Multiobjective Optimization and Local Merge for Clustering Attributed Graphs

Clara Pizzuti, Annalisa Socievole

***Abstract*—Methods for detecting community structure in complex networks have mainly focused on the network topology, neglecting the rich content information often associated with nodes. In the last years, the compositional dimension contained in many real world networks has been recognized fundamental to find network divisions which better reflect group organization. In this paper, we propose a multiobjective genetic framework which integrates the topological and compositional dimensions to uncover community structure in attributed networks. The approach allows to experiment different structural measures to search for densely connected communities, and similarity measures between attributes to obtain high intra-community feature homogeneity. An efficient and efficacious post-processing local merge procedure enables the generation of high quality solutions, as confirmed by the experimental results on both synthetic and real world networks, and the comparison with several state-of-the-art methods.**

***Index Terms*—Attributed graphs, community detection, multiobjective optimization, genetic algorithms.**

I. Introduction

GRAPHS constitute a powerful mechanism to model and analyze relationships of many real world systems. One of the most relevant activities on complex networks is the division of the nodes into groups, also called clusters or communities, satisfying some homogeneity criterion. Graph clustering for community discovery has been intensively investigated in the last decades, and a plenty of methods have been proposed [1] and applied in many different fields. The most popular and efficient methods, such as, for instance, [2] and [3], however, approached the problem by focusing only on the network topology, disregarding the content information often associated with the actors composing the network. Since the beginning of the studies in network data, Wasserman and Faust [4] pointed out that this kind of data includes two types of information, called variables or dimensions: structural and compositional. Structural variables measure the ties between pairs of actors and provide the topological structure of a network. Compositional variables measure the attributes of the single actors, such as race, gender, hobbies, etc. In real world social systems it has been observed that there exists a correlation between attribute values and connectivity, and that the *homophily* and *social influence* effects co-occur [5]. The former means that individuals are more likely to create relationships with others having similar attribute values, while the latter that people tend to modify their behavior to be like their friends. Thus, using only the topological structure might generate not accurate community divisions, missing important information.

Attributed graphs extend network models by enriching nodes and/or edges with a set of features that measure the characteristics of the actors contained in the network. In this paper we deal only with graphs having node attributes. These graphs are referred in the literature as *node-attributed graphs* [6], to distinguish them from those having attributes related to edges, and called *edge-attributed graphs*. In the following, for attributed graphs we mean node-attributed graphs. As outlined by Bothorel *et al.* [6], a good community division in attributed graphs has to optimize both the structural and compositional dimensions. The structural quality is the objective of classical community detection methods, that is communities having dense intra-cluster connections and sparse inter-cluster connections. The compositional quality can be achieved if the clusters contain nodes with similar characteristics. A balance between these two objectives is important in order to obtain both highly homogeneous and well connected groups of nodes.

In the last years, many methods for community detection in attributed graphs, based on different strategies, have been proposed. A detailed overview can be found in [6]. To obtain a community division that balances both structural and compositional quality there are two main approaches. The first one optimizes a single objective that combines the two quality functions in some way [7], [8], [9], the second one tries to simultaneously optimize both functions [10]. Single objective optimization identifies a single best solution. However, how to combine the two measures of link density and node similarity is a challenging problem because it is necessary to avoid that one of the two measures prevails on the other one, thus biasing the computation towards a particular structure inherent inside the dominating criterion. For instance, nodes with similar features could be far in the network, thus relying mainly on attribute similarity could produce sparse, eventually unconnected, groups of nodes. Optimizing multiple objectives, on the other hand, allows a simultaneous evaluation of community division from both the perspectives of link density and node similarity, which are two competing criteria to optimize. *Multiobjective evolutionary algorithms (MOEAs)* [11], in this context, offer an efficacious solution to the problem of community detection in attributed networks.

In this paper, a framework based on a multiobjective genetic algorithm [11] that combines the structural and compositional dimensions is proposed. The approach, named MOGA-@Net, *MultiObjective Genetic Algorithm for @ttributed Networks*, optimizes simultaneously the structural quality and the intra-

Clara Pizzuti and Annalisa Socievole are with the National Research Council of Italy (CNR), Institute for High Performance Computing and Networking (ICAR), Via Pietro Bucci, 8-9C, 87036 Rende (CS), Italy, e-mail: {clara.pizzuti,annalisa.socievole}@icar.cnr.it.

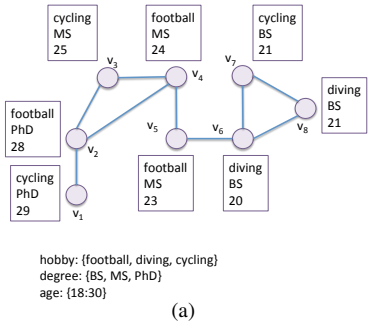
Manuscript received 2018; revised .

cluster node similarity. To this end, to maximize the first objective, we investigate three measures, well known in the community detection field, that search for high density communities. The second objective of intra-community feature homogeneity is defined according to similarity measures between attributes, chosen on the basis of the attribute type. Moreover, MOGA-@Net applies a very efficient and efficacious post-processing strategy that identifies those communities that can be merged to obtain solutions of high quality. Recently, the multiobjective evolutionary algorithm *MOEA-SA* has been proposed by Li *et al.* [10]. However, our approach sensibly differs from it in several aspects, as will be clear in the following.

The main contributions of the paper can be summarized as follows.

