], [35], [41], [42], and [48]. Agglomerative approaches consider each node a cluster and then merge similar communities recursively until the whole graph is obtained [4], [34], [40], [45], [47], [58]. Optimization methods define an objective function that allows the division of a graph in subgraphs, and try to maximize this objective in order to obtain the best partitioning of the network [1], [32], [53]. Among the optimization methods, several approaches have been developed by using evolutionary techniques. In particular, [18], [20], [26], [29], [34], [44], [55] applied genetic algorithms. Many other proposals employ multiobjective evolutionary algorithms to partition graphs or cluster objects in metric spaces [8], [12], [14], [17], [28], [38], [39], [49], [51].

In the following, we first review the main proposals coming from physics and data mining fields, and then a description of the multiobjective evolutionary clustering approaches is reported.

A. Community Detection in Networks

The community detection problem has been studied by several researchers, and a complete description of the state-of-the-art proposals is beyond the scope of this paper. Extensive and detailed overviews of community identification methods in complex networks can be found in [6], [21], and [23].

One of the most famous algorithms to detect communities has been presented by Newman and Girvan [25]. The method iteratively splits the network by removing edges. The edges to be removed are chosen by using the *betweenness* measure. The idea underlying the edge betweenness comes from the observation that if two communities are joined by a few inter-community edges, then all the paths from vertices in one community to vertices in the another must pass through these edges. Paths determine the betweenness score to compute for the edges. By counting all the paths passing through each edge, and removing the edge scoring the maximum value, the connections inside the network are broken. This process is repeated, thus dividing the network into smaller components until no edges remain.

The same authors in [42] proposed a divisive hierarchical method based on different betweenness measures. In this paper, Newman and Girvan point out the need of having a measure of the quality of the network division found by an algorithm. To this end, they introduce the concept of *modularity*. Informally, 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 (a formal definition of modularity is given in the next section). Values approaching 1 indicate strong community structure. Thus, the algorithm computes the modularity for each split of the network in communities, and the authors show that, when community structure is known *a priori*, high values of modularity closely correspond to the expected network division.

Newman [40] argued that since high values of modularity correspond to good network division, an approach to find the best possible partitioning of a network could be to simply optimize it. Thus, he presented an agglomerative hierarchical method that searches for optimal values of modularity. Newman observed that an exhaustive search of all the possible divisions to obtain the optimal value of modularity is unfeasible for networks constituted by more than 20 vertices, thus approximation methods are needed. He proposed a greedy approach that joins communities producing the greatest increase in modularity value. A faster method version, based on the same strategy, was described in [4] by Clauset, Newman, and Moore.

Blondel *et al.* [3] presented a method that partitions large networks based also 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 for 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 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 algorithm returns all the clusterings found at different hierarchical levels.

Pons and Latapy [45] introduced an agglomerative hierarchical algorithm, named *Walktrap*, to compute the community structure of a network. The approach is based on the concept of random walk on a graph and on the idea that random walks tend to get trapped in densely connected parts of the graph. A new definition of distance between two nodes is introduced by exploiting the properties of random walks, and this definition is generalized to compute the distance between communities. The algorithm thus 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 squared 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 Newman and Girvan is adopted.

Pujol *et al.* [47] proposed an agglomerative hierarchical method that combines spectral analysis and modularity optimization to obtain efficiency and accuracy in clustering a network. They used the same concept of random walk adopted by Pons and Latapy [45] to produce an initial partition of the network, then an agglomerative hierarchical method that iteratively joins two communities is applied. In order to merge two clusters, the group of nodes that gives the least contribution to the total modularity is selected and it is joined with the group that maximizes the increment of modularity.

Lancichinetti *et al.* [32] proposed a method to detect overlapping and hierarchical community structure based on the concept of community fitness of a module S. Let $k_i^{\text{in}}(S)$ and $k_i^{\text{out}}(S)$ be the internal and external degrees of the nodes belonging to a community S. The community fitness $\mathcal{P}(S)$ of S is then defined as follows:

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

where α, called resolution parameter, is a positive real-valued parameter controlling the size of the communities. When $k_i^{\text{out}}(S) = 0 \ \ \forall i$, $\mathcal{P}(S)$ reaches its maximum value for a fixed α. The community fitness has been used by [32] 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, that is

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

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. The authors found that the partitions obtained for the resolution parameter $\alpha = 1$ are relevant. However, they introduced a criterion to choose a partition based on the concept of stability. A partition is considered stable if it is delivered for a range of values of α. The length of this range determines the more stable partition, which is deemed the best result.

B. Multiobjective Clustering Methods

The application of multiobjective optimization to clustering data has recently obtained an increasing interest [14], [17], [28], [38], [39], [49], [51], though few proposals regard the partitioning of networks [8], [12].

A reference approach to multiobjective clustering algorithms for numerical and categorical data is that proposed by Handl and Knowles [28], and named multiobjective clustering with automatic K-determination (MOCK). The first objective of MOCK is to minimize the overall deviation of a partitioning, i.e., the summed distances between data items and the center of the cluster they have been assigned. The second objective is the minimization of the cluster connectedness, which evaluates for each cluster data point how many of its nearest neighbors have been placed in the same cluster. The algorithm adopts the locus-based adjacency representation proposed by Park and Song [43], described in the next sections and employed also by *MOGA-Net*, and uses a special initialization of the solutions based on the minimum spanning tree that reduces execution times. MOCK contains also a final step for selecting the best solution from the Pareto front approximations that automatically delivers the optimal number of clusters.

MOCK is not specialized for partitioning networks, though it can be adapted to clustering on graphs by considering the adjacency matrix of a network as a (dis)similarity matrix.

A proposal for graph partitioning that optimizes three different objectives was proposed by Datta *et al.* [8]. The objectives minimize the net loss in edge values when two connected nodes are placed in different groups, the difference in size of the groups, and the spread of clusters. The authors emphasized on the concept of zone in the graph, intended as group of adjacent nodes. Thus a chromosome is a collection of nodes, where each node is specified by its location in the graph. The algorithm is able to divide the graph in a variable number of zones, however the range of zones and of the number of nodes per zone must be fixed as input parameter.

