Evolutionary Computation for Community Detection in Networks: a Review

Clara Pizzuti

Abstract—In today's world, the interconnections among objects in many domains are often modeled as networks, with nodes representing the objects and edges the existing relationships among them. A key feature of complex networks is the tendency of entities to group together to form communities. The detection of communities has been receiving a great deal of interest by researchers. In fact, the knowledge of how objects organize allows a better understanding of a network, and gives a deeper insight of interesting characteristics, that could not be caught if considering the network as a whole. In the last decade, evolutionary computation techniques have given a significant contribution in this context. The aim of this review is to present the approaches based on evolutionary computation to uncover community structure. Especially, the representation schemes with the genetic operators apt for them are described, and the most popular fitness functions employed by the methods are discussed. The survey covers the most recent proposals optimizing either a single objective or multiple objectives for different types of network models, such as signed, dynamic, multidimensional.

Index Terms—Complex Networks, community detection, evolutionary computation, single objective optimization, multiobjective optimization.

1 Introduction

Network science, in recent years, has been attracting many researchers from different domains. In fact, complex networks are an effective formalism in representing the relationships among objects composing many real world systems. Networks are modeled as graphs, where nodes denote the objects of a system, and edges represent the interactions among these objects. Community structure, i.e. the division of a network into groups of nodes having dense intra-connections, and sparse inter-connections [43], is an important characteristic of networks, intensively studied in the last years. The organization in communities, in fact, takes place in both society and complex systems, such as communication and transport, biology, internet, World Wide Web [81]. The problem of uncovering community structure can be formalized as an optimization problem where an appropriate criterion function, that at best catches the intuitive concept of community, must be defined and optimized. In the past years, a lot of approaches, employing different types of heuristics and a wide variety of criteria to optimize, have been proposed. Detailed surveys describing these methods can be found in [42], [92], [41], [24], [84], [109], [79], [91], [56], [1], [7], [90].

Clara Pizzuti is with the National Research Council of Italy (CNR), Institute for High Performance Computing and Networking (ICAR), Via Pietro Bucci, 4-11C, 87036 Rende (CS), Italy, e-mail: clara.pizzuti@cnr.it.
Manuscript received ; revised , 2017.

Evolutionary Computation is a powerful search and optimization technique inspired by the process of natural evolution [37], [59], successfully applied for the solution of many difficult real world problems. Evolutionary methods are flexible methods that can be used, in principle, to solve any type of problem, provided that the problem can be formulated as an optimization task. These methods consist of population initialization, followed by variation and selection operators to improve the value of a criterion, able to escape from local minima, while exploring the search space during the optimization process.

In the last decade, we have witnessed an impressive growth of new methods based on evolutionary computation for the community detection problem. This increasing popularity is due to the capability of evolutionary computation in providing a simple, but efficacious, methodology for solving a complex problem, by requiring the definition of few basic concepts: a suitable representation for the problem, the function to optimize, and how individuals of the population evolve. Compared to classical metaheuristic methods, they present a number of advantages:

- the number of communities is automatically determined during the search process,
- domain-specific knowledge can be incorporated inside the method, such as biased initialization, or specific variation operators instead of random, allowing a more effective exploration of the state space of possible solutions,
- being population-based models, they are naturally parallel and efficient implementations can be realized to deal with large size networks.

The objective of this review is to give a comprehensive description of the state-of-the art methods proposed so far that approach the problem of community detection with computational models inspired by evolution in nature. In particular, the review will focus on methods based on *Genetic Algorithms (GAs)* [45], and evolutionary strategies in general [13], covering also other nature inspired approaches, such as particle swarm and ant colony optimization [65], [31], firefly and bat methods [111], [112] for finding communities, eventually overlapping, in different types of networks, including undirected, directed, weighted, signed, multi-dimensional, time evolving.

The paper is organized as follows. The next section introduces definitions and concepts related to the problem. Section 3 defines the problem of community detection as

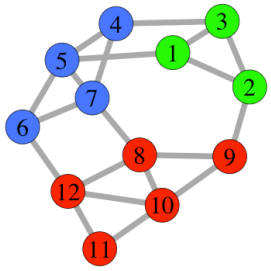


Fig. 1. An example network with 12 nodes, 20 edges, and three communities.

an optimization problem. Section 4 describes the encoding schemes, while Section 5 describes the variation operators. Section 6 illustrates the most popular objective functions, then Section 7 explains how the problem has been faced with multiobjective optimization. A comparison between single objective and mutiobjective approaches is reported in Section 8. Sections 9, 10, and 11 describe particular network models and overlapping approaches. Section 12 reports the most recent proposals of other bio-inspired methods. Section 13 concludes the paper by summarizing all the described approaches in three tables reporting for each of them, the main characteristics, and discusses future desirable developments.

2 Preliminaries

A network $\mathcal{N}$ can be modeled as a graph $G = (V, E, W)$ where V is a set of n objects, called nodes or vertices, E is a set of m links, called edges, that connect two elements of V, and $W : V \times V \rightarrow \mathcal{R}$ is a function which assigns a weight to a couple (i, j) of nodes i and j, if there exists an edge connecting i and j, and 0 if an edge between i and j does not exist [81], [4].

A graph G can be represented with the adjacency matrix A, whose elements are denoted as A_{ij}. The values of A_{ij} determine the kind of graph. Thus, an *undirected* network is such that $A_{ij} = A_{ji}$. If $A_{ij} > 1$ the network is said *weighted*, if $A_{ij} \in \{-w, 0, w\}$, the network is *signed*.

A community (also called cluster) [41] 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. A *community structure* (or clustering) is defined as a division $\mathcal{C} = \{C_1, \ldots, C_k\}$ of the network in k sub-graphs such that $V = \cup_{i=1}^{k} C_i$. When $C_i \cap C_j = \emptyset \ \forall i, j$, we have a *partitioning* of the nodes, otherwise we allow nodes to participate in more that one cluster, thus having *overlapping* communities. The degree k_i of a generic node i, is $k_i = \sum_j A_{ij}$. The degree $k_i(C)$ of a node i with respect to the community C it belongs, can be split as $k_i(C) = k_i^{in}(C) + k_i^{out}(C)$ where $k_i^{in}(C) = \sum_{j \in C} A_{ij}$ is the number of edges connecting i to the other nodes in C, and $k_i^{out}(C) = \sum_{j \notin C} A_{ij}$ is the number of edges connecting i to the rest of the network.

An example of undirected network is shown in Figure 1. This toy network will be used in the paper to illustrate genetic operators.

3 Community Detection as an Optimization Problem

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. The problem can be faced in two different ways: single objective optimization and multiple objective optimization [35].

Let $\Omega = \{\mathcal{C}_1, \ldots, \mathcal{C}_r\}$ be the set of feasible clusterings of a network. For single criterion optimization, the community detection problem can be formulated as the optimization problem $(\Omega, \mathcal{F})$ of finding a division $\mathcal{C}^*$ for which

$$\mathcal{F}(\mathcal{C}^*) = min\ \mathcal{F}(\mathcal{C}), \quad subject\ \ to\ \mathcal{C} \in \Omega \tag{1}$$

where $\mathcal{F} : \Omega \rightarrow \mathcal{R}$ is the single criterion function that determines the feasibility and quality of the clustering obtained.

For multiple objectives, the problem can be formulated as a multiobjective clustering problem $(\Omega, \mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_h)$

$$\mathcal{F}(\mathcal{C}^*) = min\ \mathcal{F}_i(\mathcal{C}), \quad i = 1, \ldots, h \qquad subject\ to\ \mathcal{C} \in \Omega \tag{2}$$

where $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_h\}$ is a set of h competing single criterion functions $\mathcal{F}_i : \Omega \rightarrow \mathcal{R}$ that must be simultaneously optimized. The aim is to find a *dominant* solution $\mathcal{C}^*$ such that, for each solution $\mathcal{C} \in \Omega$ and for each objective $\mathcal{F}_i \in \mathcal{F}$

$$\mathcal{F}_i(\mathcal{C}^*) \leq \mathcal{F}_i(\mathcal{C}) \quad i = 1, \ldots, h \tag{3}$$

Often, however, a dominant solution does not exist and the problem is how to find an *efficient solution*, i.e. one which is as good as possible respect to each criterion. Pareto optimality theory [33] allows to find these solutions. Given $\mathcal{C}_1$ and $\mathcal{C}_2 \in \Omega$, solution $\mathcal{C}_1$ is said to *dominate* solution $\mathcal{C}_2$, denoted as $\mathcal{C}_1 \prec \mathcal{C}_2$, if and only if

$$\forall i : \mathcal{F}_i(\mathcal{C}_1) \leq \mathcal{F}_i(\mathcal{C}_2) \ \wedge \ \exists\ i\ s.t.\ \mathcal{F}_i(\mathcal{C}_1) < \mathcal{F}_i(\mathcal{C}_2) \tag{4}$$

Multiobjective optimization aims to the generation and selection of *nondominated* solutions, called *Pareto-optimal*, for which an improvement in one objective requires a degradation of another. The set of Pareto-optimal solutions Π is defined as

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

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 *Pareto front*. Thus, the Pareto front represents the better compromise solutions satisfying all the objectives as best as possible.

The many proposed methods can be divided in two main categories, those optimizing only one fitness function, and those optimizing two, or more, objectives. However, independently of the number of criteria, some general principles are common for all the methods, i.e. the choice of the representation, and the type of crossover and mutation operators. In the following,

position	1	2	3	4	5	6	7	8	9	10	11	12
label	1	1	1	2	2	2	2	3	3	3	3	3

Fig. 2. Labels-based representation of the network division of the example of Figure 1.

a description of the representation schemes proposed in the literature is reported, along with the genetic operators apt for each representation and the most popular fitness functions adopted by approaches.

It is worth pointing out that these basic schemes have been introduced by single objective methods, and then exploited by the multiobjective ones. Thus, unless explicitly stated, the strategies reported in the following sections are related to single objective approaches. The multiobjective methods are then treated in detail in Section 7.

4 Encoding Schemes

The representation of a solution is a crucial part for the success of an algorithm. Several proposals exist to encode the division of a network in sub-graphs. These representations are often adapted from the encoding used to solve the classical data clustering problem with evolutionary methods [61].

4.1 Label-based representation

In this kind of encoding a genotype is an integer vector of size n, where n is the number of nodes. A position $1 \leq i \leq n$ corresponds to a node, thus, if k is the number of communities, each gene i can assume a value in the alphabet $\{1, \ldots, k\}$. This value is the label identifying the community to which node i belongs.

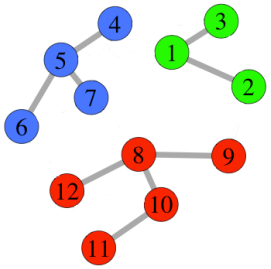
For example, consider the network of Figure 1. Figure 2 shows the label-based representation of the division of the network into the three groups $\{\{1, 2, 3\}, \{4, 5, 6, 7\}, \{8, 9, 10, 11, 12\}\}$.

Label-based representation has been widely used for data clustering [61]. Tasgin and Bingol [106] adopted it for community structure identification, making it also very popular for complex networks.

This encoding scheme, as observed in [61], is redundant because, if a genotype represents a division into k groups of nodes, there can be $k!$ different chromosomes corresponding to the same partition. The vector [3 3 3 1 1 1 1 2 2 2 2 2] represents the same network division of [1 1 1 2 2 2 2 3 3 3 3 3]. More generally, since the number of communities can be any number between 1 and n, the size of the search space can be n^n. Thus, for example, [4 4 4 10 10 10 10 6 6 6 6 6] always represents the same solution. A possible strategy to solve this problem is to apply a renumbering procedure, as suggested by Falkenauer [34], that is class labels are renumbered starting from the first available label number determined by the ordering of nodes in the chromosome. For instance, in the chromosome [4 4 4 10 10 10 10 6 6 6 6 6], class label 4 is changed to 1, class label 10 to 2, and class label 6 to 3. Though this augments the computation time, on the other hand the size of the search space is sensibly reduced. However, none

position	1	2	3	4	5	6	7	8	9	10	11	12
neighbor	2	1	1	5	7	5	5	12	8	8	10	8

(a)



(b)

Fig. 3. (a): Locus-based representation of the network division of the example of Figure 1. (b) Corresponding graph division into three connected components.

of the methods that adopt label-based representation takes into account the renumbering procedure. Gog et al. [44] proposed enriching this representation by endowing each chromosome with the value of the best ancestor individual and the value of the best individual obtained so far. Ancestors are defined as all the individuals in previous generations that contributed to the generation of the current individual. The genetic material retained through ancestors is then exploited to expand the search space, since recombination is performed only between individuals having no common ancestors.

4.2 Locus-based representation

The locus-based adjacency representation has been originally proposed in [85] for data clustering and exploited by Handl and Knowles [55] inside a multiobjective clustering method. In this graph-based representation an individual of the population consists of n genes $g_1, \ldots, g_n$ and each gene can assume allele values j in the range $\{1, \ldots, n\}$. A value j assigned to the ith gene is interpreted as a link between the nodes i and j of V. This induces a division of the network into connected components, represented through subgraphs, often trees. Consider again the network of Figure 1. The network partitioned into three groups, visualized by different colors of the nodes, can be represented, out of the many possible genotypes, by the chromosome reported in Figure 3(a), that corresponds to the graph division given in Figure 3(b).