- A multiobjective framework for detecting community structure in attributed networks is presented. The framework considers and evaluates three well known objective functions (modularity, community score and conductance) to optimize the structural dimension and three node similarity measures (Jaccard, cosine, and Euclidean based similarity), to compute attribute homogeneity in order to optimize the compositional dimension.
- Though the multiobjective framework has been exploited by several authors for community detection on different kinds of networks [12], including attributed networks [10], our method performs a thorough evaluation on both synthetic and real world networks, showing the efficacy of multiobjective genetic algorithms to deal with this problem.
- The main novelty of the framework is the introduction of a post-processing local search procedure which identifies those communities that can be merged to provide higher quality community divisions. The merging strategy, as experiments highlight, reduces the number of communities of a solution and sensibly increases the evaluation measures of the method, often obtaining the ground-truth solution, or a solution very close to the real division. For instance, on the $Cora$ and $Citeseer$ networks, the merge procedure applied on the solutions of the Pareto Front returns a unique solution which corresponds to the known division in seven and six classes, respectively.
- An extensive experimentation on synthetic and real world attributed networks shows that MOGA-@Net obtains high quality solutions, and a very good performance when compared to eight non evolutionary state-of-the-art methods, and with *MOEA-SA*. Experimental results have pointed out that our method outperforms all the contestant methods, obtaining an improvement on synthetic networks between 25% (with respect to the Louvain method, which is the second best) and 350% (with respect to the SA-cluster method, which is the worst).

The paper is organized as follows. Section II defines the problem to solve and formalizes it as a multiobjective clustering problem. Section III reviews the most recent proposals on clustering attributed graphs. Section IV describes in detail the method. Section V reports the evaluation measures adopted to assess the performance of the considered methods. Section VI presents an extensive experimentation on synthetic networks for which the ground-truth is known, and on seven real world attributed networks. Section VII compares the solutions obtained by MOGA-@Net when the MOEA/D framework is adopted instead of the NSGA-II multiobjective evolutionary optimization framework. Section VIII, finally, concludes the paper and suggests future developments.



(a)

Fig. 1. Example of attributed network.

II. Problem Definition

In this section the definition of attributed graph is recalled and the community detection problem for these graphs is formalized as a multiobjective clustering problem.

Definition An *attributed graph* is a 4-tuple $G = (V, E, A, F)$ where $V = \{v_1, v_2, ..., v_n\}$ is a set of n vertices, $E = \{(v_i, v_j) : 1 \leq i, j \leq n, i \neq j\}$ is a set of m edges, $A = \{\alpha_1, \alpha_2, ..., \alpha_{\mathcal{A}}\}$ is the set of attributes (features), and $F = \{a_1, a_2, ..., a_{\mathcal{A}}\}$ is a set of functions. Each node $v_i \in V$ is characterized by a vector of feature values $A_{v_i} = [a_1(v_i), a_2(v_i), \ldots, a_{\mathcal{A}}(v_i)]$, obtained by the functions $a_\alpha : V \rightarrow D_\alpha$, $1 \leq \alpha \leq \mathcal{A}$, with D_α the domain of attribute α.

Figure 1 shows an attributed network with eight nodes and nine edges. The set of attributes is $A = \{\alpha_1 = hobby, \alpha_2 = degree, \alpha_3 = age\}$ with domains $D_{hobby} = \{football, diving, cycling\}$, $D_{degree} = \{BS, MS, PhD\}$, $D_{age} = \{20 : 30\}$. The set of functions $F = \{a_{hobby}, a_{degree}, a_{age}\}$ is such that $a_{hobby} : V \rightarrow D_{hobby}$, $a_{degree} : V \rightarrow D_{degree}$, $a_{age} : V \rightarrow D_{age}$. Node v_5, for instance, has the feature vector $A_{v_5} = \{a_{hobby}(v_4), a_{degree}(v_4), a_{age}(v_4)\}$=$\{football, MS, 23\}$. Notice that v_5 has one link with both v_4 and v_6. Thus, it could participate to either $\{v_1, v_2, v_3\}$ or $\{v_6, v_7, v_8\}$, However, by considering the attributes, it is more similar to v_4. Thus, a community detection method should include it into the cluster $\{v_1, v_2, v_3\}$ instead of $\{v_6, v_7, v_8\}$.

The objective of the community detection problem, also called clustering, in attributed graphs is to find a partition $\mathcal{C} = \{C_1, \ldots, C_k\}$ of the nodes of V such that:

1) intra-cluster density is high and inter-cluster density is low, and
2) nodes belonging to the same community are similar, while nodes of different communities are quite dissimilar.

Thus, for attributed networks, the objectives to optimize are two: the structural quality f_S and the intra-cluster homogeneity of node attributes f_A. The community detection problem in this kind of graphs can then be formulated as a *multiobjective clustering problem*. To optimize the structural quality different indexes, used in the literature to capture the intuition of network community, can be employed. Analogously, to maximize node homogeneity, several measures computing attribute similarity, depending on the kind of features, can be adopted. In the following, the definition of multiobjective clustering problem is introduced.

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

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

where $\Omega = \{\mathcal{C}_1, \ldots, \mathcal{C}_h\}$ is the set of feasible clusterings of a network, and $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_t\}$ is a set of t competing objectives that must be simultaneously optimized and obtained through the use of Pareto optimality theory [13]. Multiobjective optimization aims to the generation and selection of nondominated solutions, that is those solutions for which an improvement in one objective requires a degradation of another one. These solutions are called *Pareto-optimal*. The vector $\mathcal{F}$ maps the solution space into the objective function space. When the nondominated solutions are plotted in the objective space, they are called the *Pareto front*. The Pareto front represents the compromise solutions satisfying all the objectives as best as possible.

III. Related Work

In the last few years, several methods for detecting communities in attributed graphs have been proposed. Moreover, many researchers have shown that enriching a network with node attributes can help to detect more meaningful communities, and that structural and compositional components can complement each other when some information is missing or noise is present. For instance, [14] and [15] compared several algorithms using attributes and not using attributes, and showed that methods with attributes outperform those not exploiting them. Moreover, Newman and Clauset [16] demonstrated that attribute data improve the understanding of network structure. Recently, a survey by Bothorel *et al.* [6] describes and classifies state-of-the-art algorithms into different categories, depending on the adopted strategy.