More recently, a multiobjective evolutionary algorithm, specialized for clustering Web user sessions, has been proposed by Nildem *et al.* [12]. The clusters obtained are then used in a Web recommendation system for representing usage patterns. The sequences of Web pages visited by a user are represented as a weighted undirected graph where each sequence is a node, and the weight of an edge connecting two sequences is the computed similarity between the two nodes. Their algorithm, named *GraSC*, uses the same representation of MOCK, but the conflicting objectives to optimize are the min-max cut [13] and the silhouette index [50]. The former tries to optimize the intra-cluster similarity and to minimize the inter-sub-graph similarity, the latter computes the average silhouette index of vertices belonging to the same cluster. The silhouette index of a node i is the normalized value of the difference between the minimum average dissimilarity between node i and the nodes of the other clusters, and the average dissimilarity among i and the vertices in the same cluster.

In the next section, the community detection problem is formalized as a multiobjective optimization problem.

IV. Community Detection as a Multiobjective Optimization Problem

Many problems in different fields are naturally formulated with multiple objectives. In particular, the division of a network in subgroups of nodes having dense intra-connections 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 cannot 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 [19], [28].

A multiobjective clustering problem $(\Omega, \mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_t)$ is defined as

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

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 $\mathcal{F}_i : \Omega \to \mathcal{R}$ 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 [15]. 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

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

A dominated solution 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

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

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 is worth noting that the Pareto-optimal solutions as outlined in [28] always include the optimal solutions of the clustering problems with a single objective to optimize.

A. Objective Functions

Our aim is to partition a network in groups of vertices $\{S_1, \ldots S_k\}$ such that the density of edges within them is higher than the density of edges between the groups. To this end, we need an objective function that maximizes the number of connections inside each community, and another objective function that minimizes the number of links between the modules found.

A quality measure of a community S that maximizes the in-degree of the nodes belonging to S has been introduced in [44]. On the other hand, a criterion that minimizes the out-degree of a community is defined in [32]. Both the approaches adopt the definition of weak community described above. We now first recall the definitions of these measures, and then show how they can be exploited in a multiobjective approach to find communities. In the following, without loss of generality, the graph modeling 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

$$\mu_i = \frac{1}{\mid S \mid} k_i^{\text{in}}(S)$$

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

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

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

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

The *volume* v_S of a community S is defined as the number of edges connecting vertices inside S, i.e., the number of 1 entries in the adjacency sub-matrix of A corresponding to S

$$v_S = \sum_{i,j \in S} A_{ij}.$$

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

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

The first objective to maximize is then the community score $\mathcal{CS}$.

As described in Section III, Lancichinetti *et al.* [32] introduced the concept of community fitness of a module S as

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

The second objective is thus 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 to minimize is thus

$$\sum_{i=1}^{k} \mathcal{P}(S_i).$$

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

V. 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 to be 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], [10].

A. Genetic Representation

Our clustering algorithm uses the locus-based adjacency representation proposed in [43] and employed by [28] and [38] 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 $G = (V, E)$ modeling 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 separate components of the corresponding graph. The nodes participating to the same component are assigned to one cluster. As observed in [28], 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. Fig. 1(a) shows a network of ten nodes partitioned in two groups. The nodes of the two partitions are depicted as circles and squares, respectively. Among the possible encoded genotypes, that shown in Fig. 1(b) is decoded in the two connected components reported in Fig. 1(c). These two components correspond to the partitioning of the graph.

B. Initialization

The initialization process takes into 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 j is substituted with one of the neighbors of i. For example, in Fig. 2(a) in the positions 3 and 10 the corresponding allele values are 9 and 5, respectively. However the edges $(3, 9)$ and $(10, 5)$ are not present in the network shown in Fig. 1(a), thus 9 is substituted by 4, and 5 is substituted by 7.

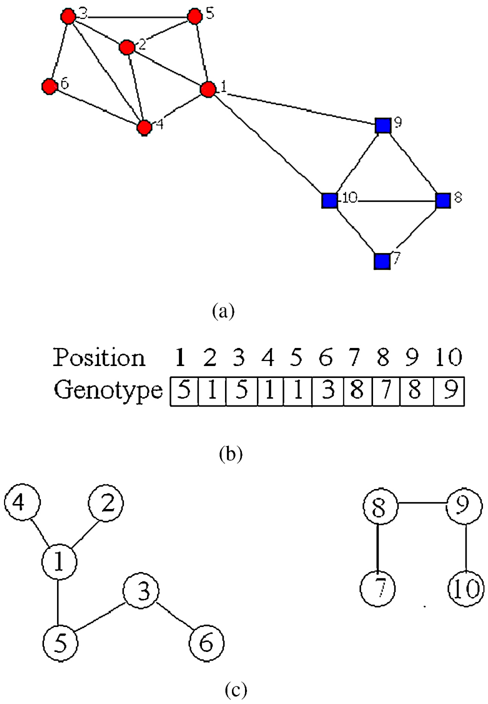

Fig. 1. (a) Network of ten nodes partitioned in two communities $\{1, 2, 3, 4, 5, 6\}$ and $\{7, 8, 9, 10\}$. (b) Locus-based representation of a genotype. (c) Graph-based structure of the genotype.

Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	9	1	1	3	8	7	8	5

(a)

Position	1	2	3	4	5	6	7	8	9	10
Genotype	5	1	4	1	1	3	8	7	8	7

(b)

Fig. 2. (a) Genotype where the couples $(3, 9)$ and $(10, 5)$ are not edges of the graph reported in Fig. 1(a). (b) Modified genotype.

C. Uniform Crossover

MOGA-Net uses a standard uniform crossover operator. First, a crossover mask of length N, i.e., the number of nodes, is randomly generated. Each value on the mask is either 0 or 1. An offspring is generated by selecting from the first parent the genes where the mask is a 0, and from the second parent the genes where the mask is a 1. 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. Fig. 3 shows an example of uniform crossover.

D. Mutation

The mutation operator that randomly changes the value j of a ith 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. For example, considering the network of Fig. 1(a), the allowed allele values of the gene in the third position are 2, 4, 5, 6. This mutation guarantees the generation of a mutated child in which each node is linked only with one of its neighbors.

Parent1 :	5	6	6	6	2	4	8	7	8	7
Parent2 :	9	4	6	3	2	4	10	10	1	9
Mask :	1	0	1	0	0	0	1	0	1	0
Offspring	9	6	6	6	2	4	10	7	1	7

Fig. 3. Example of uniform crossover.