In this representation a decoding step is necessary to identify all the components of the graph, so that nodes participating in the same component are assigned to the same cluster. This decoding step can be efficiently done in linear time by using the method reported in [23]. A main advantage of this representation is that the number k of clusters is automatically obtained by the number of components contained in an individual and determined by the decoding step.

It is worth noting that also the locus-based representation is redundant. However, the complexity of the search space

reduces from n^n of the label-based representation, to $\prod_{i=1}^{n} k_i$ where k_i is the degree of node i. Since often networks are sparse, the solution space is narrower, thus the locus-based representation can sensibly improve the efficiency of the evolutionary approach.

Locus-based representation has been first used in [86] for community detection. Since then, because of the ability of naturally mapping the community detection problem to that of automatically determining k sub-graphs (often in the form of sub-trees) of a graph, it has been adopted as a valid alternative to the label-based presentation by several authors. Chira and Gog [21], analogously to [44], extended the locus-based representation of an individual with the best potential solution, the individual's best ancestor, and added also the lowest fitness solution. This extra information is exploited to define a specialized selection function and a *collaborative* crossover operator that changes an allele value by taking into account also the ancestors. The effectiveness of this extension for both label and locus representations, however, cannot be proved since the authors experimented only on two small networks.

4.3 Medoid-based representation

The medoid-based representation uses an array of dimension k, where k, the number of communities, must be given as input parameter. The i-th element of the array contains one of the nodes composing a community. For instance, a medoid-based representation of the network of Figure 1 is the array [1 5 10], where 1 is the prototype of community $\{1, 2, 3\}$, 5 of $\{4, 5, 6, 7\}$, and 10 of $\{8, 9, 10, 11, 12\}\}$. Though this representation is more efficient in terms of space complexity, it has many drawbacks. First of all, k must be known in advance; moreover it is redundant because any element of the community can be used as medoid. Finally, it needs a decoding step to recover the communities. While for traditional clustering recovering is obtained by assigning a data object to the nearest medoid, computed with respect to a distance measure such as the Euclidean one, the concept of distance between nodes is not obvious. Firat et al. [36] discuss this problem and show that a distance measure based on *random walks* is superior to Euclidean distance. A random walk from a node i to a node j in a graph is a stochastic process modeling the path starting at i to reach j by choosing the next neighboring node at random. The authors also point out that the assignment of a node to the nearest medoid leaves parts of the search space unexplored, thus preventing the achievement of potentially good solutions. They thus propose to extend the medoid-based representation with *exception-bins*, appended to the end of the genome, containing set of nodes that are not assigned to the nearest medoid. However, how many nodes allocate to exception bins and how many bins should be used has remained an open problem, thus this proposal did not receive much attention.

4.4 Permutation-based representation

The representations described above do not allow a node to participate in more than one community. To overcome this problem, Liu et al. [77] proposed a new representation scheme that can generate overlapping communities. In this representation, in the following denoted *permutation-based*, a chromosome $\mathcal{A} = (\mathcal{A}\langle\mathbf{P}\rangle, \mathcal{A}\langle\mathbf{C}\rangle)$ is composed of two components. The first, $\mathcal{A}\langle\mathbf{P}\rangle$, is a permutation of all the nodes $\{1, 2, \ldots, n\}$

$$\mathcal{A}\langle\mathbf{P}\rangle = \{v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n}\} \tag{5}$$

and the second component, $\mathcal{A}\langle\mathbf{C}\rangle$, is a vector of n elements

$$\mathcal{A}\langle\mathbf{C}\rangle = \{c_1, c_2, \ldots, c_n\} \tag{6}$$

where c_i denotes the community of node i. In order to obtain $\mathcal{A}\langle\mathbf{C}\rangle$, the authors adopt a so called *decoder*, which actually is an incremental method that finds communities by optimizing the *community fitness* function of Lancichinetti et al. [69]. Nodes are examined in the order given by $\mathcal{A}\langle\mathbf{P}\rangle$ and added to an existing community if the fitness function augments, otherwise a singleton community is created. This implies that the same node could improve the fitness of more than one community, and thus added to many communities, giving rise to overlapped communities. When all nodes have been examined, a merging phase combines couples of communities if they have in common at least 50% of nodes. The decoding step of this representation presents two kinds of problems. The first is that at each generation, in order to obtain $\mathcal{A}\langle\mathbf{C}\rangle$, an algorithm must be executed. Thus decoding could be computationally expensive. The second problem is that many singleton communities could be generated. Though iterative merging of communities can dampen the problem, communities constituted by single nodes can still be present.

Each of the above representations has positive and negative aspects. The label-based one is the most simple but also highly redundant. The main drawback is that it generates a clustering division $C = \{C_1, \ldots, C_k\}$ such that a community C_i could contain nodes not connected to any of the nodes present in C_i. To overcome this undesirable behavior, specialized operators have been suggested [106], but the guarantee that disconnected nodes will not be present in communities cannot be assured. The main disadvantage of the locus-based representation is the need of decoding each chromosome before the fitness evaluation, thus if both the size of the population and the number of nodes are high, this could slow down an algorithm. However, decoding can be efficiently performed in $O(n\ log\ n)$ time by using a *disjoint-set* data structure, as described in [23]. The medoid-based representation needs the number of communities as input parameter. This makes it not applicable to real world networks since this information is not known in advance. The main weakness of the recently proposed permutation-based representation is the choice of the decoder to obtain the assignment of a node to a community, with a considerable increase of the computational resources. On the other hand, the decoder allows assigning a node to more than one group, thus enabling overlapping among communities.

5 Genetic Operators

5.1 Crossover

Traditional crossover operators applied to the detection of communities can present several problems, analogous to those

pointed out in [34] and discussed in [61] for data clustering. The kind of problem is related to the representation used by the method.

Medoid-based representation uses one-point crossover. As regards to the label-based representation, however, a standard one-point or two-point crossover has two main drawbacks. The first is that it could generate invalid solutions in which nodes having no connections are assigned to the same group, i.e. a cluster can contain disconnected subgroups of nodes. To mitigate this problem, Tasgin and Bingol [106] proposed *one-way* crossover, which is analogous to the *group-based* crossover described in [34], but which generates only one offspring from the two parents. One-way crossover, fixed the roles of the parents between *source* and *destination* chromosome, selects at random a node i in the source, and creates a child chromosome by transferring in the destination chromosome the community label of i to the node i, and to all the nodes having the same label of i in the source. An example of one-way crossover is shown in Figure 4(a). To better understand this kind of crossover, a graphical illustration can be seen in Figure 4(b). In this example, the node 7, whose label is 4, is chosen at random. Thus the child has the same gene values of the destination chromosome, except for positions $\{6, 7, 8\}$, since nodes 6 and 8 have the same label of node 7. The label of these three nodes is changed to 4. A modified one-way crossover, named *two-way*, which generates two offspring by exchanging the roles of source and destination of the parent chromosomes, has been proposed by Gong et al. [48].

The second problem is that the offspring does not inherit the genetic characteristics of parents, thus destroying some building blocks already obtained. This problem has been faced by He et al. [58] by introducing the definition of *multi-individual ensemble learning-based* crossover operator, that generates an offspring by using a hierarchical agglomerative clustering method. This method starts by assigning each node to a community, and iteratively merges the two communities with the maximal fitness value, provided that they contain a couple of nodes belonging to the same cluster in at least an individual, out of the M best chromosomes of the current population. Though the authors state that this kind of crossover improves the global search capability of their method, they do not discuss the computational time increase due to the execution, at each step, of the hierarchical clustering method that has to take into account the best network divisions of the current generation. Moreover, how many "promising clustering solutions" should be chosen by the current population to form the ensemble has not been argued.

Standard uniform crossover is the kind of crossover that fits well for the locus-based representation [86]. In fact, it guarantees the generation of an offspring that fully exploits the genetic information coming from the parents. A binary mask of length equal to the number of nodes is randomly created, and an offspring is generated by selecting from the first parent the genes where the mask is 0, and from the second parent the genes where the mask is 1. Since the value of a gene at position i is one of the neighbors of node i, the effect of uniform crossover is to connect a node with another

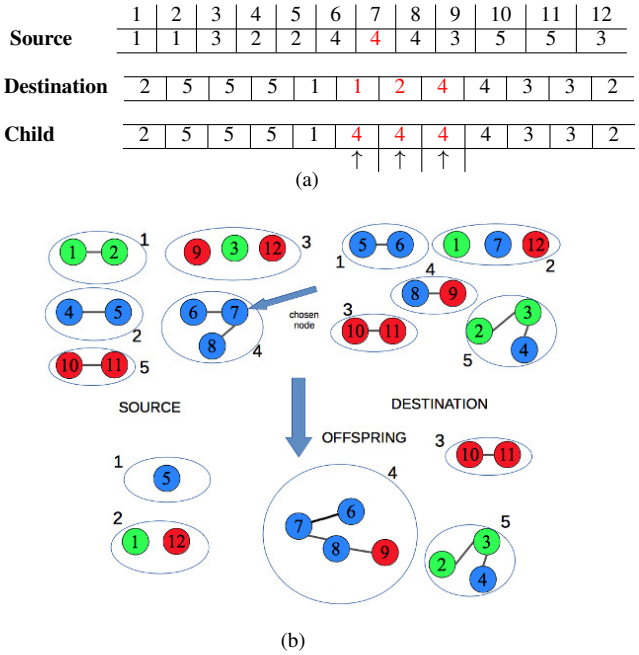


Fig. 4. (a) One-way crossover where the random position 7 is selected. The class label 4 is thus assigned to genes at positions $\{6, 7, 8\}$, which have the same label value 4 of gene 7. (b) Graphical illustration of one-way crossover.

neighboring node, thus the links of the nodes in the network are maintained in the child individual. Figure 5 shows an example of uniform crossover. Shi et al. [98] proposed the use of two-point crossover, but the advantages with respect to uniform crossover have not been investigated.

Zadeh and Kobti [116] proposed a multi-population cultural algorithm that maintains, besides the population space, a belief space having the role of knowledge repository made of selected individuals having the best fitness values. New individuals are generated by exploiting this belief space. Crossover is thus performed by choosing one parent randomly from the belief space, and the second parent from the individuals not appearing in it.

Binomial crossover is a kind of crossover employed in *Differential Evolution* approaches [26] that generates a new individual u from the target vector x and the *mutant* vector[1] v as follows:

$$u_j = \begin{cases} v_j & \text{if rand} \leq \text{CR} \quad \text{or} \quad j = j_{rand} \\ x_j & \text{otherwise} \end{cases} \tag{7}$$

where $rand$ is random number between 0 and 1, j_{rand} is an integer random number between 1 and n, and CR is a control parameter. Jia et al. [63] modified this binomial crossover operator for community detection by adding the one-way strategy of Tasgin and Bingol [105]; that is, the community label is changed not only for node j, but also for all the nodes belonging to the same community of j.

1. The concept of mutant vector [26] is explained in the next section

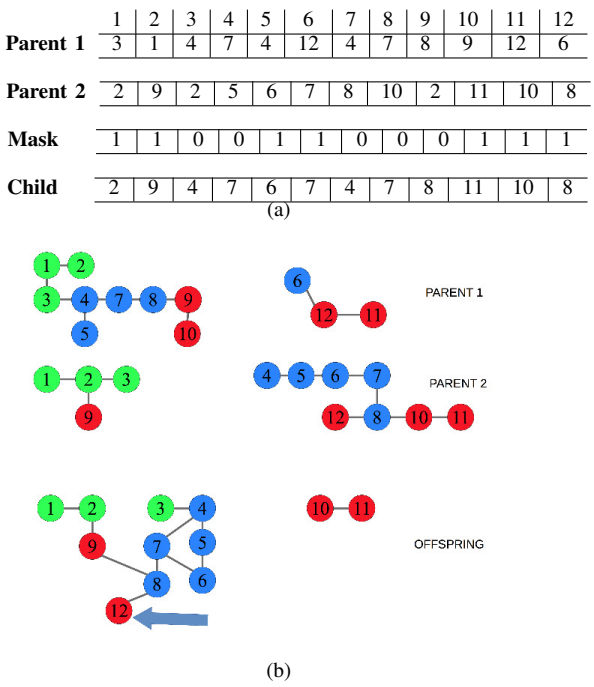


Fig. 5. (a) Uniform crossover for locus-based representation. (b) Graphical illustration.

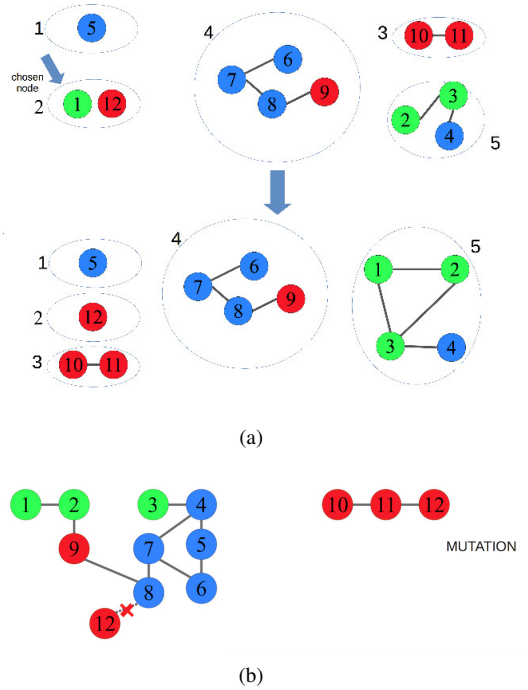


Fig. 6. (a) Mutation, for label-based representation, of the offspring of Figure 4 where node 1 is moved from cluster 2 to cluster 5. (b) Mutation, for locus-based representation, of the offspring of Figure 5 where node 12 is disconnected from node 8 and connected to node 11.