A common strategy removes the attributes from nodes and stores their content on edges, thus transforming the original attributed graph into a weighted graph where edge weights represent the similarity measured on nodal attributes. Any clustering algorithm for weighted graphs can then be applied. Following this approach, Neville et al. [17], first compute the *matching coefficient* between nodes in order to quantify the number of attributes two nodes have in common, and then apply spectral clustering [18] to the resulting weighted graph. In another work, Steinhaeuser and Chawla [19] cluster attributed graphs having both discrete and continuous attributes by extending the matching coefficient. When the attribute is discrete, for each common attribute between two nodes, the edge weight is incremented by 1; for continuous attributes, a normalized distance between the attributes is computed and added. Finally, after having normalized edge weights, all nodes sharing edges with weight greater than a given threshold are inserted into the same cluster. Differently from the above works, when computing node similarity, Cruz *et al.* [7] use the concept of entropy to measure the similarity of nodes and define a method that maximizes the modularity and minimizes the entropy of a partition, since low entropy means groups with similar objects. The approach first performs modularity optimization, then moves nodes among communities to minimize the entropy, and repeats these two steps until no more changes are possible.

The second family of methods combines structural and attributes dimensions in several different ways. In [8], for example, Combe *et al.* define a distance measure between two nodes as the sum of the attribute distance, computed for the features with any measure, such as the Euclidean or the cosine distance, and a structural distance given by the shortest path between such nodes. A hierarchical agglomerative clustering is applied on the distance matrix computed accordingly. Dang and Viennet [20] propose to extend the modularity to include the similarity among node attributes and build a k-nearest graph for finding communities with the $Louvain$ method [3].

Community detection methods based on statistical inference attempt to fit a generative model to the observed data with the aim of finding the most likely arrangements of communities. Li *et al.* [21] focus on the problem of clustering networks of documents exploiting both the content (topics) and their references/citations. Xu *et al.* [9] propose a method named $BAGC$, Bayesian Attributed Graph Clustering, that combines structural and compositional attributes by developing a Bayesian probabilistic model for attributed graphs. The model generates all the possible combinations of a graph with features and assigns a probability for each possible clustering of the vertices, with the aim to find the clustering that gives the highest probability.

Other methods include walk-based approaches [22], methods focusing on the discovery of significative patterns [23], hybrid methods [24] that use either the structure data, or the attribute data depending on the type of graph. Zhou *et al.* [22] proposed a method, named *SA-Cluster* that builds an attribute augmented graph by adding to the initial graph new vertices representing the attributes. An edge between a graph vertex and an attribute vertex is present if the graph vertex has that attribute and the edge weight between them reflects the importance of that attribute. The method uses the neighborhood random walk model on the attributed augmented graph to compute a unified distance measure between vertices (i.e., combination of structural closeness and attribute similarity). Then it applies a framework similar to the k-medoid clustering method that iteratively re-assigns nodes to communities until the overall unified distance measure improves.

It is worth pointing out that the methods of Neville *et al.* [17], Xu *et al.* [9], Zhou *et al.* [22] need as input parameter the number of clusters to find.

Elhadi and Agam [24] presented the *Selection* method that, instead of combining structure and attribute data, it makes

the choice to use either the structure data, or the attribute data depending on the type of graph (clear or ambiguous structure). This method detects the boundaries between clear and ambiguous graph structure content and relies on the structure-only method when the graph has a clear structure, while it applies the attribute-only method when the graph has an ambiguous structure. Thus it executes the $Louvain$ method in the former case, and the *k-means* in the latter case.

Though many approaches based on evolutionary computation have been proposed for different types of networks, such as unweighted/weighted [25], [26], [27], [28], [29], dynamic [30], [31], signed [32], [33], multi-layer [34] (see [12] for a recent review), proposals for attributed networks are very few. An evolutionary algorithm that optimizes a fitness function based on the concept of connection significance has been presented by He and Chan [35]. The significance of a connection between a couple of node attributes is computed by taking into account the frequency of occurrences of edges connecting nodes with those attributes.

Recently, Li *et al.* [10] proposed a multiobjective evolutionary algorithm for attributed networks, named *MOEA-SA*, which employs the modularity of Newman and Girvan [2] as objective optimizing link connections, and an attribute similarity function S_A that measures the quality of feature node similarity. S_A is defined as:

$$S_A = \frac{\sum_{k=1}^{c} \sum_{i,j \in C_k, i<j} 2s(i,j)}{\sum_{k=1}^{c} r_k(r_k - 1)} \tag{1}$$

where c is the number of communities, r_k is the number of nodes inside the community C_k, $s(i,j)$ is the similarity of nodes i and j computed as the cosine similarity when nodes have multiple attributes, while, if a node has a single discrete attribute, $s(i,j) = 1$ if the attribute value of nodes i and j are the same, 0 otherwise. The authors do not consider continuous attributes. The method uses the locus-based representation [36] for the population initialization in order to obtain an initial good solution, then it decodes it into the label-based representation, where each gene value contains the class label of the community it belongs to, and uses this representation for the remaining steps of the algorithm. The genetic operators are a two-way crossover [37], followed by a neighborhood correction strategy to repair genes assigned to a wrong community, and multi-individual-based mutation operator, that changes the value of a gene by taking into account the values of other two chromosomes chosen by binary tournament selection. Finally, a hill-climbing strategy, proposed in [37], is performed at each generation on the individual having the best modularity value, and the returned final solution is the knee point [38] of the Pareto Front.

The method we propose is also based on multiobjective optimization, but the differences between MOGA-@Net and *MOEA-SA* are significant, as will be clear in the next section.

IV. The MOGA-@Net method

In this section the algorithm MOGA-@Net is described, along with the objective functions optimized by the method, the individual representation, and the genetic operators.

A. Objective Functions

As described in Section II, to obtain a good clustering of an attributed graph, the two objectives of structural quality and intra-cluster node similarity must be optimized. Yang and Leskovec [39] classified topological measures in four categories, based on internal connectivity, external connectivity, combination of internal and external connectivity, and based on network model, i. e. $modularity$. We experimented several of these measures and then those performing the best from each category have been chosen, excluding the external category ($expansion$ and *Cut Ratio*) because giving poor results. Thus, as first objective to optimize we considered $modularity$ for network model, *community score* for the internal category, and *conductance* for the combined class. The same reasoning applies to the second objective of intra-community feature homogeneity. We experimented different similarity measures, and then adopted classical similarity measures between attributes, chosen on the basis of attribute type. It is worth pointing out that the multiobjective framework receives as input parameters the kind of fitness functions to use, thus it is possible to try any kind of combination of the two objectives. In the following, the topological and attribute homogeneity measures are described.