E. Model Selection

Multiobjective clustering returns the set of Pareto-optimal 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 numbers of clusters. This gives a great chance to analyze several clusterings at different hierarchical levels. However, a criterion should be established to automatically select one solution with respect to another. To this end, we adopt the concept of *modularity*, introduced by Newman and Girvan [42]. Modularity is the most used and known function to assess the quality of a partitioning obtained by a clustering method. Let k be the number of modules found inside a network, the *modularity* is defined as

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

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. We thus select, among the solutions found on the Pareto front, that having the highest value of modularity.

Fig. 4 reports the pseudo-code of *MOGA-Net*. Given a network $\mathcal{N}$ and the graph G modeling it, *MOGA-Net* starts with a population initialized at random. 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, among the set of solutions contained in the Pareto front, that having the highest value of modularity. In the next section, experimental results will prove the ability of *MOGA-Net* in partitioning a network, and we show that 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.

Given a network $\mathcal{N}$ and the graph $\mathcal{G} = (V, E)$ modeling it, *MOGA-Net* performs the following steps:
create a population of random individuals whose length equals the number $N = \mid V \mid$ of nodes of G
while termination condition is not satisfied, execute the following sub-steps
decode each individual $I = \{g_1, \ldots, g_N\}$ of the population to generate
a partitioning $C = \{C_1, \ldots, C_k\}$ of the graph G in k connected components.
evaluate the two fitness values of the translated individuals
assign a rank to each individual and **sort** them according to nondomination rank
create a new population of offspring by applying the variation operators
combine the parents and offspring into a new pool and partition it into fronts
select points on the lower front (with lower rank), apply the variation operators on them to create the next population
end while
return the solution $C = \{C_1, \ldots C_k\}$ of the Pareto front having the maximum modularity value

Fig. 4. Pseudo-code of the *MOGA-Net* algorithm.

VI. 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 other state-of-the-art approaches on some real-world networks for which the partitioning in communities is known. In both cases, we show that our algorithm successfully detects the network structure and it is competitive with the other approaches.

The *MOGA-Net* algorithm has been written in MATLAB, using the Genetic Algorithms and Direct Search Toolbox 2. The multiobjective genetic algorithm (MOGA) we used is the nondominated sorting genetic algorithm (NSGA-II) proposed by Deb *et al.* [11] and implemented in the GA 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 [10]). It is known that setting parameter values is a challenging research problem in evolutionary algorithms [16]. Recently, Smith and Eiben [54] found that it is possible to find good parameter values for a set of problems, but general tuning that allows for good performance on a wide range of problems raises specific difficulties. As regards *MOGA-Net*, we employed a trial-and-error procedure and then selected the parameter values giving good results for the benchmark data sets. Thus, we set 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 100.

A. Evaluation Metrics

Community detection methods are supposed to identify *good* partitions [21]. In order to determine what good partition means, validity indices must be defined to assess the quality of the results obtained by an algorithm. A validity index, also called quality function, is a function that assigns a score to each partition of a network. The higher the score, the better the partition obtained. Validity indices can be internal, i.e., they rely on the connections and separation between the communities, or external, through the use of additional domain knowledge to assess the clustering outcomes. The most popular internal quality function is the *modularity* of Newman and Girvan, described in the previous section, thus it has been used as internal validity index. On the other hand, the normalized mutual information (NMI) is an external measure to estimate the similarity between the true partitions and the detected ones, that has been proved more appropriate for network partitioning by Danon *et al.* [7].

The normalized mutual information is a well-known entropy measure in information theory [37]. 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 follows:

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

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, $I(A, B) = 0$.

B. 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.* [33], which is an extension of the classical benchmark proposed by Girvan and Newan [25]. 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. We generated ten 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.

Figs. 5 and 6 show the normalized mutual information and modularity, averaged over the ten runs, for different values of the exponent r when the mixing parameter γ increases from 0.1 to 0.5. Fig. 5 points out that, independently the value of r, *MOGA-Net* is able to recover more than the 80%

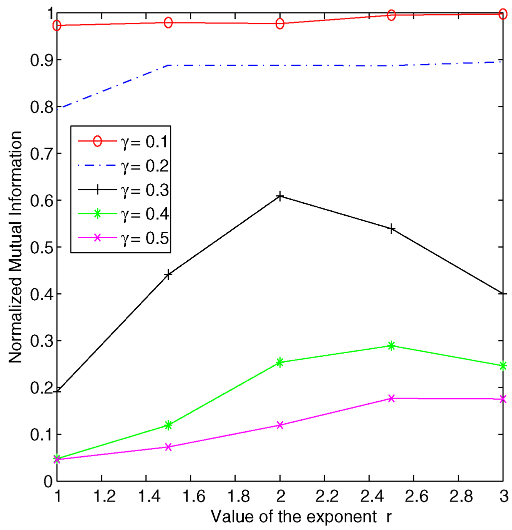

Fig. 5. Normalized mutual information obtained by MOGA-NET on the synthetic network for different values of the exponent r when the mixing parameter varies from 0.1 to 0.5.

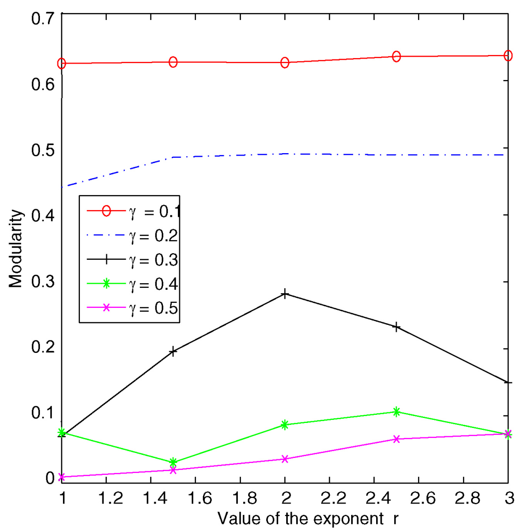

Fig. 6. Modularity obtained by MOGA-NET on the synthetic network for different values of the exponent r when the mixing parameter varies from 0.1 to 0.5.

of community structure when for each node, the number of neighbors inside its group is lower with respect to that toward other groups (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. As expected, the modularity values of the communities obtained reflect the corresponding normalized mutual information.