5.2 Mutation

The task of mutation is to modify gene values to allow the exploration of the search space towards regions not yet inspected. However, mutation must not be too destructive and nullify the process of finding an optimal solution. For the label based representation the simplest strategy is to randomly change the membership of a node by assigning it to one of the other existing communities [106], [70] (see Figure 6(a)). The same approach is adopted in the medoid-based representation [36]. A variant adopted by [48] is to assign a node to the cluster of one of its neighbors, while in [58] the majority label of the neighbors is adopted. The *rand/1* mutation strategy of differential evolution [26] has been employed by Jia et al. in [63]. This strategy randomly selects three individuals x_{r1}, x_{r2}, x_{r3} from the population and generates the *mutant* individual v as

$$v = x_{r1} + F \times (x_{r2} - x_{r3}) \tag{8}$$

where F is a real number between 0 and 1. Each element of the mutant vector is then checked to contain one of the allowed labels, i.e. an integer number in the interval $[1, n]$. If this constraint is violated, a function that takes back the label in the correct range values is applied.

In the locus based representation, chosen at random a node i whose allele value is j, the neighbor node j is substituted with another node among its neighbors [86]. This simple, but very effective method, causes either the split of a community or the union of two communities, thus modifying the community structure. This kind of mutation can be seen in Figure 6(b). Jin et al. [64] introduced the concept of marginal node, that is a node in a chromosome, with locus-based representation, that never appears as an allele value. Mutation is performed only on these nodes. The allele value of a marginal node i is changed to another neighbor j if the fitness of the community C to which j belongs has the best increase with respect to the other communities, when i is added to C. The same local search mutation is adopted in [76].

5.3 Population Initialization

The initial population is generally generated by assigning random values to each individual. Such a strategy, however, gives initial divisions of the network of poor quality, with true communities highly mixed. For label-based representation, Tasgin and Bingol [106] suggested choosing some nodes and assigning their community label to all their neighbors. This approach induces the generation of small initial communities that can improve the convergence of the method. Gong et al. [48] used the same strategy and suggested applying it to 20% of individuals. He et al. [58] proposed a Markov random walk method based on the probability that an agent can reach a node j from a node i in a number of steps.

In the locus-based representation, assigning to a gene i one of its neighbors is a simple approach that guarantees an initial division of the network in connected groups of nodes [86]. Liu et al. [76], analogously to [58], adopted a Markov random walk strategy.

5.4 Local search operators

Genetic operators often can produce solutions that assign nodes to the wrong community. In order to improve the

quality of the community division, a number of heuristics have been proposed. Tasgin and Bingol [106] proposed a *clean-up* process at the end of each generation that chooses a number of nodes and computes the *community variance* for such nodes. The community variance of a node i is defined as

$$CV(i) = \frac{\sum_{(i,j)\in E} f_{(i,j)}}{k_i} \tag{9}$$

where k_i is the degree of node i, and $f_{(i,j)}$ is 0 if i and j belong to the same community, 1 otherwise. Community variance is thus the ratio between the number of different communities among i and its neighbors $\{i_1, \ldots, i_{k_i}\}$, and the number of its neighbors. If this value is above a fixed threshold, then i and all its neighboring nodes are assigned to the community containing the highest number of nodes, among $\{i, i_1, \ldots, i_{k_i}\}$. Otherwise, no action is performed. The authors argue that community variance induces connected nodes to belong to the same group; however, how many nodes should undergo this check, how to select them, and which threshold value should be used have not been discussed. A different strategy is proposed by Li et al. [70] consisting in making n_l copies of a chromosome, then, for each individual, a row j is chosen at random from the adjacency matrix, and the community label of j is assigned to all its neighbors. The chromosome is then substituted by the best, in terms of modularity value, among the n_l copies. This process is repeated for each individual in the population. Also in this case, which is the best value to use for n_l is an open problem.

Gong et al. [48] perform a local search at the end of each generation, after crossover and mutation, only on the individual with the best fitness value. Chosen a node i belonging to a community C_r of the clustering $C = \{C_1, \ldots, C_k\}$, determined by such an individual, it is deleted from C_r and assigned to another cluster $C_s \in C$. The new partition with the modified communities is called a neighbor of C. The local search procedure finds all the neighbor partitions of the best individual and, if one of them has a fitness value higher than that of C, it substitutes C with this new one. This approach, as the authors also observe, is sensitive to the starting point and requires more computational effort. However, the authors reported better results when applying this strategy, though they do not say how much the computational demand increased.

Shang et al. [95] observed that a local search based on hill-climbing can prevent exploration of parts of the search space and give poor local optimal solutions. Thus, they proposed the simulated annealing method [94] and showed that it can improve the capability of the genetic algorithm to find high quality solutions.

6 Fitness Functions

The choice of the fitness function is another critical step for obtaining good solutions. In the context of community detection the most popular function is *modularity*, originally introduced by Newman and Girvan in [43], [83] to evaluate clustering results, and then used as criterion to optimize in [82]. More formally, modularity is defined as follows:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j) \tag{10}$$

where A is the adjacency matrix of the graph, m is the number of edges, k_i and k_j are the degrees of nodes i and j respectively, and δ is the Kronecker function which yields one if i and j are in the same community, zero otherwise.

Let C_1 and C_2 be two disjoint subsets of the vertex set V, $\overline{C_1} = V - C_1$, $L(C_1, C_2) = \sum_{i\in C_1, j\in C_2} A_{ij}$, $L(C_1, C_1) = \sum_{i\in C_1, j\in C_1} A_{ij}$. Since only the pairs of vertices belonging to the same cluster contribute to the sum, modularity can be rewritten as

$$Q = \sum_{i=1}^{k} \frac{L(C_i, C_i)}{2m} - \left(\frac{L(C_i, V)}{2m} \right)^2 \tag{11}$$

where k is the number of communities. The first term of each summand is the fraction of edges inside a community, while the second one is the expected value of the fraction of edges that would be in the community if the network where a random one with the same expected vertex degree. Values higher than 0.3 indicate good community structure.

Extensions to modularity to deal with weighted and directed networks have been proposed by Arenas et al. [11]. Let W be the weighed adjacency matrix of a graph, then:

$$Q = \frac{1}{2w} \sum_{ij} \left(W_{ij} - \frac{w_i^{out} w_j^{in}}{2w} \right) \delta(C_i, C_j) \tag{12}$$

where $w_i^{out} = \sum_j W_{ij}$, $w_j^{in} = \sum_i W_{ij}$, and $2w = \sum_i \sum_j W_{ij}$.

Shen et al. [96] proposed an extension to modularity for overlapping communities that takes into account the number of communities a node belongs to. The *extended modularity EQ* is defined as:

$$EQ = \frac{1}{2m} \sum_{i} \sum_{v\in C_i, w\in C_i} \frac{1}{O_v O_w} [A_{vw} - \frac{k_v k_w}{2m}] \tag{13}$$

where O_v is the number of communities to which v participates.

Fortunato and Barthélemy [40] pointed out that the optimization of modularity has a resolution limit that depends on the total size of the network and the interconnections of the modules. Moreover, the formula does not take into account the size of communities. This implies that partitions obtained by the maximization of modularity could not discover small groups, hidden within large communities having higher modularity value. A modification of modularity to overcome this problem has been proposed in [73] with the concept of *modularity density*, defined as:

$$D = \sum_{i=1}^{k} \frac{L(C_i, C_i) - L(C_i, \overline{C_i})}{|C_i|} \tag{14}$$

The first term is the average inner degree of a community C_i, which is twice the number of edges in C_i divided its

number of nodes, while the second is the average out degree of C_i, that is the number of edges having a node inside C_i and the other node outside C_i, divided by the number of nodes of C_i. The authors prove that modularity density has a number of advantages with respect to modularity, such as detecting communities of different size.

A quality measure of a community C that maximizes the in-degree of the nodes belonging to C has been defined in [86] as follows.

$$score(C) = \frac{\sum_{i \in C} \left(\frac{1}{|C|} \sum_{j \in C} A_{ij} \right)^{\alpha}}{|C|} \times \sum_{i,j \in C} A_{ij} \qquad (15)$$

where α is a positive real-valued *resolution parameter* controlling the size of the communities, $\mid C \mid$ is the cardinality of C, $\frac{1}{|C|} \sum_{j \in C} A_{ij}$ is the fraction of edges connecting node i to the other nodes in C, and $\sum_{i,j \in C} A_{ij}$ is the double of the number of edges connecting vertices inside C, i.e the number of 1 entries in the adjacency sub-matrix of A corresponding to C.

The *community score* of a clustering $\mathcal{C} = \{C_1, \ldots C_k\}$ is defined as

$$\mathcal{CS} = \sum_{i}^{k} score(C_i) \qquad (16)$$

The concept of *community fitness* $\mathcal{P}(\mathcal{C})$ of a community C has been introduced in [69] as

$$\mathcal{P}(\mathcal{C}) = \sum_{i \in C} \frac{k_i^{in}(C)}{(k_i^{in}(C) + k_i^{out}(C))^{\alpha}} \qquad (17)$$

where α is a *resolution parameter*. When $k_i^{out}(C) = 0 \ \ \forall i$, $\mathcal{P}(\mathcal{C})$ reaches its maximum value for a fixed α.

In the literature many other scoring functions, such as conductance, expansion, cut ratio, have been defined to capture the concept of community [110], and classified with respect to their characteristics of being based on either internal or external connectivity, on a combination of both, and on a network model. These other criteria did not receive much attention as functions to optimize, probably because of obtaining solutions of lower quality when compared to modularity.

Modularity [83], and its extensions [11], [96], are based on the idea that a random graph does not present community structure. Thus, the existence of communities can be uncovered by a comparison between the edge density of a group of nodes and the expected density of this group of nodes if they were attached randomly. Though this quality function is one of the most popular functions, because of the resolution limit problem, it may be biased towards network partitions with small communities merged into larger communities [40].

Community score relies on internal connectivity, and community fitness on both internal and external connectivity. Both functions have introduced a resolution parameter α that allows the exploration of community structure at different levels of granularity, thus overcoming the resolution limit problem of modularity. However, which value of α gives the best partition is not an easy task, also because, as will be discussed in Section 8, a formal definition of community does not exist [41].

In the next section, multiobjetive approaches that integrate these fitness functions to unveil different aspects of community structure, are described.

7 Multiobjective Optimization

The approaches described so far optimize only one of the objective functions reported in the previous section. Though these single-objective methods have obtained very good results on both artificial and real world networks, the intuitive notion of community that the number of edges inside a community should be much higher than the number of edges connecting to the remaining nodes of the graph, has two different objectives: maximizing the internal links and minimizing the external links. Thus, the community detection problem is naturally formulated with multiple competing objectives.

The first proposal of using a multiobjective framework to uncover community structure has been presented by Pizzuti in [87], [89]. In particular, the method maximizes the *community score* (formula (16)) and minimizes the *community fitness* (formula (17)), and uses as multiobjective framework the *Non-dominated Sorting Genetic Algorithm (NSGA-II)* proposed by Deb et al. in [28]. NSGA-II builds a population of competing individuals and ranks them on the basis of nondominance. The solution of the Pareto front having the highest value of modularity is chosen as final result. It is worth to outline that a main characteristic of the multiobjective approach is that the set of Pareto optimal solutions reveals the hierarchical organization of the network, where solutions with a higher number of groups are included in solutions having a lower number of communities. This peculiarity gives a great chance to analyze the network at various hierarchical levels and study communities with different modular levels.

A variation to this method has been proposed by Agrawal [2]. The objectives to minimize are

$$\begin{cases} f_Q = 1 - Q \\ f_{QCS} = f_Q + \frac{10}{(1 - \mathcal{CS})} \end{cases} \qquad (18)$$

where Q is the modularity (formula (11)) and $\mathcal{CS}$ is the community score (formula (16)). The weight 10, as the authors state, has been obtained empirically.