Topological measures. The measures we consider are the *modularity* function of Newman and Girvan [2], the *community score* [40], and the *conductance* [41].

Modularity: let k be the number of obtained clusters, l_c is the total number of edges joining vertices inside the community C, and d_c is the sum of the degrees of the nodes of C. The modularity Q is defined as

$$Q = \sum_{c=1}^{k} [\frac{l_c}{m} - (\frac{d_c}{2m})^2] \tag{2}$$

The first term of each summand is the fraction of edges inside a community, and the second one is the expected value of the fraction of edges that would be in the network if edges fall at random without regard to the community structure. Values approaching 1 indicate strong community structure.

Community score: let M be the adjacency matrix of G, r a resolution parameter that has been fixed to 2 for all the experimentations. The community score is defined as

$$\mathcal{CS} = \sum_{i}^{k} score(C_i) \tag{3}$$

where

$$score(C) = \frac{\sum_{i \in C} (\frac{1}{|C|} \sum_{j \in C} M_{ij})^r}{|C|} \times \sum_{i,j \in C} M_{ij} \tag{4}$$

It measures the fraction of internal edges of each community with respect to its size.

Conductance: let C be a cluster with m_c edges, and $b_c = \{(i,j) \mid i \in C, j \notin C\}$ be the number of edges on the boundary of C, the conductance is defined as

$$CO = \sum_{c=1}^{k} \frac{b_c}{2m_c + b_c} \tag{5}$$

It measures the fraction of edges starting from a community and pointing outside it.

Attribute homogeneity. Let A_{v_i} and A_{v_j} be the attribute vectors of nodes v_i and v_j. The quality of a clustering with respect to attribute homogeneity can be measured as follows.
Similarity based on Jaccard index:

$$sim_{JI} = \frac{1}{k} \sum_{C \in \mathcal{C}} \sum_{\substack{v_i, v_j \in C \\ v_i \neq v_j}} \frac{\left| A_{v_i} \cap A_{v_j} \right|}{\left| A_{v_i} \cup A_{v_j} \right|} \quad (6)$$

It measures the average fraction of common attributes within communities. This objective function is suitable for discrete attributes assuming a finite set of values (e.g., profession, zip code, etc.).
Cosine-based similarity:

$$sim_{COS} = \frac{1}{k} \sum_{C \in \mathcal{C}} \sum_{\substack{v_i, v_j \in C \\ v_i \neq v_j}} \frac{A_{v_i} \cdot A_{v_j}}{\|A_{v_i}\| \left\|A_{v_j}\right\|} \quad (7)$$

It measures the average fraction of similar attributes in terms of the cosine angle between attribute vectors. Two vectors with the same orientation have a similarity of 1, two vectors at 90° have a similarity of 0, and two diametrically opposed vectors have a similarity of -1, independently of their magnitude. This objective function is suitable for textual attributes as documents represented as term frequencies vectors (in this case the cosine similarity is bounded in [0,1]) and for binary attributes.
Similarity based on Euclidean distance:

$$sim_{ED} = -\frac{1}{k} \sum_{C \in \mathcal{C}} \sum_{\substack{v_i, v_j \in C \\ v_i \neq v_j}} \sqrt{\sum_{\alpha \in A} (a_\alpha(v_i) - a_\alpha(v_j))^2} \quad (8)$$

$a_\alpha(v_i)$ and $a_\alpha(v_j)$ are the values of attribute α for node v_i and v_j respectively. It measures the average distance between attributes within communities. This objective function is suitable for continuous attributes having real numbers as values (e.g., temperature, height, weight, etc.).

We highlight that, in the implementation, instead of maximizing the similarity, we minimize the complementary concept of distance.

B. Representation and genetic operators

The method uses the locus-based adjacency representation [36] where an individual of the population consists of n genes, with n the number of nodes. Each gene represents a node and can assume a value j in the range $\{1, \ldots, n\}$. A value j assigned to the i-th gene means that there is a link between the nodes i and j of V, thus i and j are in the same cluster. A decoding step identifies all the communities of the network. Uniform crossover is adopted to generate the offspring. Given two parents, a random binary vector is generated. The child is obtained by combining the genes of the parents by taking at position i the value j coming from the first parent, when the vector value at position i is a 0, and from the second parent when the vector value is 1. The mutation operator for each node i randomly changes the gene value with one of the neighbors of i.

C. Algorithm description

A detailed description of the method is reported in Fig. 2. The method receives in input the two objectives f_S and f_A to optimize and the maximum number of generations. At the first step it initializes the population by assigning to each node one of its neighbors at random. Until the termination condition is not satisfied, i.e either a maximum number of generations has been reached or the objective function does not improve anymore, each individual of the population is decoded to obtain a partitioning, and the two objectives are evaluated (steps 2-5). A rank is then assigned to solutions based on Pareto dominance (step 6), and a new population is created by applying the genetic operators to the best selected points from the combined parent and offspring populations (steps 8-9). At the end of the computation the method returns the set of Pareto-optimal solutions. The solution $\mathcal{P} = \{P_1, \ldots, P_l\}$ having the highest value of the objective f_S is chosen. As last step the *LocalMerge* procedure is executed to produce the final solution $\mathcal{C} = \{C_1, \ldots, C_k\}$. The pseudo-code of this method is described in Fig. 3.