C. Real-Life Data Sets

We now show the application of *MOGA-Net* on four real-world networks, the Zachary's Karate Club, the Bottlenose Dolphins, the 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 three algorithms coming from network analysis, Blondel *et al.* [3] (referred to as BGLL), Clauset *et al.* [4] (referred to as CNM), and Pons and Latapy [45] (referred to as PL), and other two coming from the evolutionary computation field that apply multiobjective optimization, Handl and Knowles [28] (MOCK), and Nildem *et al.* [12] (GraSC). In the following, we first repot a brief description of each data set used.

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.

Bottlenose Dolphins is a social network of 62 bottlenose dolphins living in Doubtful Sound, New Zealand, compiled by Lusseau [36] 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 [25] 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 four inter-conference matches and seven 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.

Krebs' books on American politics is a network of political books compiled by V. Krebs. The nodes represent 105 recent books on American politics brought from Amazon.com, and edges join pairs of books frequently purchased by the same buyer [41]. Books were divided by Newman [41] according to their political alignment (conservative or liberal), except for a small number of books (13) having no clear affiliation.

All the algorithms have been executed ten times. As regards the algorithms of Clauset *et al.* [4], Blondel *et al.* [3], and Pons and Latapy [45], at each run the solution having the best modularity value is selected and the corresponding NMI value is computed. As regards MOCK [28], GraSC [12], and *MOGA-Net*, each run generates a set of solutions, those of the Pareto front. Among these optimal solutions we adopted the same selection criterion, thus the solution having the maximum modularity value is chosen and the corresponding NMI computed. The average and standard deviation values over these ten runs of both modularity and normalized mutual information are calculated and reported in Tables I and II. In *MOGA-Net* the value of the parameter r for the computation of the community score has been set to 2 because we experimented that the communities found are relevant. However, it is worth noting that the multiobjective approach implicitly explores the search space by finding solutions that could be obtained for different values of r.

The tables clearly show the very good performance of *MOGA-Net* with respect to the other approaches. In fact,

TABLE I

BEST MODULARITY RESULTS AND CORRESPONDING NORMALIZED MUTUAL INFORMATION OBTAINED BY *MOGA-Net* AND THE OTHER ALGORITHMS FOR THE REAL-LIFE NETWORKS ZAKARY'S KARATE CLUB AND BOTTLENOSE DOLPHINS

	Zakary's Karate Club		Bottlenose Dolphins	
Method	Modularity	NMI	Modularity	NMI
MOGA-Net	0.416 (0.740e-16)	0.602 (0.117e-15)	0.505 (0.0095)	0.506 (0.0468)
BGLL (Blondel *et al.* [3])	0.415	0.707	0.495	0.450
CNM (Clauset *et al.* [4])	0.380	0.692	0.495	0.573
PL (Pons and Latapy [45])	0.394	0.562	0.517	0.675
MOCK (Handl and Knowles [28])	0.326 (0.0347)	0.549 (0.1203)	0.419 (0.0271)	0.437 (0.0805)
GraSC (Nildem *et al.* [12])	0.120 (0.0292)	0.198 (0.0217)	0.073 (0.0106)	0.096 (0.0333)

TABLE II

BEST MODULARITY RESULTS AND CORRESPONDING NORMALIZED MUTUAL INFORMATION OBTAINED BY *MOGA-Net* AND THE OTHER ALGORITHMS FOR THE REAL-LIFE NETWORKS AMERICAN COLLEGE FOOTBALL AND KREB'S BOOKS

	American College Football		Krebs' Books	
Method	Modularity	NMI	Modularity	NMI
MOGA-Net	0.515 (0.0161)	0.775 (0.0234)	0.518 (0.0044)	0.537 (0.0251)
BGLL (Blondel *et al.* [3])	0.601	0.926	0.515	0.442
CNM (Clauset *et al.* [4])	0.577	0.762	0.502	0.530
PL (Pons and Latapy [45])	0.602	0.879	0.515	0.543
MOCK (Handl and Knowles [28])	0.454 (0.0608)	0.721 (0.0648)	0.437 (0.0081)	0.302 (0.1393)
GraSC (Nildem *et al.* [12])	0.285 (0.2900)	0.447 (0.3866)	0.036 (0.0391)	0.078 (0.0192)

TABLE III

BEST NMI RESULTS OBTAINED BY *MOGA-Net* ON THE REAL-LIFE DATA SETS

	MOGA-Net			
	Avg Best NMI	Std Best NMI	Avg Mod	Std Mod
Zackary's Karate Club	1	0	0.371	0
Bottlenose Dolphins	1	0	0.373	0
American College Football	0.795	0.016	0.497	0.027
Krebs' books	0.597	0.014	0.470	0.021

though the algorithm of Pons and Latapy [45] obtains a slightly better modularity value on the Dolphins network (0.517 versus 0.505) and American College Football (0.602 versus 0.536), the solutions found by *MOGA-Net* are comparable on these two data sets and better on the other two. It is worth noting that the multiobjective methods MOCK and GraSC are not able to reveal the community structure. However this is comprehensible, since the objectives they optimize are not much relevant for the problem of community detection.

Often best modularity does not correspond to the true network partition. To show that *MOGA-Net* is effective in discovering the effective network structure, over the ten runs, instead of choosing the partitioning having the best modularity value, we selected that having the best NMI value, and computed the corresponding modularity. The average values over these ten runs are reported in Table III. The table 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 table shows that on the Zackary's Karate Club and Bottlenose Dolphins *MOGA-Net* found the exact solution for all the ten runs with a modularity value of 0.371 and 0.373, respectively. On the Krebs' books network again *MOGA-Net* obtained the partitioning more similar to the true one, while on the Football network the average best NMI is lower with respect to BGLL and PL.