Shi et al. [101], [99] observed that the modularity formula $Q = \sum_{i=1}^{k} \frac{L(C_i,C_i)}{2m} - \left(\frac{L(C_i,V)}{2m} \right)^2$ is composed of two terms, where the left term considers the number of internal links of communities, thus it should be maximized, while the right one should be minimized because it includes the connections within different communities. To obtain the first objective, communities should be densely connected, to obtain the second one, the network should be divided in many groups with small total degree. In order to minimize two objectives, the first term is redefined as

$$intra(C) = 1 - \sum_{i=1}^{k} \frac{L(C_i, C_i)}{2m} \qquad (19)$$

and

$$inter(C) = \sum_{i=1}^{k} \left(\frac{L(C_i, V)}{2m} \right)^2 \quad (20)$$

These two objectives balance the tendency of each other's to increase or decrease the number of communities. If the number of communities increases, the number of edges inside each community diminishes, thus the first term of modularity diminishes, consequently $intra(C)$ augments, while $inter(C)$ diminishes. When, instead, the number of communities diminishes, $inter(C)$ increases, since the inter-connections between communities increases. Using them as the two objectives to optimize thus, as the authors state, avoids convergence to trivial solutions. Regarding the model selection from the Pareto front, the authors use two approaches: one chooses the solution having the maximum modularity value, the other introduces the concept of *Max-Min distance* between models. This strategy generates a random network $\mathcal{N}'$ with the same scale of the real network $\mathcal{N}$, and obtains the Pareto front CF of $\mathcal{N}'$. Then the distance between the two Pareto front solutions is computed as

$$dist(C, C') = \sqrt{(intra(C) - intra(C')^2 + (inter(C) - inter(C')^2} \quad (21)$$

where C and C' are solutions from the real and the random Pareto front, then

$$S_{Max-Min} = maxarg\{min\{dist(C, C') \mid C' \in CF\}\}$$

Gong et al. [51] followed a similar approach to that of Shi et al. [99] by splitting the modularity density formula in two. Thus, the first term, called *Negative Ratio Association (NRA)* is

$$NRA = -\sum_{i=1}^{k} \frac{L(C_i, C_i)}{|C_i|} \quad (22)$$

and the second term, called *Ratio Cut (RC)*, is

$$RC = \sum_{i=1}^{k} \frac{L(C_i, \overline{C_i})}{|C_i|} \quad (23)$$

Wu and Pan [108] proposed enriching a multiobjective evolutionary algorithm with a local search procedure to improve the solution. They adopt the *Nondominated Neighbor Immune* algorithm ($NNIA$) [50] as optimization mechanism, label-based representation of individuals along with one-way crossover and neighbor-based mutation, and the $inter(C)$ and $intra(C)$ objective functions of Shi et al. [99]. The local search procedure is executed after the application of crossover and mutation operators to the current nondominated individuals of the Pareto front, and uses a label propagation rule to change class membership of nodes.

A multiobjective evolutionary algorithm that obtains both separated and overlapping communities has been proposed by Liu et al. [77]. The main novelty of this approach is the introduction of the permutation-based representation described in Section 4.4. To obtain both separated and overlapping communities the authors optimize three functions:

$$\begin{cases} f_{quality}(\mathbf{A}) = (\sum_{i=1}^{k} \mathcal{P}(\mathcal{C})_i)/k \\ f_{separated}(\mathbf{A}) = - \mid V_{overlapping} \mid \\ f_{overlapping}(\mathbf{A}) = \sum_{i \in V_{overlapping}}^{k} min \frac{k_i^c}{k_i} \end{cases} \quad (24)$$

where $\mathcal{P}(\mathcal{C})_i$ is the community fitness of [69] (formula (17)), $V_{overlapping}$ is the set of nodes belonging to more than one community, and k_i^c is the number of edges connecting node i with community c. This method uses the *NSGA-II* framework and applies neither crossover nor mutation, but only the reverse operator on the permutation component $\mathcal{A}\langle\mathbf{P}\rangle$ (formula (5)) of a chromosome.

Multiobjective evolutionary approaches, analogously to single objective ones, are able to discover community structures of quality comparable with, or even better than, those obtained by computational methods not based on evolutionary computation. The choice of the objectives to optimize should take into account the suggestions given by Shi et al. in [100], where a comparison of several objective functions in a multiobjective framework has been performed. Eleven functions have been considered, and a correlation analysis revealed that couples of negatively correlated objectives give better results of positively correlated fitness functions. The authors experimented that negative correlation has opposite influence on the number of communities, thus it enhances diversity and avoids obtaining trivial solutions. Optimizing pairs of positively correlated objectives, instead, is equivalent to a single objective approach, thus it does not yield any benefit to the algorithm. It is worth pointing out that neither of the above methods performs a correlation analysis among the objectives, also because many methods are antecedent to this analysis.

8 Single Objective versus Multiobjective

The concept of community in a network is based on the idea that internal connections are dense, while few ties should exist with the rest of the graph. A formal definition of community, however, does not exist. Wasserman and Faust [107] defined four general properties that cohesive groups of nodes should satisfy to be considered communities: *complete mutuality*, *closeness or reachability*, *frequency of internal ties*, *relative tie frequencies among group members versus non-members*. The quality of a community can be defined with respect to one, or more than one, of these properties, and it measures how well the properties are satisfied. Single objective methods optimize a single property, while multiobjective approaches simultaneously optimize competing objectives [22]. The two approaches present advantages and disadvantages. Single objective optimization identifies a single best solution that gives insights on the graph organization, however this solution could be biased towards a particular structure inherent inside the criterion to optimize. Optimizing multiple objectives, on the other hand, allows a simultaneous evaluation of community

structure from different perspectives, but then it is the user's responsibility to choose a solution. Consider for example the toy network of Figure 1. By maximizing modularity the solution obtained divides the network into the three groups $\{\{1,2,3\},\{4,5,6,7\},\{8,9,10,11,12\}\}$. However, by optimizing the two objectives of community score and community fitness, we obtain two solutions. One is the same division into three communities, the other one merges the first two communities giving $\{\{1,2,3,4,5,6,7\},\{8,9,10,11,12\}\}$. As can be observed from Figure 1, this second solution is actually a possible and plausible solution that gives a different view of group organization. It is worth pointing out that, for single layer networks the choice of single or multiple objectives can depend on the application domain. However, for other types of network models, such as multilayer networks, described in the following, many objective methods seem to fit better. For example, the evolution of dynamic networks with temporal smoothness is well represented as a multiobjective problem optimizing snapshot quality and temporal cost, as will be clear in Section 10.

9 Signed Networks

Signed networks are an extension of networks to model the relationships between individuals that, actually, can be either positive or negative, such as like-dislike, friends-enemies. Positive links denote friendly relations, while negative links represent antagonistic relations. Detecting community structure on these kinds of networks is an important research topic since it allows us to determine instability inside relationships, and, consequently, to predict changes in group organization.

In order to deal with signed networks, Gomez et al. [46] extended modularity as follows:

$$Q_S = \frac{1}{2m^+ + 2m^-} \sum_{i,j \in V} \left(A_{i,j} - \left(\frac{a_i^+ a_j^+}{2m^+} - \frac{a_i^- a_j^-}{2m^-}\right) \right) \delta(C_i, C_j) \tag{25}$$

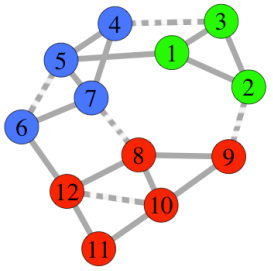
where A is the weighted adjacency matrix associated with the graph $G = (V, E, W)$ modeling a signed network, m^+ and m^- are the number of positive and negative entries in A, a_i^+ and a_i^- are the positive degree and the negative degree of node i, respectively.

A signed version of the toy network of Figure 1, along with the corresponding adjacency matrix, is shown in Figure 7.

A multiobjective approach that detects communities in a signed network has been proposed by Amelio and Pizzuti in [5], [8]. The goal of obtaining communities having dense intra-connections and most edges within clusters positive, while sparse inter-connections and most of these edges negative, is achieved by optimizing the concepts of signed modularity and *frustration*, introduced by Doreian and Mrvar [30].

Frustration $F(C)$ of a community C is defined as the sum of the number of negative edges between nodes inside the same community and the number of positive edges between nodes into different communities.

$$F(C) = \sum_{i,j \in V} A_{i,j}^- \delta(c_i, c_j) + A_{i,j}^+ (1 - \delta(c_i, c_j)) \tag{26}$$



$$A = \begin{pmatrix}
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 1 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 1 & 0
\end{pmatrix}$$

Fig. 7. An example of signed network with the corresponding adjacency matrix. Dashed edges denote negative connections.

Li et al. [71] performed a comparative analysis of four evolutionary and memetic algorithms. *EA-SN* adopts a label based representation, one-way crossover, a mutation operator that randomly changes the neighbor of a node with one of its positively connected nodes, and signed modularity as objective function; *CSA-SN* expands the clonal expansion operator of [53] to signed networks; EA_{HC}*-SN* and CSA_{HC}*-SN* integrate the hill climbing strategy of [49] in a multiobjective algorithm that optimizes signed modularity and modularity density, extended for signed networks. The authors showed that CSA_{HC}*-SN* performs better than the other methods.

Liu et al. [75] used the same representation scheme proposed in [77] to define a multiobjective evolutionary method to find communities in signed networks. The two objectives to optimize are based on the concepts of positive and negative cluster similarity between two neighboring nodes, introduced by Huang et al. [62], and extended to signed links. The objectives to maximize are the following:

$$\begin{cases} f_{pos-in}(\mathbf{C} = \{C_1, \ldots, C_k\}) = \frac{1}{m} \left(\sum_{i=1}^{k} \frac{P_{in}^{C_i}}{P_{in}^{C_i} + P_{out}^{C_i}} \right) \\ f_{neg-out}(\mathbf{C} = \{C_1, \ldots, C_k\}) = \frac{1}{m} \left(\sum_{i=1}^{k} \frac{N_{out}^{C_i}}{N_{in}^{C_i} + N_{out}^{C_i}} \right) \end{cases} \tag{27}$$

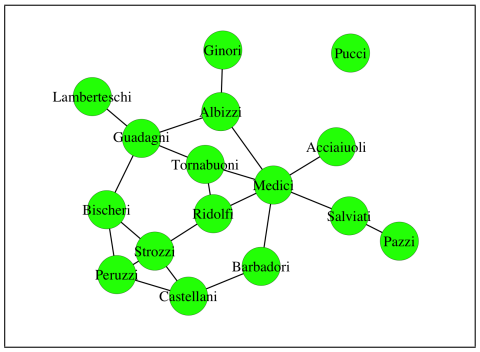
where $P_{in}^{C_i}$ and $P_{out}^{C_i}$ are the positive internal and external similarity of a community, while $N_{in}^{C_i}$ and $N_{out}^{C_i}$ are the negative internal and external similarity.

The similarity between two nodes i and j is defined as

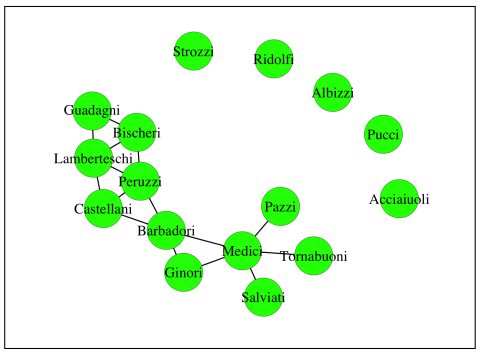
$$s_{signed}(i,j) = \frac{\sum_{x \in \Gamma(i) \cap \Gamma(j)} \psi(x)}{\sqrt{\sum_{x \in \Gamma(i)} w^2(i,x)} \times \sqrt{\sum_{x \in \Gamma(j)} w^2(j,x)}}$$

where

$$\psi(x) = \begin{cases} 0 & \text{if } w(i,x) < 0 \text{ and } w(j,x) < 0 \\ w(i,x) \times w(j,x) & \text{otherwise} \end{cases} \tag{28}$$



(a) (business relation)



(b) (marriage relation)

Fig. 8. An example of a multilayer network with two elementary layers.

Notice that, in [8] a correlation analysis of signed modularity and frustration revealed that these two objectives are negatively correlated, while a correlation analysis of the two objectives employed by [75] produced a positive correlation value.

10 Multilayer Networks

The representation of complex networks with graphs consisting of single static connections between couples of nodes has been universally adopted by researchers for many years. Recently, however, the need of richer models able to represent the variety of interconnections of real-world systems has led to the investigation of networks with multiple types of connections, the so-called *multilayer networks* [27], [68], [12]. Each layer represents a combination of different features of the network, called $aspects$ or $facets$. Thus, for each aspect a, there is a set of elements L_a, where each element is called an *elementary layer*. A layer will then be obtained by a combination of elementary layers from all the aspects.

More formally, a multilayer network $\mathcal{M}$ is defined as a quadruple [68]:

$$M = (V_M, E_M, V, \mathbf{L})$$

where V is the set of nodes, $\mathbf{L} = \{\mathbf{L_a}\}_{\mathbf{a=1}}^{\mathbf{l}}$ is a sequence of sets of elementary layers L_a, $V_M \subseteq V \times L_1 \times \ldots \times L_l$ contains only the set of combinations of nodes and elementary layers effectively present in a layer, $E_M \subseteq V_M \times V_M$ is a set of couples of possible combinations. Nodes could be connected to any other both inside the same layer and across layers. When the network has only one aspect with multiple types of edges and the same set of nodes, the network is called *multiplex* or *multirelational*. An example of a multiplex network, taken from [107], having two types of relationships, namely business and marriage, regarding Florentine families, can be seen in Figure 8. Notice that the connections between the same nodes appearing in both layers are implicit.