The MOGA-@Net Method:
Input: An attributed graph $G = (V, E, A, F)$, structure fitness function f_S, attribute fitness function f_A, maximum number of generations T
Output: A partitioning $\mathcal{C} = \{C_1, \ldots, C_k\}$ of the nodes of G in communities

1 **Initialize** a population of random individuals by assigning to each node one of its neighbors
2 **while** termination condition is not satisfied do
3 **for each** individual $I = \{g_1, \ldots, g_n\}$ of the population
4 **Decode** I to generate a partitioning
5 **evaluate** the two objectives f_S and f_A
6 **Assign** a rank based on Pareto dominance
7 **end for each**
8 **Combine** parents and offspring and partition into fronts;
9 **Select** the best points, and apply the variation operators to create the next population;
10 **end while**
11 **choose** the solution $\mathcal{P} = \{P_1, \ldots, P_l\}$ from the Pareto front having the best value of f_S
12 **Perform** local merge on $\mathcal{P}$ and
13 **Return** the merged solution $\mathcal{C} = \{C_1, \ldots, C_k\}$

Fig. 2. The pseudo-code of the MOGA-@Net algorithm.

The $LocalMerge$ Method:
Input: A clustering $\mathcal{P} = \{P_1, \ldots, P_l\}$ of the nodes of the attributed graph $G = (V, E, A, F)$ in communities
Output: A clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ merging communities of $\mathcal{P}$

1 **Let** L be a vector of size n such that $L(i) = j$ if node $v_i \in P_j$ and CM the confusion matrix of size $l \times l$ such that $CM(i, j)$ is the number of edges between the communities P_i and P_j
2 **for** i=1, l do
3 **if** (i has not already been included into another community)
4 **let** $\overline{j}$ the column index such that $CM(i, \overline{j}) \geq CM(i, j), j \in 1, \ldots, n, j \neq \overline{j}$
5 **let** $\overline{P} = P_i$ if $\mid P_i \mid \leq \mid P_{\overline{j}} \mid$ $\overline{P} = P_{\overline{j}}$ otherwise
6 **if** ($m_{\overline{P}} \leq CM(i, \overline{j})$
7 $L(\overline{P}) = \overline{j}$
8 **end for**
9 $k = max(L)$
10 **return** the merged solution $\mathcal{C} = \{C_1, \ldots, C_k\}$

Fig. 3. The pseudo-code of the $LocalMerge$ algorithm.

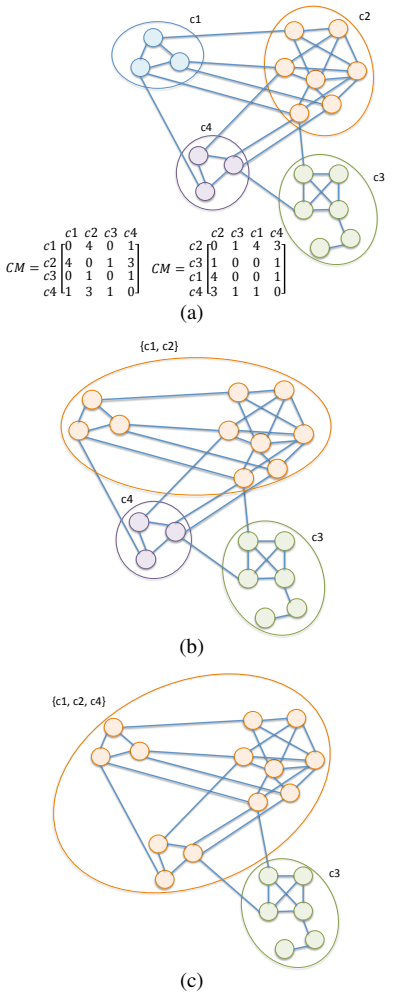


Fig. 4. Example of the $LocalMerge$ procedure with two confusion matrices to show that the community ordering does not influence the result.

The $LocalMerge$ method uses a vector L of size n, and the confusion matrix CM of size $l \times l$ containing in the position $CM(i,j)$ the number of connections between the communities P_i and P_j (step 1). It then examines each row of CM to check if a community is a candidate to be included into another one. To this end, it finds the community having the maximum number of connections with P_i and considers that having the smallest size, denoted with $\overline{P}$ (steps 4-5). If the number $m_{\overline{P}}$ of internal connections of the community $\overline{P}$ is less than the number $CM(i,\overline{j})$ of links with the other community, then the two communities P_i and $P_{\overline{j}}$ are merged (steps 6,7). The number k of communities after merging and the new community structure are obtained by the label vector L (steps 9,10). It is worth noting that the merge procedure does not depend on the order in which the communities are analyzed. In fact, each community P_i is compared always with the community P_j of the solution $\mathcal{P} = \{P_1, \ldots, P_l\}$ with which it has the maximum number of connections, independently if P_j has already been examined or merged into another community. It is the vector L that relabel the cluster node to avoid inconsistencies.

Fig. 4 shows an example of execution of the method. Consider the confusion matrix on the left and the first community C_1 in Fig. 4(a), thus $i = 1$. C_1 has 4 links with community C_2, thus $\overline{j} = 2$ and $\overline{P} = C_1$, C_1 being the smallest between C_1 and C_2. Since $3 = m_{\overline{P}} < CM(1,2) = 4$, C_1 is included in C_2 and its nodes take the label of C_2 (Fig. 4(b)). For $i = 2$ and $i = 3$ the communities C_2 and C_3 have a number of internal links higher than the inter-layer links, thus they do not change. For $i = 4$, instead, $\overline{P} = C_4$ also satisfies $3 = m_{\overline{P}} \leq CM(1,2) = 3$, thus C_4 is merged with C_2. The final division is constituted by two clusters, one including $\{C_1, C_2, C_4\}$ and the other C_3 (Fig. 4(c)). As already pointed out, the order in which the communities are examined does not influence the final result. In fact, if we consider the confusion matrix on the right in Fig. 4(a), the final partitioning is the same.

V. Evaluation Metrics

To assess the quality of the solutions obtained by the algorithm, when the ground-truth division of a network is known, we use the popular *Normalized Mutual Information* measure [42] and a variant introduced in [24] that takes into account also node attributes. Otherwise, the internal indexes of $density$ and $entropy$ are employed.