D. Comparing the Multiobjective Solutions

When dealing with multiobjective optimization, an important aspect to consider is the evaluation of the solutions obtained by an algorithm. In this section, the performances of *MOGA-Net* and MOCK are compared with respect to a metric specialized to assess the quality of the outcomes produced by multiobjective optimization methods. Zitzler *et al.* [59] argued that results of a multiobjective method should meet three main issues. The distance of the Pareto front generated by the algorithm from the optimal Pareto front should be minimized, the solutions should be uniformly distributed over the solution space, and the number of elements of the Pareto optimal set should be maximized. Metrics that try to measure the last issue, like error rate [57] and generational distance [56], or all the three issues, like space covered [60], assume the knowledge of the Pareto optimal front, which could not be available for real-life problems. Zitlzer and Thiele [60] proposed also a metric, named coverage metric, that evaluate whether the outcomes of an algorithm dominate the results of another algorithm. This metric is not apt to compare *MOGA-Net* and MOCK since the objectives optimized by the two methods are not the same. Schott [52] introduced a metric called *spacing* that measures the distribution of the solutions over the nondominated front. Spacing between solutions is computed as

$$S = \sqrt{\frac{1}{Q}\sum_{i=1}^{|Q|}(d_i - d)^2}$$

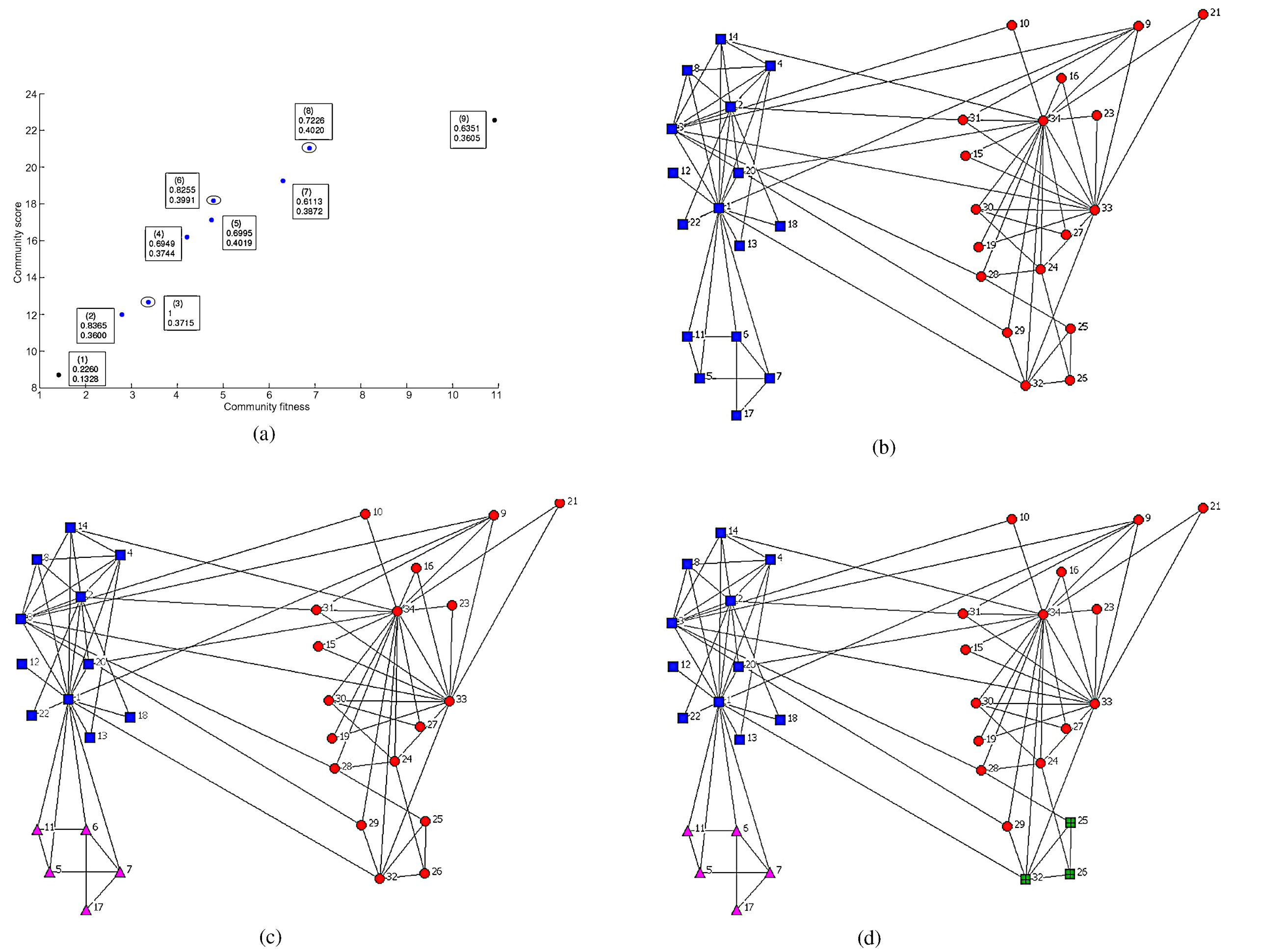

Fig. 7. (a) Pareto front of one run. (b) Network corresponding to the exact solution [node number (3) on the Pareto front]. (c) Network corresponding to (6). (d) Network corresponding to (8).

where $d_i = min_{k \in Q \text{ and } k \neq i} \sum_{m=1}^{M} \mid f_m^i - f_m^k \mid$ and f_m^i (f_m^k resp.) is the mth objective value of the ith (kth) solution in the nondominated solution set Q. d is the mean value of all the d_i. The nearer the value of S to zero, the more uniformly distributed the solutions found over the Pareto-optimal front. When the values of the objective values vary widely, a normalization of these values is necessary to avoid wrong results. To this end, the term $\mid f_m^i - f_m^k \mid$ is divided by $\mid F_m^{\max} - F_m^{\min} \mid$ where $F_m^{\max}$ and $F_m^{\min}$ are the maximum and minimum values of the mth objective.

This measure fails to measure a distribution when there is a large gap between two nondominated solutions. To overcome this problem, Bandyopadhyay *et al.* [2] defined a modified measure, called minimal spacing (in the following referred to as *MS*), that considers the distance from a solution to the nearest neighbor not already considered.

Table IV shows the average minimal spacing values and the corresponding standard deviation over the ten runs obtained by *MOGA-Net* and MOCK. The table points out that the nondominated solutions found by *MOGA-Net* are distributed more uniformly than those obtained by MOCK. In fact, the average *MS* is much lower than that computed for MOCK.