Though the interest in multilayer networks is rapidly increasing, there are still few approaches that detect communities in these kinds of networks [80], [104]. As pointed out in [68], the development of community detection methods for multilayer networks is just at the beginning. Also, the concept of community is not well-defined. Mucha et al. [80] generalized modularity for multilayer networks, while Tang et al. [104] introduced the notion of *shared latent community structure*, that is a division of nodes that optimizes the same criterion for each dimension.

As regards evolutionary methods, there are few proposals. In [6] multiplex networks are considered by extending both the locus-based representation and modularity. The extended representation is such that an individual $I = \{I_1, \ldots, I_d\}$ of the population is composed by a set of d elements $I_s, 1 \leq s \leq d$, each element I_s being the locus-based representation of the corresponding layer s. The concept of modularity for multilayer networks is defined by combining the modularity values computed for each layer in such a way that the value for each layer is influenced by the values of all the other layers. The main drawbacks of this method are the computation time needed to compute the fitness function and the space requirements.

Other proposals concentrated mainly on the dynamic aspect of networks. In fact, a dynamic or temporal network can be considered as a multilayer network restricted to two aspects. The first aspect $L_1 = \{T^1, \ldots, T^T\}$ represents the temporal information, i. e. the time in which a connection between two nodes occurred, while the second one $L_2 = \{D_1, \ldots, D_d\}$, gives the type of interaction among nodes. The set of combinations of a fixed elementary layer $T^t \in L_1$ with all the elementary layers $D_j \in L_2$, $j = 1, \ldots, d$, will be called *multiplex (or multidimensional) network at time* t, and denoted as $\mathcal{T}^t = \{\mathcal{N}_1^t, \mathcal{N}_2^t, \ldots, \mathcal{N}_d^t\}$, where each $\mathcal{N}_i^t$ is the network representing one of the elementary layers of L_2. A *temporal or dynamic multilayer network* is defined as a sequence $\mathcal{DM} = \{\mathcal{T}^1, \ldots, \mathcal{T}^T\}$ of networks, where each $\mathcal{T}^t$, $t = 1, \ldots, T$ is a snapshot of the network at time t, referred as timestamp or timestep.

In this context, there are two types of proposals. In the former [38], [66], [52], [39], methods consider only one type of interaction of the aspect L_2, i.e. $d = 1$, in the latter $d > 1$ [9]. All these methods are based on the concept of *evolutionary clustering* introduced by Chakrabarti et al. in [18] for data clustering. Evolutionary clustering is a framework assuming that abrupt changes of clustering in a short time period are not desirable, thus it $smooths$ each community over time. For smoothing, a cost function composed by two sub-costs, *snapshot cost* ($\mathcal{SC}$) and *temporal cost* ($\mathcal{TC}$), is defined. The snapshot cost $\mathcal{SC}$ measures how well a community structure $\mathcal{CC}^t$ represents the data at time t. The temporal cost $\mathcal{TC}$

measures how similar the community structure $\mathcal{CC}^t$ is with the previous clustering $\mathcal{CC}^{t-1}$. A specialized version of this function in the context of dynamic networks has been introduced in [74] as follows:

$$cost = \alpha \cdot \mathcal{SC} + (1 - \alpha) \cdot \mathcal{TC} \quad (29)$$

where α is an input parameter fixed by the user to emphasize one of the two objectives. When α = 1 the approach returns the clustering without temporal smoothing. When α = 0, however, the same clustering of the previous time step is produced, i.e. $\mathcal{CC}^t = \mathcal{CC}^{t-1}$. Thus, a value between 0 and 1 is used to control the preference degree of each sub-cost.

In [38], [39] the detection of community structure with temporal smoothness has been formulated as a *multiobjective optimization problem* where the first objective is the maximization of the snapshot quality, and the second objective is the minimization of the temporal cost. Several fitness functions have been experimented to optimize snapshot quality, such as $modularity$, *community score*, $conductance$, and *normalized cut*.

The *Normalized Mutual Information*, a well known entropy measure in information theory [25], has been employed as second objective to minimize the temporal cost $\mathcal{TC}$. $NMI(\mathcal{CC}^t, \mathcal{CC}^{t-1})$ measures the similarity between the community structure $\mathcal{CC}^t$ at the current time step t and the previous one $\mathcal{CC}^{t-1}$.

The normalized mutual information $NMI(A, B)$ of two partitions A and B, is defined as:

$$NMI(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} C_{ij} log(C_{ij}N/C_{i.}C_{.j})}{\sum_{i=1}^{c_A} C_{i.} log(C_{i.}/N) + \sum_{j=1}^{c_B} C_{.j} log(C_{.j}/N)} \quad (30)$$

where C is the confusion matrix whose element C_{ij} is the number of nodes of the community $A_i \in A$ that are also in the community $B_j \in B$, c_A (c_B) is the number of groups in the partitioning A (B), $C_{i.}$ ($C_{.j}$) is the sum of the elements of C in row i (column j), and N is the number of nodes. If $A = B$, $NMI(A, B) = 1$. If A and B are completely different, $NMI(A, B) = 0$.

A main advantage of this approach is that the parameter α, that must trade-off the benefit of maintaining a consistent clustering over time (temporal cost) with the cost of deviating from an accurate representation of the current data (snapshot cost), is automatically determined during the computation of the non-dominated solutions.

A variation of this approach, with the same objective functions of modularity and NMI, that uses as multiobjective optimization method the *Nondominated Neighbor Immune* $NNIA$ algorithm [50] has been proposed by of Gong et al. [52]. Moreover, the same authors [78] extend the framework of multiobjective evolutionary algorithm based on decomposition [51] to deal with dynamic networks by optimizing again modularity and NMI. Chen et al. [20] use the same framework by changing the first objective with modularity density.

A multiobjective method based on immigrant schemes, that replaces a proportion of the population with the aim of maintaining population diversity and adapting to changes, has been proposed by Kim et al. [66]. The method introduces three immigrant strategies inside the multiobjective evolutionary algorithm NSGA-II [28] to deal with networks that can increase the number of edges and/or nodes with time. A chromosome, using the locus-based representation, is extended with new genes if the number of nodes augments. The objective functions to optimize are the *min-max cut* introduced in [29] and the *global silhouette index* [93]. A comparison among the three immigrant schemes has been performed on a synthetic data set. However, no comparison with classical community detection methods is present, thus the capability of the approach to discover high quality clusters is not known. Moreover, as the authors point out, the method is applicable only to networks that grow, but no node or edge can disappear, which is not a realistic scenario.

In [6] the evolutionary clustering framework is modified by introducing the concepts of *facet quality* $\mathcal{FQ}$, and *sharing cost* $\mathcal{SQ}$. Facet quality guarantees that the clustering found for the i-th dimension under consideration maximizes the quality function as much as possible, while the sharing cost means that the clustering of the current facet agrees as much as possible with the clustering obtained for the previously considered i-1 dimensions. In [9] an extension that encompasses both time and multiple dimensions is defined. In this extended framework, a shared community structure among the networks $\mathcal{N}_i^t$ of $\mathcal{T}^t$ is obtained by iteratively optimizing both facet quality and sharing cost. The community structure obtained for the last layer d is considered the best sharing community structure among the d layers.

Let $\mathcal{CC}_1^t, \ldots, \mathcal{CC}_d^t$ be the community structures obtained for each elementary layer of a network $\mathcal{T}^t = \{\mathcal{N}_1^t, \mathcal{N}_2^t, \ldots, \mathcal{N}_d^t\}$, at timestamp t. The concept of shared community structure introduced in [104] is formalized as follows: $\mathcal{CC} = \{C_1, \ldots, C_k\}$ is a shared community structure of $\mathcal{T}^t$ if the two functions are maximized:

$$f_q(\mathcal{CC}, \mathcal{N}_i^t), i = 1, \ldots, d \quad (31)$$

$$f_s(\mathcal{CC}, \mathcal{CC}_i^t), i = 1, \ldots, d \quad (32)$$

where (31) is the quality function computed on the network $\mathcal{N}_i^t$ by using the community structure $\mathcal{CC}$, and (32) is a function that computes the similarity between $\mathcal{CC}$ and the community structure obtained for $\mathcal{N}_i^t$ by maximizing f_q, independently from the other layers. f_q and f_s can be any functions computing the quality of a clustering and the similarity between two clusterings, respectively.

Thus, the method searches for a community structure $\mathcal{CC}$ that maximizes a fitness function on each elementary layer $\mathcal{N}_i^t$, while taking into account the similarity with the clustering obtained on the other layers. This framework is then utilized between couples of consecutive timestamps $t - 1$ and t, by resorting to the dynamic evolutionary approach where the temporal cost $\mathcal{TC}$ is guaranteed by considering the similarity between the community structure $\mathcal{CC}^{t-1}$ obtained for the previous timestamp and that found for the first elementary layer $\mathcal{CC}_1^t$ of the current timestamp.

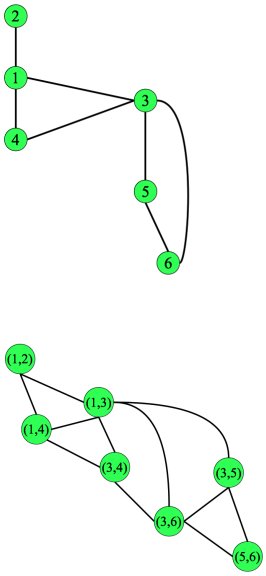


Fig. 9. An example network with 6 nodes and 7 edges, and the corresponding line graph with 7 nodes and 11 edges.

11 Overlapping Community Detection

In the last few years a lot of effort in defining efficient and efficacious methods for community detection has been directed to finding disjoint communities. However, in real world networks the membership of an entity to many groups is very common, thus the interest in defining methods for finding overlapping communities has been growing. It is worth pointing out that the representation schemes described in Section 6, except for the permutation-based representation, do not allow a node to be a member of more than one community, thus only recently a number of evolutionary computation methods, both single-objective and multiobjective, have been proposed to find overlapping communities.

In [88] the concept of *line graph* has been exploited to define a link clustering method that detects overlapping communities by partitioning the set of links, rather than the set of nodes. The *line graph* $L(G)$ of an undirected graph G is another graph $L(G)$ such that each vertex of $L(G)$ represents an edge of G, and two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G. A line graph represents the adjacency between edges of G. By applying a community detection method to the line graph generates an overlapping division of the original interaction network, thus allowing nodes to be present in multiple communities.

An example of network and the corresponding line graph is shown in Figure 9. A community division of the line graph into the two clusters $C_1 = \{(1,2),(1,3),(1,4),(3,4)\}$, C_2=$\{(3,5),(3,6),(5,6)\}$, gives the division $\{\{1,2,3,4\},\{3,5,6\}\}$ of the original graph in which node 3 appears in both the clusters.

The method described in [88] adopts the locus-based representation on the line graph. This means that each gene corresponds to an edge of G, and the value it contains is one of the neighboring edges, that is an edge having a node in common. The algorithm finds a community structure of the line graph $L(G)$ and evaluates its quality by computing the community score of the corresponding division of the original graph G.

Shi et al. [97] proposed a similar method that clusters links, which is equivalent to using the line graph since two edges are connected only if they share a node. However, their method uses as fitness function the concept of *partition density* proposed by Ahn et al. [3], which is based on the number of links, thus its evaluation can be done on the original graph. Let $\{P_1,\ldots,P_C\}$ be the partition of the links in C subsets. Each subset P_c has $m_c =\mid P_c \mid$ links and $n_c =\mid \cup_{e_{ij}\in P_c}\{i,j\} \mid$ nodes. The *link density* of P_c is defined as

$$D_c = \frac{m_c-(n_c-1)}{n_c(n_c-1)/2-(n_c-1)} = 2\frac{m_c-(n_c-1)}{(n_c-2)(n_c-1)} \quad (33)$$

D_c is thus the normalization of the number of links m_c by the minimum and maximum number of possible l links between n_c connected nodes. It is assumed that $D_c = 0$ when $n_c = 2$. The *partition density* PD is the average of the D_c, weighted by the fraction of present links:

$$PD = \frac{2}{m}\sum_c m_c \frac{m_c-(n_c-1)}{(n_c-2)(n_c-1)} \quad (34)$$

Another proposal that clusters the set of links by optimizing the two objective functions of *modularity density* D (formula (14)) and *extended modularity* EQ (formula (13)) has been proposed by Du et al. [32]. Yuxin et al. [115], instead, consider the community fitness, and define the *negative fitness sum (NFS)* and the *unfitness (US)* of a community structure by substituting the numerator of *community fitness* (formula (17)) with the sum of external degrees. Let

$$unfit(C) = \sum_{i\in C}\frac{k_i^{out}(C)}{(k_i^{in}(C)+k_i^{out}(C))^{\alpha}} \quad (35)$$

be the external connection density of a community $C \in \mathcal{C} = \{C_1,\ldots C_k\}$, then the two modified objective functions are the following:

$$\begin{cases} NFS = -k - \sum\limits_{C\in\mathcal{C}} \mathcal{P}(\mathcal{C}) \\ US = \sum\limits_{C\in\mathcal{C}} unfit(C) \end{cases} \quad (36)$$

To improve the convergence, the algorithm adopts an initialization strategy that expands a seed node by merging adjacent edges until the community fitness improves. This process is repeated until all edges are assigned to a community.