Normalized Mutual Information (NMI). The normalized mutual information $NMI(A,B)$ of two divisions A and B of a network is defined as follows. Let C be the confusion matrix whose element C_{ij} is the number of nodes of community i of the partition A that are also in the community j of the partition B.

$$NMI(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} C_{ij} log(C_{ij} n / C_{i.} C_{.j})}{\sum_{i=1}^{c_A} C_{i.} log(C_{i.}/n) + \sum_{j=1}^{c_B} C_{.j} log(C_{.j}/n)} \quad (9)$$

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

Cumulative NMI (CNMI). CNMI [24] is a modified NMI measure allowing the integration of NMI values over different settings of structure mixing parameter (μ) and attribute noise (ν):

$$CNMI = \frac{\sum^{\mu}\sum^{\nu} NMI}{S} \quad (10)$$

where S is the number of samples of the considered network graphs.

Density. It is defined as

$$D = \sum_{C \in \mathcal{C}} \frac{m_c}{m} \quad (11)$$

where m_c is the number of edges of the community C and m is the total number of edges of the network. It measures the internal edge density of a partitioning.

Entropy. It is based on the information theory concept of entropy, and it measures the average Shannon information content of a set. A highly disordered set with different elements has a high entropy. Thus, the lower the entropy, the more homogeneous the attribute values. Entropy is defined as

$$E = -\sum_{C \in \mathcal{C}} \frac{n_c}{n} \sum_{a \in A} p_{ac} \log(p_{ac}) \quad (12)$$

where p_{ac} is the percentage of nodes in community C with the attribute value a, n_c is the number of nodes on the community C and n is the number of vertices of the network.

VI. Experimental Evaluation

We test the effectiveness of our approach on two classes of datasets: synthetic and real world datasets. The former are computer generated networks allowing the creation of the ground-truth useful to assess the similarity between the synthetically generated and the detected communities. The latter datasets, extracted from real environments, better represent the real network behavior. For some of them, the partitioning in communities is not known. We compare MOGA-@Net with eight state-of-the-art methods on the synthetic networks by exploiting the experimentation performed by Elhadi and Agam [24], while on the real world datasets, a comparison with the evolutionary multiobjective method *MOEA-SA* of Li *et al.* [10], which is the most similar to our approach, is carried out by using the results reported by the authors. Notice that *MOEA-SA* has not been tested on synthetic networks. The MOGA-@Net algorithm has been written in MATLAB2015b, by using the Global Optimization Toolbox, which implements the NSGA-II framework of Deb *et al.* [43]. The following sections describe in detail the datasets, the algorithms used for comparison, and the results obtained by the simulations.

A. Synthetic Networks

The synthetic networks have been generated by using the benchmark proposed by Elhadi and Agam [24], named LFR-EA, which is an extension of the LFR benchmark of Lancichinetti et al. [44]. The generator uses two parameters μ and ν, both ranging in the interval [0.1, 0.9], to control the structure and the attribute values, respectively. μ is called *mixing parameter* and determines the rate of intra- and inter-communitiy connections. Low values of μ give a clear community structure where intra-cluster link are much more than inter-cluster links. Analogously for ν, called *attribute noise*, low values generate similar features of nodes belonging to the same community. Besides ν, the number of attributes and the size of the domain D_α of each attribute α must be specified. The combination of μ and ν values produces graphs with a clear to ambiguous structure and/or attributes.

We generated a benchmark of networks consisting of 1000 nodes, in the following named LFR-EA-1000. The parameters used to generate it are the same of those employed in [24] and shown in Table I. All the nodes in a community share the same attribute domain values. Specifically, the nodes are labeled with two attributes that assume numerical values. Finally, the *attribute's domain cluster assignment* is set to random selection without replacing, in order to cover all the domain values across the different communities. We generated ten different instances of the combination of μ and ν parameters reported in Table I.

TABLE I
LFR-EA-1000 PARAMETERS SETTING.

Parameter	Value
Number of nodes (N)	1000
Average degree (k)	25
Maximum degree ($maxk$)	40
Exponent for the degree distribution ($t1$)	2
Mixing parameter (μ)	[0.1; 0.9]
Exponent for the community size distribution ($t2$)	1
Minimum for the community sizes ($minc$)	60
Maximum for the community sizes ($maxc$)	100
Number of overlapping nodes (on)	0
Number of memberships of the overlapping nodes (om)	0
Number of attributes ($nattr$)	2
Attribute's domain cluster assignment ($ainf$)	1
Attribute # 1 domain size	3
Attribute # 1 noise (ν)	[0; 0.9]
Attribute # 2 domain size	15
Attribute # 2 noise (ν)	[0; 0.9]

1) Comparison with existing algorithms: to assess the quality of the results obtained by MOGA-@Net, we compare it with eight different algorithms, each representative of a type of approach: (1) *structure-only*, (2) *attribute-only*, (3) *composite*, (4) *selection* and (5) *ensemble*. These methods have been used by Elhadi and Agam [24] to evaluate their $Selection$ algorithm. The structure-only (*Louvain* [3]) and the attribute-only (*k-means* [45]) focus on just one of the two aspects of the attributed graphs, i.e. the links and the attributes of the nodes, respectively. Composite algorithms, on the contrary, consider both the structure of the graphs and their attributes (*SA-cluster* [22], *BAGC* [9], *Entropy based* [7]). The *Selection* approach [24], fixed a structure-only method and an attribute-only method, opportunistically switchs between the two methods to manage the graph structure ambiguity. These latter methods have been described in Section III. Finally, ensemble methods make use of a structure-only method and an attribute-only method, and then use cluster ensemble techniques to merge the results of the two classes of algorithms. Elhadi and Agam proposed to combine the results of the *Louvain* and *k-means* methods inside the two cluster ensemble methods $HGPA$ and $CSPA$ of Strehl and Ghosh [46]. It is worth pointing out that the results of these methods are those reported in [24].