TABLE IV
MINIMAL SPACING VALUES OBTAINED BY *MOGA-Net* AND MOCK ON THE REAL-LIFE DATA SETS

	MOGA-Net		MOCK	
	Avg MS	Std MS	Avg MS	Std MS
Zackary's Karate Club	0.0201	0.0032	0.0788	0.0231
Bottlenose Dolphins	0.0096	0.0016	0.01903	0.0039
American College Football	0.0075	0.0014	0.0179	0.0043
Krebs' books	0.0128	0.0042	0.0188	0.0073

E. Hierarchical Pareto Front Solutions

As already observed, the solutions of the Pareto front have a hierarchical structure that allows the analysis of the network at different organization levels. To show this characteristics, Fig. 7(a) displays the Pareto front in one out of the ten runs for the Zackary's Karate Club, and the networks (3), (6), and (8) corresponding to the best value of NMI [solution (3)] and the best two values of modularity [(6), (8)]. Network (3), visualized in Fig. 7(b), corresponds to the true partitioning of the Zackary's Karate Club in two groups. These two main groups, actually, could be spilt into tighter sub-groups.

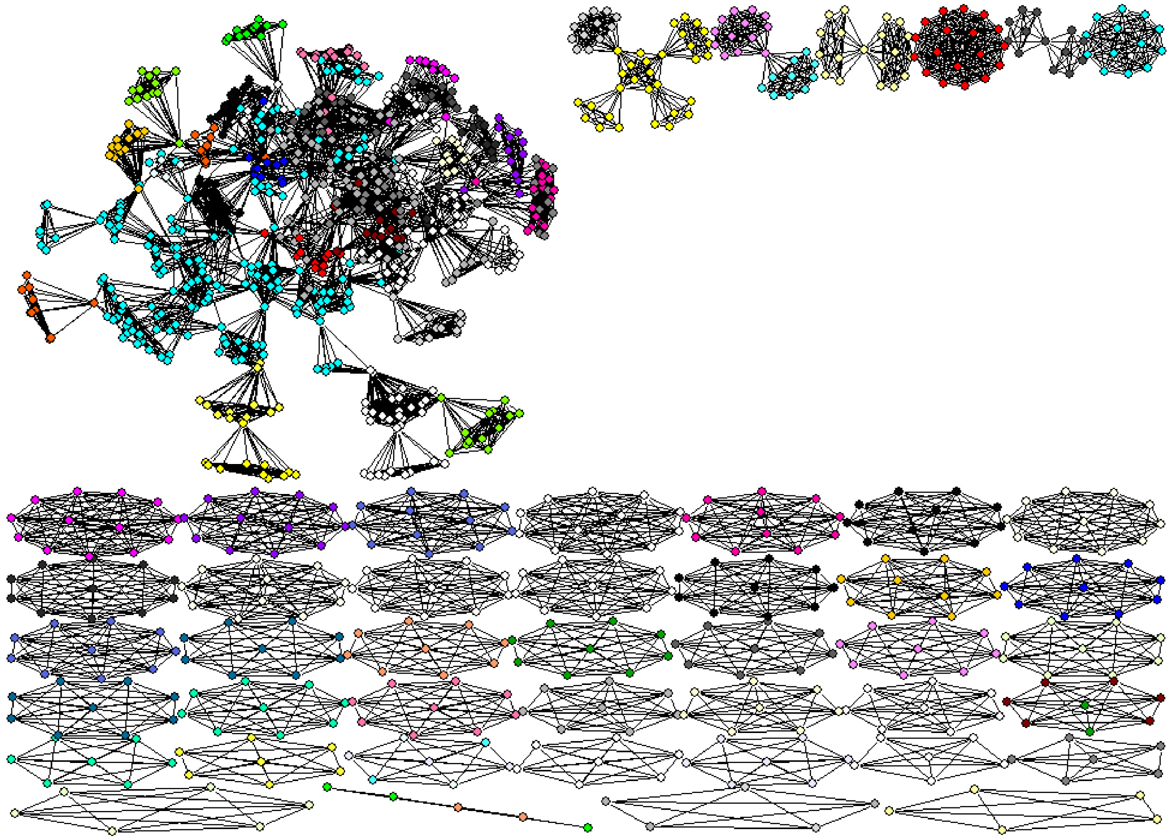

Fig. 8. 85 communities obtained by *MOGA-Net* for the Director Boards network. Different colors identify different modules.

Network (6), shown in Fig. 7(c), for example, contains three communities, obtained by the division of the community on the left of Fig. 7(b) in two subgraphs identified by blue squares (nodes 1, 2, 3, 4, 8, 12, 13, 14, 18, 20, 22) and pink triangles (nodes 5, 6, 7, 11, 17). Network (8), displayed in Fig. 7(d), consists of four modules obtained by the split of the two main groups of Fig. 7(b) in two subgroups, respectively. This division has the highest value of modularity found (0.4020). Notice the small group constituted by only three nodes (25, 26, 32).

These results show that the multiobjective approach is effective in dealing with community identification in networks and has the great advantage, with respect to single objective methods, to provide at the same time a set of optimal solutions, that contained in the Pareto front, thus allowing the exploration of the modular organization of the network.

F. Results on Large Networks

In this section, we further analyze the algorithm *MOGA-Net* by considering other three networks modeling different complex systems, and compare the results with those obtained by the method proposed by Pujol *et al.* [47], referred to as PDB after the initials of the authors, and the Newman's algorithm described in [40], referred to as Newman. The three networks are the Erdös collaboration network [46], the citation Scientometrics network [30], and the affiliation network among the Spanish top directors board [24]. In Table V, the network size, the number of communities found (*NC*), and the modularity values (*Mod*) obtained by *MOGA-Net*, *PBD* and *Newman* algorithms, respectively, are reported. The values of the last two methods are those published in [47]. The table points out that when the size of the network is large, the number of communities found by *MOGA-Net* is much higher than the number of communities found by *PBD* and *Newman*. Furthermore, the modularity values of the last two methods are higher for Erdös and Scientometrics networks, while as regards the Directors Board *MOGA-Net* it reaches almost the same value of *PBD* and it is higher than *Newman*. It is worth noting that both these two methods, as described in Section III, are agglomerative hierarchical methods that merge groups of nodes when the modularity value is optimized.