12 Other Bio-Inspired Approaches

In recent years, bio-inspired computation has attracted the interest of many researchers in several fields to solve optimization problems. The basic principle of these methods is *self-organization*, that is if a system is allowed to evolve for a

sufficiently long period, self-organized structures may emerge [113]. In the last decade, a relevant number of these new metaheuristic algorithms have been employed for community detection, including *swarm intelligence* [114], in particular *Particle Swarm Optimization (PSO)* [65] and *Ant Colony Optimization (ACO)* [31], *Firefly* [111] and *Bat* [112] algorithms.

Particle Swarm Optimization. PSO is an optimization technique based on the swarm behavior of bird and fish schooling [65]. Each particle i is characterized by two components: the position vector x_i and the velocity vector v_i. Particles are attracted towards the best position g^* of the swarm, and its personal best position x^*, while moving randomly at the same time. The new velocity and position vectors are updated as

$$v_i^{t+1} = v_i^t + \alpha\epsilon_1(g^* - x_i^t) + \beta\epsilon_2(x^* - x_i^t) \tag{37}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{38}$$

where α and β are acceleration parameters, and ϵ_1, ϵ_2 are two random vectors taking values in the range [0,1]. Cai et al. [15] applies the particle swarm method to detect communities in signed networks by optimizing the signed modularity. The position vector represents a partition of the network where x_i is the community label of node i. The update rules of the particle status are redefined to fit in the discrete context as follows:

$$v_i^{t+1} = \Gamma(\omega v_i^t + \alpha\epsilon_1(g^* \oplus x_i^t) + \beta\epsilon_2(x^* \oplus x_i^t)) \tag{39}$$

$$x_i^{t+1} = x_i^t \Theta v_i^{t+1} \tag{40}$$

where ω is an inertia weight [102] that, when its value is high, it is better for global search, while, when small, for local search, $\oplus$ is the xor operator. The Γ function assigns 1 to v_i if $x_i \geq 1$, 0 otherwise. The operator Θ is a neighbor operator that updates the position of a node by considering its neighbors. The same method is applied for unsigned networks in [14], and for signed networks in Gong et al. [47]. In this latter paper the problem has been formulated as a multiobjective optimization problem where the objective functions are obtained by extending the *Negative Ratio Association (NRA)* and *Ratio Cut (RC)*, introduced in [51].

Thus, the signed network clustering problem is reformulated as the minimization of the objectives:

$$\begin{cases} SRA = -\sum_{i=1}^{k} \frac{L^+(C_i,C_i) - L^-(C_i,C_i)}{|C_i|} \\ SRC = \sum_{i=1}^{k} \frac{L^+(Ci,\overline{Ci}) - L^-(Ci,\overline{Ci})}{|C_i|} \end{cases} \tag{41}$$

where $L^+(C_i,C_i) = \sum_{i\in C_i, j\in Ci} A_{ij}$, $A_{ij} > 0$ and $L^-(C_i,C_i) = \sum_{i\in C_i, j\in Ci} \mid A_{ij} \mid$, $A_{ij} < 0$.

A multiobjective variant of these methods, also for signed networks, has been proposed by Li et al. [72]. The objective functions are the same of Gong et al. [47] (formula (41)). The main differences with Cai et al. [15] and Gong et al. [47] are the definition of the Γ function, and a replacement operation that substitutes only a subset of the solutions in the new generation. The new Γ function is defined as:

$$\Gamma(y) = \begin{cases} 1 & if\ rand(0,1) \leq 1/(1+e^{-y}) \\ 0 & othewise \end{cases} \tag{42}$$

Ant Colony Optimization. *ACO* mimics the foraging behavior of ants [31]. Ant movement is controlled by pheromone, which evaporates over time, and its concentration is an indicator of the quality of the solution. In these algorithms there are two important issues: the probability of choosing a route and the evaporation rate of pheromone. The probability of choosing a route from node i to node j is given by the rule:

$$p_{ij} = \frac{\phi_{ij}^\alpha d_{ij}^\alpha}{\sum_{i,j=1}^{n} \phi_{ij}^\alpha d_{ij}^\alpha} \tag{43}$$

where $\alpha > 0$, $\beta > 0$ are the influence parameters, ϕ_{ij} is the pheromone concentration of the route between i and j, and d_{ij} is a heuristic function that reflects the tendency of selecting the edge from i to j.

Chen et al. [19] proposed an algorithm based on ant colony optimization that adopts the concept of *associate degree* between nodes as a heuristic function. Let $A = (A_{ij})$ be the adjacency matrix of the network and $A^k = (A_{ij}^k)$ the number of k-step paths connecting two nodes. The associate degree is defined as:

$$d_{ij} = k_1 A_{ij}^1 + k_2 A_{ij}^2 + \ldots + k_p A_{ij}^p \tag{44}$$

where p is a positive constant integer, $k_i, i = 1,\ldots,p$ are coefficients. The pheromone updating is then performed according to the formula:

$$\phi_{ij}(t+1) = \rho\phi_{ij}(t) + \sum_{k=1}^{m} \Delta\phi_{ij}^k(t) \tag{45}$$

m is the number of ants, $\Delta\phi_{ij}^k = C \times Q(S_k)$, with C a constant, and $Q(S_k)$ the modularity value of the solution S_k, if i and j are in the same community, 0 otherwise.

A different heuristic information, based on the Pearson correlation, has been proposed by Honghao et al. [60]. Given two nodes i and j, the Pearson correlation is defined as:

$$C(i,j) = \frac{\sum_{l\in V}(A_{il} - \mu_i)(A_{jl} - \mu_j)}{n\sigma_i\sigma_j} \tag{46}$$

where A_{il} is the lth element of the ith row in the adjacency matrix, μ_i the average and σ_i the standard deviation. Then

$$d_{ij} = \frac{1}{1+e^{-C(i,j)}} \tag{47}$$

The formula for pheromone updating uses the best modularity value obtained at the current iteration, and only edges whose nodes belong to the current best solution receive this value.

Firefly algorithm. This metaheuristic method is based on the flashing patterns and behavior of fireflies [111]. It assumes that fireflies are unisexual, they are attracted to other fireflies proportionally to their brightness, the brightness is determined

by the landscape of the objective function. The movement of a firefly is defined as

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha \epsilon_i^t \quad (48)$$

where the second term in the formula is the attractiveness function, with β_0 the attractiveness when the distance $r = 0$. The third term is a randomization with parameter α, and ϵ_i^t is a vector of random numbers.

Amiri et al. [10] adopted this approach to design a multi-objective method that optimizes community score and community fitness, by introducing some variations to improve solutions. They maintain an external repository to store the non-dominated solutions and apply a niching mechanisms to preserve diversity. Moreover, they assume that the fireflies can have different sex, and the parameter α is dynamically tuned by using a chaotic sequences, as proposed in [17], instead of random ones.

Bat Algorithm. This approach is inspired by the behavior of bats and their capability of echolocation, a type of sonar, that allows them to detect prey and to avoid obstacles [112]. Bats emit sound waves whose loudness gradually reduces while frequency of emission gets faster, as the distance to the prey is closer. If x_i is the position of the bat at time t, f_i the frequency varying in the interval $[f_{min}, f_{max}]$, v_i the velocity, i.e. the change degree of its position, r the emission rate and A the loudness, these values are updated with the rules:

$$f_i = f_{min} + (f_{max} - f_{min})\epsilon, \quad v_i^{t+1} = v_i^t + (x_i^t - x^*)f_i \quad (49)$$

where x^* is the current best solution.

$$x_i^{t+1} = x_i^t + v_i^t, \quad A_i^{t+1} = \alpha A_i^t, \quad r_i^t = r_i^0[1 - exp(-\beta t)] \quad (50)$$

Hassan et al. [57] observed that the bat algorithm cannot directly be applied for community detection. Thus, a discretization and redesign of the bat movement is necessary before using the approach. Let the vector state of an artificial bat be $x = (x_1, \ldots, x_n)$, the velocity vector $v = (v_1, \ldots, v_n)$, and $g(x_i)$ the group assignment of node i. The distance to the current best solution x^* is computed as :

$$d_i = (x_i - x_i^*) = \begin{cases} 1 & if\ g(x_i) \neq g(x_i^*) \\ 0 & \text{if}\ g(x_i) = g(x_i^*) \end{cases} \quad (51)$$

Then the new position value is computed as

$$x_i^{new} = \begin{cases} x_i^* & if\ v_i \geq 1 \\ x_i & otherwise \end{cases} \quad (52)$$

Another discrete bat algorithm has been proposed by Song et al. [103] to discover communities by making discrete the values of x and v. The new discrete velocity formula is defined as follows. Let $Sig(v_i^t) = 1/(1 + exp(-v_i^t))$ be the sigmoid function, and $rand$ a random number in the range (0,1), then $v_i^t = 1$ if $Sig(v_i^t) > rand$, 0 otherwise.

13 Conclusion

Evolutionary computation has been successfully applied to many real-world problems as an optimization technique, and showed to be competitive also for the study of complex networks. The paper presented an up-to-date review on evolutionary methods for community detection. Though research in this context is rather recent, there has been a surge of interest and many methods have been proposed to deal with complex networks. A main contribution of the survey is that it systematizes the several approaches presented in the literature by providing the basic common principles for the design of methods that solve the problem of uncovering community structure. In particular, the most popular representation schemes, along with the crossover and mutation operators apt for them are described in detail, by discussing advantages or drawbacks of each, and the most common fitness functions adopted by methods are also analyzed. A categorization in single objective and multiple objectives optimization has been given. Though many surveys for community detection are available in the literature [42], [92], [41], [24], [84], [109], [79], [91], [56], [1], [7], [90], specific reviews for evolutionary based approaches are few [16]. The paper sensibly extends the work of Cai et al. [16] by including multilayer networks, and by giving a more detailed description of individual representation and associated operators.

To summarize the approaches described in the paper, Table 1 reports the single objective methods, and Table 2 the multiobjective ones. For each approach, the kind of representation, crossover, mutation, fitness function employed, and if overlapping is allowed, are reported. When present, local search strategies adopted to improve the methods are included. For the multiobjective methods, also the type of network and the multiobjective evolutionary optimization method used are added. Moreover, Table 3 summarizes the other bio-inspired approaches. The tables, for all the methods, report also the real-world networks and/or the kind of synthetic dataset used for evaluating the quality of results. Finally, Table 4 contains the list of all these networks, along with the web address from which it is possible to download the network. Links to the source codes for the methods, when available, are reported in the References Section.

The review highlighted that, though there is a lot of work on networks representing a single type of interaction, further investigation is necessary as regards overlapping community detection and multilayer networks. In fact, new representation schemes are desirable to efficiently deal with overlapping communities, and novel ideas to tackle the dynamic and multi-relational aspects of networks.

Another aspect that it is worth pointing out is that evolutionary algorithms are time consuming, thus, though they are competitive with the non-evolutionary approaches as regards the quality of the obtained solution, they are not able to cope with large networks, very common in the current big data era, where networks with millions of nodes are generated. The need of developing parallel implementations, considering the inbuilt parallel characteristics of evolutionary methods, to accelerate the response times is an important issue to make them comparable with the other methods available in the literature. Moreover, more efficient representations, such as variable length chromosomes, should be investigated to reduce both time and space requirements.

The survey can be a starting point for researchers interested

TABLE 1
A summarization of single-objective methods.

METHOD	REPR.	FITNESS	CROSSOVER	MUTATION	LOCAL SEARCH	OVERLAP	NETWORKS
Tasgin and Bingol [106] (2007)	label	Q	one-way	random	clean-up	no	ZKC, KPB, GN
Firat et al. [36] (2007)	medoid	pair-wise distances sum	two-point	random	-	no	synthetic
Gog et al. [44] (2007)	label	Q	collaborative	random	-	no	ZKC
Pizzuti [86] (2008)	locus	$\mathcal{CS}$	uniform	neighbor	best	no	ZKC, BD, ACF, KPB, GN
Pizzuti [88] (2009)	locus	$\mathcal{CS}$	uniform	neighbor	best	yes	ZKC, BD, ACF, KPB, GN
Li et al. [70] (2009)	label	Q	one-way	random	n_l copies	no	ZKC, BD, LM, Ucinet, Pajek
He et al. [58] (2009)	label	Q	multi-individual	majority neig. label	-	no	ZKC, ACF, GN
Shi et al. [98] (2009)	locus	Q	two-point	random	-	no	ZKC, ACF, CN
Jin et al. [64] (2010)	locus	Q	uniform	neighbor	marginal node	no	ZKC, BD, ACF, KPB, JM, WA, SFI
Chira and Gog [21] (2011)	locus	$\mathcal{CS}$	collaborative	random	-	no	ZKC, BD, KPB
Gong et al. [49] (2011)	label	D	two-way	neighbor	neighbor label	no	ZKC, BD, ACF, KPB,LFR
Gong et al. [48] (2012)	label	D	two-way	neighbor label	-	no	ZKC, BD, ACF, KPB, LFR
Jia et al. [63] (2012)	label	Q	binary	rand/1	clea-up	no	ZKC, ACF, GN
Shang et al. [95] (2013)	label	Q	two-way	random	simulated annealing	no	ZKC, BD, ACF, KPB, LFR
Liu et al. [76] (2013)	locus	Q	uniform	neighbor	inside mutation	no	GN, LFR, ZKC, BD, ACF, KPB, JM, WA, SFI
Shi et al. [97] (2013)	locus	Q	uniform	neighbor	inside mutation	yes	ZKC, BD, ACF, KPB, WA, LM, PG, LFR
Zadeh et al. [116] (2015)	locus	$\mathcal{CS}$	uniform	neighbor	no	no	ZKC, BD, KPB

in approaching the problem of community detection with computational models inspired by evolution in nature. The knowledge of a different computational paradigm with respect to traditional approaches can be beneficial to explore new strategies and principles to deal with this problem.