2) Results: **Parameter setting**. As first experiment, we analyzed the behavior of MOGA-@Net for different genetic parameter values with the aim to find the best setting giving good results for the benchmark datasets. To this end, we fixed the attribute noise to ν=0.5, in order to have an attributed graph with attributes that are sufficiently ambiguous, and then varying the mixing parameter μ in the interval [0.1, 0.5, 0.9] to have clear, less clear, and mixed community structure, respectively. For this experiment, we used modularity as fitness function for the intra-community link optimization, and the similarity based on the Euclidean distance as intra-community attributes' homogeneity fitness function, being the attributes numerical, and computed the NMI values obtained by the method when varying the population size, the crossover fraction and the mutation rate. Fig. 5(a) shows the NMI values as a function of the crossover fraction for different mutation rates when $\mu = 0.1$ and the population size 100. In this case, the structure of the attributed graph is clear: it is well structured

in communities with few inter-communities edges and many intra-community edges. In such situation, a high crossover fraction (0.8) with a low mutation rate (0.2) gives the highest NMI. For $\mu = 0.5$ (Fig. 5(b)), the structure of the attributed graph is more ambiguous. Again the crossover fraction 0.8 gives the highest NMI value but combined with a mutation rate of 0.4. As the graph structure becomes totally ambiguous, as in the case of $\mu = 0.9$ (Fig. 5(c)), the highest NMI is given by a lower crossover fraction of 0.4 with a mutation rate of 1. However, we observe that for a crossover fraction of 0.8, the NMI remains high also for lower mutation rates (0.6, 0.4 and 0.8). Thus, we conclude that for MOGA-@Net a good combination of the genetic parameters is crossover fraction 0.8 and mutation rate 0.4, since they result in high NMI values over all the mixing parameters considered.

Table II shows the NMI values, with the standard deviation in parenthesis, for different population sizes and mixing parameters with crossover fraction 0.8 and mutation rate 0.4. The simulations show that the best setting for the population is 300 individuals since it results in the highest NMI values.

TABLE II
MOGA-@NET NMI VALUES FOR DIFFERENT POPULATION SIZES WITH CROSSOVER FRACTION 0.8 AND MUTATION RATE 0.4 ON THE LFR-EA-1000 DATASET.

Population size	μ	NMI
100	0.1	0.9 (0.073)
	0.5	0.962 (0.058)
	0.9	0.934 (0.072)
300	0.1	0.943 (0.061)
	0.5	0.963 (0.042)
	0.9	0.924 (0.069)
500	0.1	0.934 (0.062)
	0.5	0.943 (0.046)
	0.9	0.933 (0.046)

LFR-EA-1000 network. Fixed the genetic parameters, we compared MOGA-@Net with state-of-the art algorithms by executing the method for all the combinations of μ and ν values, by using the three fitness functions of modularity, community score, and conductance. Since the attributes are numerical, we considered as second objective the similarity based on the Euclidean distance sim_{ED} (formula (8)). Table III shows the values of $CNMI$ obtained by MOGA-@Net and the other algorithms considered for the comparison. MOGA-@Net, finding on average 13 communities, outperforms all the other algorithms. The modularity function, in particular, achieves the highest $CNMI$ value, showing the effectiveness of using a multiobjective genetic algorithm for exploiting both graph structure and attributes in a composite way. The other algorithms of the same class, such as *BAGC* and *EntropyBased*, achieve $CNMI$ values comparable to the structure-only *Louvain*. *SA-Cluster* performs the worse among the composite methods, showing to not be able to correctly identify communities. The ensemble methods $HGPA$ and $CSPA$, combining the *Louvain* and the *K-means* methods, obtain medium-low $CNMI$ values. This is due to the fact that the low NMI values of the *K-means* negatively influence the *Louvain* results, inducing low $CNMI$ values. The *Selection* method shows a good $CNMI$ value compared to the other methods. When the structure of the graph becomes less clear (high μ), it is able to properly find the boundary between clear and ambiguous graph structure content, opportunistically exploiting the attribute-based clustering through the *K-means*. In fact, the *Louvain* method is able to achieve very high NMI values for high and medium mixing parameters, independently from the attribute noise since it works only on the graph structure, the *K-means*, independently from the mixing parameter, obtains high NMI values only for low attribute noise. Thus the *Selection* method performs the best among the considered approaches, though MOGA-@Net outperforms it.

TABLE III
COMPARISON OF CUMULATIVE NMI BETWEEN MOGA-@NET AND THE OTHER STATE-OF-THE ART ALGORITHMS ON THE LFR-EA-1000 DATASET. THE SECOND OBJECTIVE IS THE EUCLIDEAN DISTANCE.

Method	Type	CNMI
MOGA-@Net (f_S=modularity)	Composite	0.878 (0.071)
MOGA-@Net (f_S=community score)	Composite	0.8717 (0.07)
MOGA-@Net (f_S=conductance)	Composite	0.863 (0.061)
Louvain [3]	Structure-only	0.699
K-means [45]	Attributes-only	0.354
BAGC [9]	Composite	0.613
EntropyBased [7]	Composite	0.696
SA-Cluster [22]	Composite	0.193
Selection [24]	Switching	0.776
HGPA [46]	Ensemble	0.454
CSPA [46]	Ensemble	0.482

B. Real world Networks

We now test the performance of MOGA-@Net over a set of real world attributed graphs. Specifically, we focus on networks having the ground-truth (*Cora* and *Citeseer*)[1], and on networks that do not have the ground-truth. Two networks, *Amazon US Politics Books* [2] and *Political Blogs* [47] have a single attribute, while the *Facebook Ego Networks* [48], have multiple attributes. Table IV summarizes their features. MOGA-@Net has been executed 10 times and the average results are reported. In the following, we briefly introduce the datasets and detail the results of the experiments we performed.