TABLE V

COMPARISON BETWEEN BEST MODULARITY VALUE AND NUMBER OF COMMUNITIES OBTAINED BY *MOGA-Net*, PBD, AND NEWMAN ALGORITHMS

		MOGA-Net		PBD		Newman	
Network	Size	Mod	NC	Mod	NC	Mod	NC
Erdös	6927	0.5502	302	0.6817	20	0.6723	57
Scientometrics	2678	0.2879	148	0.5629	10	0.5555	24
Directors Board	1130	0.8253	85	0.8273	16	0.8046	21

Recently, Fortunato and Barthélemy [22] 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. This implies that partitions obtained by the maximization of modularity could fail to obtain modules below this scale, even if tightly connected. Thus, important structures at small scales, hidden within large groups having higher modularity value, could not be discovered. This problem is further discussed by Good *et al.* [27], where it is argued that optimal modularity partitions may not coincide with the intuitive partition that correctly detects the modular structure of a network. In particular, they state that high modularity values mean that the partitioning obtained is very different from a random graph with the same degree sequence, and not necessarily that the partitioning is highly modular.

Since *MOGA-Net* does not optimize the modularity value, the partitioning it finds differs from those obtained by the other two methods. Consider Fig. 8 where the Director Boards network is depicted. Different colors of the nodes indicate the 85 different communities obtained by *MOGA-Net*.[1] It is clear from the figure that the low number of groups obtained by *PBD* (16) and *Newman* (21) indicates that the two algorithms suffer of the resolution limit problem, since the many small intuitive groups present in the network are merged together in few large communities. *MOGA-Net*, instead, though for some networks obtains partitioning of lower modularity value, has no scale problems and allows the analysis of the network at local level.

VII. Discussion and Conclusion

This paper proposed the formalization of the problem of community detection in complex networks as a multiobjective clustering problem, and presented an evolutionary multiobjective approach to uncover community structure. The method maximizes the intra-connections inside each community and minimizes inter-connections between different communities. A main characteristic of the algorithm is that it automatically affords a network partitioning without the need of knowing *a priori* the precise number of clusters. This is particularly useful in all those applications where no information about the group division is available. The approach has been tested on synthetic and real life networks, showing to be able to correctly detect communities and to be competitive with state-of-the-art methods. The multiobjective approach has the advantage, with respect to single objective approaches, to contemporarily optimize multiple criteria and to provide, not a single partitioning, but a set of solution, each corresponding to a different number of clusters, constituting the best tradeoff between the competing objectives. Experiments showed that the nondominated solutions contained in the Pareto front are meaningful and allow the analysis of the community structure at different hierarchical levels. The investigation of the network properties at various resolution levels is very important since often organizations are arranged in a hierarchical form, where small groups aggregate to produce larger communities. The choice of one model with respect to another can be done by adopting an internal criterion of quality, like that adopted by the approaches described in this paper, i.e., selecting the partitioning with the highest modularity value, or it can be delegated to a expert on the base of the application domain.

It is known that genetic algorithms can require high execution times when large populations of individuals are used. Though fitness computation of the two objectives can be done in linear time with respect to the number of network nodes, the multiobjective approach employed has a time complexity quadratic in the population size [11]. On the other hand, genetic algorithms 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 were planning to realize an implementation of *MOGA-Net* on a parallel machine.

[1]The figure has been realized by using Pajek [9]. It is worth noting that this visualization program uses at most 40 colors. When the number of clusters is above, Pajek cycles through the first forty colors again.

References

[1] A. Arenas and A. Díaz-Guilera, "Synchronization and modularity in complex networks," *Eur. Phys. J.*, vol. 143, no. 1, pp. 19–25, Apr. 2007.

[2] S. Bandyopadhyay and S. K. Pal, "Multiobjective GAs, quantitative indices, and pattern classification," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 34, no. 5, pp. 2088–2099, Oct. 2004.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefevre, "Fast unfolding of communities in large networks," *J. Statist. Mech. Theory Exp.*, vol. 2008, no. 10, p. P10008, Oct. 2008.

[4] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 6, p. 066111, 2004.

[5] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multiobjective Problems*. Berlin, Germany: Springer, 2007.

[6] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Community structure identification," *Large Scale Structure and Dynamics of Complex Networks: From Information Technology to Finance and Natural Science*. Singapore: World Scientific, 2007, pp. 93–113.

[7] L. Danon, J. Duch, A. Arenas, and A. Díaz-Guilera, "Comparing community structure identification," *J. Stat. Mech.*, vol. 2005, p. P09008, Sep. 2005.

[8] D. Datta, J. R. Figueira, C. M. Fonseca, and F. Tavares-Pereira, "Graph partitioning through a multiobjective evolutionary algorithm: A preliminary study," in *Proc. GECCO*, 2007, pp. 625–632.

[9] W. de Nooy, A. Mrvar, and V. Batagelj, *Exploratory Social Network Analysis with Pajek*. Cambridge, MA: Cambridge University Press, 2005.

[10] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[12] G. Nildem Demir, A. Sima Uyar, and S. Gündüz Öğüdücü, "Multiobjective evolutionary clustering of web user sessions: A case study in web page recommendation," *Soft Comput.*, vol. 14, no. 6, pp. 579–597, 2010.

[13] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proc. IEEE ICDM*, Dec. 2001, pp. 107–114.

[14] J. Du, E. E. Korkmaz, R. Alhajj, and K. Barker, "Novel clustering that employs genetic algorithm with new representation scheme and multiple objectives," in *Proc. 6th Int. Conf. DAWAK*, 2004, pp. 219–228.

[15] M. Ehrgott, *Multicriteria Optimization*, 2nd ed. Berlin, Germany: Springer, 2005.

[16] Á. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, Jul. 1999.

[17] K. Faceli, A. C. P. L. F. de Carvalho, and M. C. P. de Souto, "Multiobjective clustering ensemble," *Int. J. Hybrid Intell. Syst.*, vol. 4, no. 3, pp. 145–156, 2007.

[18] Z. Feng, X. Xu, N. Yuruk, and T. A. J. Schweiger, "A novel similarity-based modularity function for graph partitioning," in *Proc. 9th Int. Conf. DAWAK*, 2007, pp. 385–396.

[19] A. Ferligoj and V. Batagelj, "Direct multicriterion clustering," *J. Classification*, vol. 9, no. 1, pp. 43–61, Jan. 1992.

[20] A. Firat, S. Chatterjee, and M. Yilmaz, "Genetic clustering of social networks using random walk," *Comput. Statist. Data Anal.*, vol. 51, no. 12, pp. 6285–6294, Aug. 2007.