ACKNOWLEDGMENT

This work has been partially supported by MIUR D.D. n. 0001542, under the project $BA2KNOW - PON03PE\_00001\_1$.

REFERENCES

[1] Charu Aggarwal and Karthik Subbian. Evolutionary network analysis: A survey. *ACM Comput. Surv.*, 47(1):10:1–10:36, May 2014.

[2] Rohan Agrawal. Bi-objective community detection (BOCD) in networks using genetic algorithm. In *Proceedings of the 4th International Conference on Contemporary Computing, IC3 2011, Noida, India, August 8-10, 2011.*, pages 5–15, 2011.

[3] Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.

[4] Réka Albert and Albert lászló Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys*, 74(1):47–97, 2002.

[5] Alessia Amelio and Clara Pizzuti. Community mining in signed networks: a multiobjective approach. In *Advances in Social Networks Analysis and Mining 2013, ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*, pages 95–99, 2013.

[6] Alessia Amelio and Clara Pizzuti. Community detection in multidimensional networks. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014, Limassol, Cyprus, November 10-12, 2014*, pages 352–359, 2014.

[7] Alessia Amelio and Clara Pizzuti. *Overlapping Community Discovery Methods: A Survey. In Social Networks: Analysis and Case Studies*, pages 105–125. Springer, Vienna, 2014.

[8] Alessia Amelio and Clara Pizzuti. An evolutionary and local refinement approach for community detection in signed networks. *International Journal on Artificial Intelligence Tools*, 25(4):1–44, 2016. [Code available at: http://staff.icar.cnr.it/pizzuti/codes.html].

[9] Alessia Amelio and Clara Pizzuti. Evolutionary clustering for mining and tracking dynamic multilayer networks. *Computational Intelligence*, 33(2):181–209, 2017. [Code available at: http://staff.icar.cnr.it/pizzuti/codes.html].

[10] Babak Amiri, Liaquat Hossain, John W. Crawford, and Rolf T. Wigand. Community detection in complex networks: Multi-objective enhanced firefly algorithm. *Knowl.-Based Syst.*, 46:1–11, 2013.

[11] Alex Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9:176, 2007.

[12] Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. *Phys. Rev. E*, 89(3):032804, 2014.

[13] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[14] Qing Cai, Maoguo Gong, Lijia Ma, Shasha Ruan, Fuyan Yuan, and Licheng Jiao. Greedy discrete particle swarm optimization for large-scale social network clustering. *Inf. Sci.*, 316:503–516, 2015.

[15] Qing Cai, Maoguo Gong, Bo Shen, Lijia Ma, and Licheng Jiao. Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Netw.*, 58:4–13, October 2014.

[16] Qing Cai, Lijia Ma, Maoguo Gong, and Dayong Tian. A survey on network community detection based on evolutionary computation. *International Journal on Bio-Inspired Computation*, 8(2):84–98, 2016.

[17] R. Caponetto, L. Fortuna, S. Fazzino, and M. G. Xibilia. Chaotic sequences to improve the performance of evolutionary algorithms. *Trans. Evol. Comp*, 7(3):289–304, June 2003.

[18] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proc. of the 12th ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD'06)*, pages 554–560, 2006.

[19] Bolun Chen, Ling Chen, and Yixin Chen. Detectingcommunitystructureinnetworksbasedonantcolonyoptimization. In *Int. Conf. on Information & Knowledge Engineering, (WORLDCOMP'12), July 16-19, Las Vegas, Nevada,USA*, pages 247–253, 2012.

[20] Guoquiang Chen, Yuping Wang, and Jingxuan Wei. A new multiobjective evolutionary algorithm for community detection in dynamic complex networks. *Mathematical Problems in Engineering*, (161670), 2013.

[21] Camelia Chira and Anca Gog. Collaborative community detection in complex networks. In *Hybrid Artificial Intelligent Systems - 6th International Conference, HAIS 2011, Wroclaw, Poland, May 23-25, 2011, Proceedings, Part I*, pages 380–387, 2011.

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

[23] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms - Second Edition*. Mit Press, 2007.

[24] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for community discovery methods in complex networks. *Stat. Anal. Data Min.*, 4(5):512–546, October 2011.

[25] Leon Danon, Jordi Duch, Alex Arenas, and Albert Díaz-Guilera. Comparing community structure identification. *Journal of Statistical Mechanics*, 2005(9):P09008, 2005.

[26] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *Trans. Evol. Comp*, 15(1):4–31, February 2011.

[27] M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M.A. Porter, S. Gómez, and A. Arenas. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

[28] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[29] C.H.Q Ding, He Xiaofeng, Zha Hongyuan, Ming Gu, and H.D Simon. A min-max cut algorithm for graph partitioning and data clustering. In *IEEE International Conference on Data Mining (ICDM'01), IEEE*, pages 107–114, 2001.

[30] Patrick Doreian and Andej Mrvar. A partitioning approach to structural balance. *Social Networks*, 18:149–168, 1996.

[31] Marco Dorigo and Gianni Di Caro. Ant colony optimization: A new meta-heuristic. In *Proceedings of the Congress on Evolutionary Computation*, pages 1470–1477. IEEE Press, 1999.

[32] Jingfei Du, Jianyang Lai, and Chuan Shi. Multi-objective optimization for overlapping community detection. In *Advanced Data Mining and Applications - 9th International Conference, ADMA 2013, Hangzhou, China, December 14-16, 2013, Proceedings, Part II*, pages 489–500, 2013.

[33] Matthias Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition, 2005.

[34] Emanuel Falkenauer. *Genetic Algorithms and Grouping Problems*. John Wiley & Sons, Inc., New York, NY, USA, 1998.

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

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

[37] Lawrence J. Fogel, Alvin J. Owens, and Michael J. Walsh. *Intelligence Through Simulated Evolution*. Wiley, 1966.

[38] Francesco Folino and Clara Pizzuti. A multiobjective and evolutionary clustering method for dynamic networks. In *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2010, Odense, Denmark, August 9-11, 2010*, pages 256–263, 2010.

[39] Francesco Folino and Clara Pizzuti. An evolutionary multiobjective approach for community discovery in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1838–1852, 2014. [Code available at: http://staff.icar.cnr.it/pizzuti/codes.html].

[40] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proc. National Academy of Science, USA*, 104(36), 2007.

[41] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[42] Santo Fortunato and Claudio Castellano. *Community Structure in Graphs*, pages 1141–1163. Springer, New York, 2009.

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

[44] Anca Gog, D. Dumitrescu, and Beat Hirsbrunner. Community detection in complex networks using collaborative evolutionary algorithms. In *9th European Conference on Artificial Life (ECAL'07), Springer*, pages 886–894, 2007.

[45] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[46] Sergio Gómez, Pablo Jensen, and Alex Arenas. Analysis of community structure in networks of correlated data. *Physical Review*, E80(1):016114, 2009.

[47] Maoguo Gong, Qing Cai, Xiaowei Chen, and Lijia Ma. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Trans. Evolutionary Computation*, 18(1):82–97, 2014.

[48] Maoguo Gong, Qing Cai, Yangyang Li, and Jingjing Ma. An improved memetic algorithm for community detection in complex networks. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, Australia, June 10-15, 2012*, pages 1–8, 2012.

[49] Maoguo Gong, Bao Fu, Licheng Jiao, and Haifeng Du. A memetic algorithm for community detection in networks. *Physical Review*, E84:056101, 2011.

[50] Maoguo Gong, Licheng Jiao, Haifeng Du, and Liefeng Bo. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation*, 16(2):225–255, 2008.

[51] Maoguo Gong, Lijia Ma, Qingfu Zhang, and Licheng Jiao. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Physica A*, 391(15):4050–4060, 2012.

[52] Maoguo Gong, Ling-Jun Zhang, Jing-Jing Ma, and Licheng Jiao. Community detection in dynamic social networks based on multiobjective immune algorithm. *Journal of Computer Science and Technology*, 27(3):455–467, 2012.

[53] Maoguo Gong, Lining Zhang, Licheng Jiao, and Wenping Ma. Differential immune clonal selection algorithm. In *Proceedings of the Intern. Symposium on Intelligent Signal Processing and Communication Systems*, pages 666–669, 2007.

[54] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *International Conference on Advances in Social network Analysis and Mining (ASONAM'10)*, pages 176–183, 2010.

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

[56] Steve Harenberg, Gonzalo Bello, L. Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *WIREs Comput. Stat.*, 6(6):426–439, 2014.

[57] Eslam Ali Hassan, Ahmed Ibrahem Hafez, Aboul Ella Hassanien, and Aly A. Fahmy. A discrete bat algorithm for the community detection problem. In *10th International Conference on Hybrid Artificial Intelligence Systems, HAIS*, pages 188–199, 2015.

[58] Dongxiao He, Zhe Wang, Bin Yang, and Chunguang Zhou. Genetic algoritm with ensemble learning for detecting community structure in complex networks. In *4th International Conference on Computer Sciences and Convergence Information Technology, IEEE*, pages 702–707, 2009.

[59] John H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, USA, 1992.

[60] Chang Honghao, Feng Zuren, and Ren Zhigangg. Community detection using ant colony optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2013, Cancún, Mexico, 20-23 June, 2013*, pages 3072–3078, 2013.

[61] Eduardo Raul Hruschka, Ricardo J. G. B. Campello, Alex A. Freitas, and André C. Ponce Leon F. De Carvalho. A survey of evolutionary algorithms for clustering. *Trans. Sys. Man Cyber Part C*, 39(2):133–155, March 2009.

[62] Jianbin Huang, Heli Sun, Yaguang Liu, Qinbao Song, and Tim Weninger. Towards online multiresolution community detection in large-scale networks. *PlosOne*, 6(8):e23829, 2011.

[63] Guanbo Jia, Zixing Cai, Mirco Musolesi, Yong Wang, Dan A. Tennant, Ralf J. Weber, John K. Heath, and Shan He. Community detection in social and biological networks using differential evolution. In *Revised Selected Papers of the 6th International Conference on Learning and Intelligent Optimization - Volume 7219*, LION 6, pages 71–85, 2012. [Code available at: http://www.cs.bham.ac.uk/ szh/DECDandCCDECD.zip].

[64] Di Jin, Dongxiao He, Dayou Liu, and Carlos Baquero. Genetic algorithm with local search for community mining in complex networks. In *22nd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2010, Arras, France, 27-29 October 2010 - Volume 1*, pages 105–112, 2010.

[65] James Kennedy and Russell Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[66] Keehyung Kim, Robert Ian (Bob) McKay, and Byung Ro Moon. Multiobjective evolutionary algorithms for dynamic social network clustering. In *Genetic and Evolutionary Computation Conference, GECCO 2010, Proceedings, Portland, Oregon, USA, July 7-11, 2010*, pages 1179–1186, 2010.

[67] Min-Soo Kim and Jiawei Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc. VLDB Endow.*, 2(1):622–633, August 2009.

[68] Mikko Kivelä, Alexandre Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[69] Andrea Lancichinetti, Santo Fortunato, and Janos Kertész. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 11(033015), 2009.

[70] Shuzhuo Li, Yinghui Chen, Haifeng Du, and Marcus W. Feldman. A genetic algorithm with local search strategy for improved detection of community structure. *Complexity*, 15(4):53–60, 2009.

[71] Y. Li, J. Liu, and C. Liu. A comparative analysis of evolutionary and memetic algorithms for community detection from signed networks. *Soft Computing*, 18(2):329–348, 2014.

[72] Zhaoxing Li, Lile He, and Yunrui Li. A novel multiobjective particle swarm optimization algorithm for signed network community detection. *Applied Intelligence*, 44:621–633, 2016.

[73] Zhenping Li, Shihua Zhang, Rui-Sheng Wang, Xiang-Sun Zhang, and Luonan Chen. Quantitative function for community detection. *Physical Review E*, 77:036109, 2008.

[74] Yu-Ru Lin, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data*, 3(2, Article 18), 2009.

[75] C. Liu, J. Liu, and Z. Jiang. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *IEEE Transactions on Cybernetics*, 44(12):2274–2287, 2014. [Code available at: http://see.xidian.edu.cn/faculty/liujing/publication.htm].

[76] Dayou Liu, Di Jin, Carlos Baquero, Dongxiao He, Bo Yang, and Qiangyuan Yu. Genetic algorithm with a local search strategy for discovering communities in complex networks. *Int. J. Computational Intelligence Systems*, 6(2):354–369, 2013.

[77] Jing Liu, Weicai Zhong, Hussein A. Abbass, and David G. Green. Separated and overlapping community detection in complex networks using multiobjective evolutionary algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010*, pages 1–7, 2010.

[78] Jingjing Ma, Jie Liu, Wenping Ma, Maoguo Gong, and Licheng Jiao. Decomposition-based multiobjective evolutionary algorithm for community detection in dynamic networks. *The Scientific World Journal*, (402345), 2014.