[21] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, pp. 75–174, 2010.

[22] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Natl. Acad. Sci. USA*, vol. 104, no. 1, pp. 36–41, Jan. 2007.

[23] S. Fortunato and C. Castellano, "Community structure in graphs," in *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, Ed. Berlin, Germany: Springer, 2009, pp. 1141–1163.

[24] *Data from the Project "Small Worlds of Corporate Networks,"* IESE Business School, Univ. Navarra, Pamplona, Spain, 2005.

[25] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[26] A. Gog, D. Dumitrescu, and B. Hirsbrunner, "Community detection in complex networks using collaborative evolutionary algorithms," in *Proc. 9th ECAL*, 2007, pp. 886–894.

[27] B. H. Good, Y.-A. de Montjoye, and A. Clauset, "The performance of modularity maximization in practical contexts," *Phys. Rev. E*, vol. 81, no. 4, p. 046106, 2010.

[28] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, Feb. 2007.

[29] D. He, Z. Wang, B. Yang, and C. Zhou, "Genetic algorithm with ensemble learning for detecting community structure in complex networks," in *Proc. 4th Int. Conf. Comput. Sci. Convergence Inform. Technol.*, Nov. 2009, pp. 702–707.

[30] *HISTCITE* [Online]. Available: http://www.garfield.library.upenn.edu/histcomp

[31] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[32] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure of complex networks," *New J. Phys.*, vol. 11, p. 033015, Mar. 2009.

[33] A. Lancichinetti, S. Fortunato, and F. Radicchi. (2008). "New benchmark in community detection," *arXiv:0805.4770v2 [physics.soc-ph]* [Online]. Available: http://arxiv.org/pdf/0805.4770v2

[34] M. Lipczak and E. Milios, "Agglomerative genetic algorithm for clustering in social networks," in *Proc. GECCO*, 2009, pp. 1243–1250.

[35] S. Lozano, J. Duch, and A. Arenas, "Analysis of large social datasets by community detection," *Eur. Phys. J. Special Top.*, vol. 143, no. 1, pp. 257–259, 2007.

[36] D. Lusseau, "The emergent properties of dolphin social network," *Biol. Lett. Proc. R. Soc. Lond. B*, vol. 270, pp. S186–S188, Nov. 2003.

[37] D. J. C. MacKay, "Information theory," *Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2002.

[38] N. Matake, T. Hiroyasu, M. Miki, and T. Senda, "Multiobjective clustering with automatic k-determination for large-scale data," in *Proc. Int. GECCO*, 2007, pp. 861–868.

[39] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 991–1005, Oct. 2009.

[40] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, no. 6, p. 066133, 2004.

[41] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[42] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, 2004.

[43] Y. J. Park and M. S. Song, "A genetic algorithm for clustering problems," in *Proc. 3rd Annu. Conf. Genet. Algorithms*, 1989, pp. 2–9.

[44] C. Pizzuti, "GA-NET: A genetic algorithm for community detection in social networks," in *Proc. 10th Int. Conf. PPSN*, 2008, pp. 1081–1090.

[45] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *J. Graph Algorithms Applicat.*, vol. 10, no. 2, pp. 191–218, 2006.

[46] *Erdös Number Project* [Online]. Available: http://www.oakland.edu/enp/thedata

[47] J. M. Pujol, J. Béjar, and J. Delgado, "Clustering algorithm for determining community structure in large networks," *Phys. Rev. E*, vol. 74, no. 1, p. 016107, Jul. 2006.

[48] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 9, pp. 2658–2663, 2004.

[49] R. Romero-Záliz, C. Rubio-Escudero, J. P. Cobb, F. Herrera, O. Cordón, and I. Zwir, "A multiobjective evolutionary conceptual clustering methodology for gene annotation within structural databases: A case of study on the gene ontology database," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 679–701, Dec. 2008.

[50] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

[51] S. Saha and S. Bandyopadhyay, "A new multiobjective clustering technique based on the concept of stability and symmetry," *Knowl. Inform. Syst.*, vol. 23, no. 1, pp. 1–27, 2010.

[52] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," M.S. thesis, Dept. Aeronautics Astronautics, Massachusetts Instit. Technol., Cambridge, 1995.

[53] P. Schuetz and A. Caflish, "Multistep greedy algorithm identifies community structure in real-world and computer-generated networks," *Phys. Rev. E*, vol. 78, no. 2, p. 026112, Aug. 2008.

[54] S. K. Smit and Á. E. Eiben, "Parameter tuning of evolutionary algorithms: Generalist versus specialist," in *Applications of Evolutionary Computation*. Berlin, Germany: Springer, 2010, pp. 542–551.

[55] M. Tasgin, A. Herdagdelen, and A. Bingol. (2007). "Communities detection in complex networks using genetic algorithms," *arXiv.org:0711.0491v1 [physics.soc-ph]* [Online]. Available: http://arxiv.org/pdf/0711.0491v1

[56] D. A. van Veldhuizen and G. B. Lamon, "Multiobjective evolutionary algorithm research: A history and analysis," Dept. Electr. Comput. Eng., Graduate School Eng., Air Force Instit. Technol., Wright-Patterson AFB, OH, Tech. Rep. TR-98-03, 1998.

[57] D. A. van Veldhuizen and G. B. Lamon, "Multiobjective evolutionary algorithm test suites," in *Proc. ACM SAC*, 1999, pp. 551–557.

[58] K. Wakita and T. Tsurumi. (2007). "Finding community structure in mega-scale social networks," *arXiv:cs/0702048v1* [Online]. Available: http://arxiv.org/pdf/cs.CY/0702048

[59] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[60] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

Clara Pizzuti received the Laurea degree in mathematics from the University of Calabria, Cosenza, Italy.

She is currently a Senior Researcher with the Institute of High Performance Computing and Networking, National Research Council of Italy, Rende, Italy. Since 1995, she has been a Contract Professor with the Department of Computer Science, University of Calabria. In the past, she worked in the research division of a software company on deductive databases, advanced logic based systems, and abduction. She has published more than 70 papers in conference proceedings and journals. Her current research interests include knowledge discovery in databases, data mining, data streams, bioinformatics, e-health, social network analysis, evolutionary computation, genetic algorithms, and genetic programming.

Ms. Pizzuti is serving as a program committee member of international conferences and as a reviewer for several international journals.