[79] Fragkiskos D. Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95 – 142, 2013.

[80] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[81] Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[82] Mark E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review*, E69:066133, 2004.

[83] Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review*, E69:026113, 2004.

[84] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Min. Knowl. Discov.*, 24(3):515–554, May 2012.

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

[86] Clara Pizzuti. GA-NET: a genetic algorithm for community detection in social networks. In *Proc. of the 10th International Conference on Parallel Problem Solving from Nature (PPSN 2008), Springer*, pages 1081–1090, 2008. [Code available at: http://staff.icar.cnr.it/pizzuti/codes.html].

[87] Clara Pizzuti. A multi-objective genetic algorithm for community detection in networks. In *ICTAI 2009, 21st IEEE International Conference on Tools with Artificial Intelligence, Newark, New Jersey, USA, 2-4 November 2009*, pages 379–386, 2009.

[88] Clara Pizzuti. Overlapped community detection in complex networks. In *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009*, pages 859–866, 2009. [Code available at: http://staff.icar.cnr.it/pizzuti/codes.html].

[89] Clara Pizzuti. A multiobjective genetic algorithm to find communities in complex networks. *IEEE Trans. Evolutionary Computation*, 16(3):418–430, 2012. [Code available at: http://staff.icar.cnr.it/pizzuti/codes.html].

[90] Clara Pizzuti and Simona E. Rombo. Algorithms and tools for protein-protein interaction networks clustering, with a special focus on population-based stochastic methods. *Bioinformatics*, 30(10):1343–1352, 2014.

[91] Michel Plantié and Michel Crampes. *Survey on Social Community Detection In Social Media Retrieval*, pages 65–85. Springer, London, 2013.

[92] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

[93] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computing and Applied Mathematics*, 20(1):53–65, 1987.

[94] C. D. Gellant S. Kirkpatrick and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[95] Ronghua Shang, Jing Bai, Licheng Jiao, and Chao Jin. Community detection based on modularity and an improved genetic algorithm. *Physica A*, 392:1215–1231, 2013.

[96] H. Shen, X. Cheng, K. Cai, and M.-B. Hu. Detect overlapping and hierarchical community structure in networks. *Physica A*, 388(8):1706–1712, 2009.

[97] Chuan Shi, Yanan Cai, Di Fu, Yuxiao Dong, and Bin Wu. A link clustering based overlapping community detection algorithm. *Data & Knowledge Engineering*, 87:394 – 404, 2013.

[98] Chuan Shi, Yi Wang, Bin Wu, and Cha Zhong. *A New Genetic Algorithm for Community Detection*, pages 1298–1309. Springer Berlin Heidelberg, 2009.

[99] Chuan Shi, Zhenyu Yan, Yanan Cai, and Bin Wu. Multi-objective community detection in complex networks. *Appl. Soft Comput.*, 12(2):850–859, 2012.

[100] Chuan Shi, Philip S. Yu, Zhenyu Yan, Yue Huang, and Bai Wang. Comparison and selection of objective functions in multiobjective community detection. *Computational Intelligence*, 30(3):562–582, 2014.

[101] Chuan Shi, Cha Zhong, Zhenyu Yan, Yanan Cai, and Bin Wu. A multi-objective approach for community detection in complex network. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010*, pages 1–8, 2010.

[102] Yuhui Shi and Russel C. Eberhart. Empirical study of particle swarm optimizer. In *Proceedings of the Congress on Evolutionary Computation*, pages 1945–1950. IEEE Press, 1999.

[103] Anping Song, Mingbo Li, Xuehai Ding, Wei Cao, and Ke Pu. Community detection using discrete bat algorithm. *IAENG International Journal of Computer Science*, 43(1):37–43, 2016.

[104] Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data Mining and Knowledge Discovery*, 25(1):1–33, 2012.

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

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

[107] S. Wasserman and K. Faust. *Social Network Analysis Methods and Applications*. Cambridge University Press, 2009.
[108] Peng Wu and Li Pan. Multi-objective community detection based on memetic algorithms. *PLOS One*, 10(5):e0126845, 2015.
[109] Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *ACM Computing Survey*, 45(4), 2013.
[110] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.*, 42(1):181–213, January 2015.
[111] Xin-She Yang. Firefly algorithms for multimodal optimization. In *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications*, SAGA'09, pages 169–178, 2009.
[112] Xin-She Yang. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pages 65–74, 2010.
[113] Xin-She Yang. *Nature-Inspired Optimization Algorithms*. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1st edition, 2014.
[114] Xin-She Yang, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet Karamanoglu. *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1st edition, 2013.
[115] Zhao Yuxin, Li Shenghong, and Jin Feng. Overlapping community detection in complex networks using multi-objective evolutionary algorithms. *Computational and Applied Mathematics*, 36(1):749–768, 2017.
[116] Pooya Moradian Zadeh and Ziad Kobti. A multi-population cultural algorithm for community detection in social networks. *Procedia Computer Science*, 52:342–349, 2015.

Clara Pizzuti received the Master's degree in Mathematics from University of Calabria, Italy, and a Ph.D in Science from the Radboud Universiteit Nijmegen, NL. She is senior researcher at the Institute of High Performance Computing and Networking (ICAR) of the Italian National Research Council(CNR), where she leads the Smart Data and Models research laboratory. From 1995 to 2010 she was contract professor in the department of Computer Science at the University of Calabria. Her research interests include evolutionary computation, knowledge discovery in databases, data mining, data streams, bioinformatics, social network analysis and mining.

TABLE 2
A summarization of multiobjective methods.

METHOD	REPR.	FITNESS	CROSS	MUTATION	MOEA	OV	NETWORKS
Pizzuti [87] (2009), [89] (2012)	locus	$F1$: $\mathcal{CS}$ $F2$: $\mathcal{P}(\mathcal{C})$	uniform	neighbor	NSGA-II	no	undirected ZKC, BD, ACF, KPB, Erdos, SC, DB
Folino and Pizzuti [38] (2010)	locus	$F1$: $\mathcal{CS}$ $F2 : NMI$	uniform	neighbor	NSGA-II	no	dynamic FO, Kim and Han
Kim et al. [66] (2010)	locus	$F1$: *min-max cut* *F2: silhouette*	uniform	-	NSGA-II	no	dynamic YTV
Agrawal [2] (2011)	locus	$F1$: f_Q $F2$: f_{QCS}	uniform	neighbor	NSGA-II	no	undirected ZKC, BD, ACF,GN
Shi et al. [101], (2010), [99] (2012)	locus	$F1$: $intra(C)$ $F2$: $inter(C)$	two-point	random	PESA-II	no	undirected, ZKC, ACF, KPB WA, LM, CN, CM, NS, PG, GN
Liu et al. [77] (2010)	$(\mathbf{A}\langle\mathbf{P}\rangle, \mathbf{A}\langle\mathbf{C}\rangle)$	$F1$: $f_{quality}(\mathbf{A})$ $F2$: $f_{separated}(\mathbf{A})$ F3: $f_{overlapping}(\mathbf{A})$	-	-	NSGA-II	yes	undirected ZKC, BD ACF, KPB
Gong et al. [52] (2012)	locus	$F1$: Q $F2 : NMI$	uniform	neighbor	NNIA	no	dynamic CPC, FO
Gong et al. [51] (2012)	locus	$F1$: NRA $F2 : RC$	uniform	neighbor	MOEA/D	no	undirected ZKC, BD, ACF, KPB,GN
Amelio and Pizzuti [5] (2013), [8] (2016)	locus	$F1$: Q_S $F2 : F(C)$	uniform	neighbor	NSGA-II	no	signed GGS, SPP, WE, signed LFR
Chen et al. [20] (2013)	locus	$F1$: D $F2 : NMI$	uniform	neighbor	NSGA-II	no	dynamic modified LFR
Du et al. [32] (2013)	locus	$F1 : PD$ $F2$: EQ	exchange one gene	random	PESA-II	no	undirected BD, ACF, LM
Folino and Pizzuti [39] (2014)	locus	$F1$: Q $F2 : NMI$	uniform	neighbor	NSGA-II	no	dynamic CPC, EM, Lin, Greene, Kim and Han
Amelio and Pizzuti [6] (2014)	locus	$F1$: Q $F2 : NMI$	uniform	neighbor	NSGA-II	no	multilayer YTC,ECCS,Tang,Greene
Ma et al. [78] (2014)	locus	$F1 : Q$ $F2 : NMI$	locus	neighbor	MOEA/D	no	dynamic CPC, FO, Kim and Han
Li et al. [71] (2014)	label	$F1$: Q_S $F2$: signed D	one-way	positive neighbor	NSGA-II	no	signed GGS, SPP, signed LFR
Liu et al. [75] (2014)	$(\mathbf{A}\langle\mathbf{P}\rangle, \mathbf{A}\langle\mathbf{C}\rangle)$	$F1$: f_{pos-in} $F2$: $f_{pos-out}$	-	-	NSGA-II	yes	signed GGS, SPP, signed LFR
Wu and Pan [108] (2015)	label	$F1$: $1 - Intra(C)$ $F2 : 1 - Inter(C)$	one-way	neighbor	NNIA	no	unsigned ZKC, BD, ACF, KPB, WA, LM, CN, CM, NS, PG, GN
Yuxin et al. [115] (2015)	locus	$F1$: NFS $F2 : NS$	uniform	neighbor	NSGA-II	no	unsigned ZKC, BD ACF, SFI, NS, PG, LFR
Amelio and Pizzuti [9] (2016)	locus	$F1$: Q $F2 : NMI$	uniform	neighbor	NSGA-II	no	dynamic, multilayer Greene, Tang, ECCS

TABLE 3
A summarization of bio-inspired methods

APPROACH	REFERENCE	FITNESS	NETWORKS
PARTICLE SWARM	Cai et al. [15]	Q_S	SPP, GGS, EGFR, MP, Yeast, EC
	Gong et al. [47] (2014)	$F1$: SRA $F2$: SRC	ZKC, BD, ACF SFI, NS, PG, LFR
	Cai et al. [14]	Q	ZKC, DB, ACF, SFI, EMC, NS, PG, PGP, LFR
	Li et al. [72]	Q_S	signed LFR, SPP, GGS, EGFR, MP, Yeast, EC
ANT COLONY	Chen et al. [19]	Q	ZKC, BD, ACF, KPB, GN
	Honghao et al. [60]	Q	ZKC, BD, ACF, KPB, LFR
FIREFLY	Amiri et al. [10]	$F1$: $\mathcal{CS}$ $F2$: $\mathcal{P}(\mathcal{C})$	ZKC, BD, ACF, KPB, LFR
BAT	Hassan et al. [57]	Q	ZKC, BD, ACF
	Song et al. [103]	Q	ZKC, ACF, KPB, GN

TABLE 4
Networks and reference web site to download.

NETWORK	WEB ADDRESS
Zackary's Karate Club (ZKC) Bottlenose Dolphins (BD) American College Football (ACF) Kreb's Political books (KPB) World adjacencies (WA) Santa Fe Institute (SFI) Les Misérables (LM) Celegans neural (CN) Netscience(NS) Power grid (PG)	http://www-personal.umich.edu/ mejn/netdata/
Erdos	http://www.oakland.edu/enp/thedata.html
Scientometrics (SC)	http://www.garfield.library.upenn.edu/histcomp/index.html
Jazz musicians (JM) Celegans methabolic (CM)	http://deim.urv.cat/ alexandre.arenas/data/welcome.htm
Florentine Families (FF) Eu. Conf. on Complex Systems (ECCS 2013)	http://deim.urv.cat/ manlio.dedomenico/data.php
Gahuku-Gama (GGS)	http://networkrepository.com/ucidata_gama.php
English Wikipedia (EW) E-mail communication (EMC)	http://konect.uni-koblenz.de/networks/elec http://konect.uni-koblenz.de/networks/arenas-email
Cell Phone Calls (CPC)	http://www.cs.umd.edu/hcil/VASTchallenge08/
Enron mail (EM)	ftp://ftp.isi.edu/sims/philpot/data/enronmysqldump.sql.gz
Football (FO)	http://www.jhowell.net/cf/scores/scoresindex.htm
You Tube Videos (YTV)	http://netsg.cs.sfu.ca/youtubedata
PPI Yeast	http://faculty.uaeu.ac.ae/nzaki/ProRank.htm
Macrophage (MP)	http://www.macrophages.com
Escherichia coli (EC)	http://regulondb.ccg.unam.mx
Slovene Parliamentary Party (SPP) Ucinet Pajek	http://vlado.fmf.uni-lj.si/pub/networks/data/soc/samo/stranke94.htm http://vlado.fmf.uni-lj.si/pub/networks/data/UciNet/UciData.htm http://vlado.fmf.uni-lj.si/pub/networks/pajek/data/gphs.htm
Wikipedia Elections WE	http://konect.uni-koblenz.de/networks/elec
GN benchmark LFR benchmark	https://sites.google.com/site/santofortunato/inthepress2
Directors Board (DB) You Tube contact network (YTC) Tang et al. [104] Lin et al. [74] Greene et al. [54] Kim and Han [67]	upon request to the authors

LIST OF FIGURES

LIST OF TABLES