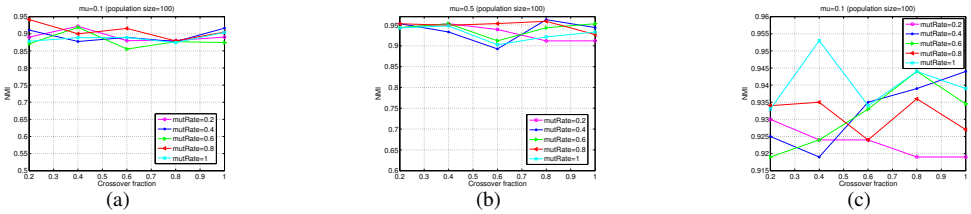
1) Datasets: **Cora** contains a set of nodes representing scientific publications, where an edge between two nodes is a citation from a publication to another. The dictionary, consisting of a set of unique words, represents the attributes' domain of this network. If a word is present in the paper, the attribute for that word is set to 1, 0 otherwise. Each publication has been classified into seven classes: 1) neural networks, 2) rule learning, 3) reinforcement learning, 4) probabilistic methods, 5) theory, 6) genetic algorithms, and 7) case-based reasoning.

Citeseer is another dataset of publication citations where each node belongs to one of the following six categories: 1) agents, 2) information retrieval (IR), 3) databases (DB), 4) artificial intelligence (AI), 5) human-computer interaction (HCI), and 6) machine-learning (ML).

Amazon US Politics Books is composed by US politics books sold by Amazon.com during the presidential elections

[1]Both datasets are available at https://linqs.soe.ucsc.edu/

[2]Available at http://www-personal.umich.edu/ mejn/netdata/



(a) (b) (c)

Fig. 5. NMI values for different crossover fractions and mutation rates, by setting (a) $\mu = 0.1$, (b) $\mu = 0.5$, (c) $\mu = 0.9$, and the population size to 100.

of 2004. The edges between books represent co-purchasing relationships. Each book has an attribute indicating the political position: 1) conservative, 2) liberal, or 3) neutral.

Political Blogs contains hyperlinks between political weblogs in the US. Each weblog has an attribute with two values: conservative or liberal.

Facebook Ego Networks is a collection of 10 ego networks including 4039 Facebook users and 193 circles. Each user, directly connected to his/her friends connected to other friends, has several attributes that have been anonymized through one-hot encoding. For our analysis we consider the 3 ego networks of the dataset named 686, 3437 and 3980 having nodes distributed in 14, 32 and 17 circles, respectively.

2) Results: **Citation networks with ground-truth**. Table V shows the comparison between MOGA-@Net, *MOEA-SA*, $Louvain$ and *K-means* on the Cora and Citeseer datasets in terms of NMI. For both datasets, having binary attributes, we considered the cosine distance as second fitness function. It can be seen that MOGA-@Net is able to perfectly match the ground-truth both in Cora and in Citeseer. It is worth pointing out that the choice of the first fitness function does not influence the output of the algorithm. We highlight that MOGA-@Net is able to find the right number of communities in a situation in which the number of attributes is very high. *MOEA-SA*, on the contrary, achieves NMI values of 0.46 and 0.35. $Louvain$ obtains 38 communities and an NMI value of 0.603 on the Cora dataset, and 72 communities and $NMI = 0.534$ on the Citeseer dataset. Thus this method, with the topological information alone, divides the network in many small communities. The *K-means* method, instead, even though it receives as input the correct number of groups, achieves NMI values rather low, 0.289 and 0.327, respectively, showing that the compositional variables alone are not sufficient to find a good clustering. These results confirm the importance of considering both structural and attribute components to obtain high quality partitions.

Single-attributed political networks with no ground-truth. The Amazon US Politics Books and the Political Blogs networks, indicated as 'Polbooks' and 'Polblogs', have a single attribute denoting the political leaning, often used as ground-truth division. In such a case, since we use this information as attribute, the two indexes of density and entropy are computed, by considering the Euclidean distance as second objective to minimize. As already outlined, high values of density represent communities well-separated in terms of structure, while low entropy values indicate homogeneous communities from the attributes perspective. From Table VI we can observe that for the Polbooks dataset, MOGA-@Net average values of density D, when using modularity and community score, are lower than *MOEA-SA*, though MOGA-@Net achieves the maximum value of density of 0.9751 for conductance. Regarding the entropy E, we find that MOGA-@Net always outperforms *MOEA-SA*. Considering the number of communities, MOGA-@Net finds a number of communities between 3, which corresponds to the nodes's division in conservative, liberal and neutral, and 9, while *MOEA-SA* finds 5 communities. In Polblogs, independently from the intra-community link optimization function, MOGA-@Net always results in an entropy value of 0, while *MOEA-SA* achieves an average value of 0.0813. Moreover, MOGA-@Net finds the solution having the maximum density with 2 communities, that effectively corresponds to the bipartition of weblogs in conservative and liberal. *MOEA-SA*, on the contrary, with a number of communities ranging between 3 and 11, results in an average density value of 0.9062.

TABLE IV
FEATURES OF THE REAL WORLD DATASETS.

Dataset	Graph Type	Nodes	Edges	Attributes
Cora	Citation	2708	5429	1433
Citeseer	Citation	1787	3285	3703
Polbooks	Books co-purchasing	105	441	1
Polblogs	Blogs hyperlinks	1490	19090	1
Ego 686	Friendship	170	1656	63
Ego 3437	Friendship	542	4749	262
Ego 3980	Friendship	58	143	42

Multi-attributed Facebook networks with no ground-truth. The results obtained for the Facebook ego networks with multi-attributes are shown in Table VII. Since the attributes are binary, we minimize the cosine distance as fitness function for the attribute similarity. In Ego 686 and Ego 3437, the MOGA-@Net results in terms of average density and average entropy are similar for all the objective functions. MOEA-SA performs worst on Ego 686, but achieves higher values of density on the Ego 3437 and Ego 3980, though higher values of entropy on Ego 3437. Figure 6 shows the circles with the corresponding number of nodes they contain. It is worth pointing out that MOGA-@Net finds a number of communities that reflects the distribution of nodes within these circles. For instance, the number of circles with size greater than one in Ego 3437 is 22, the same average number of communities obtained by MOGA-@Net. All these results demonstrate the very good performance of MOGA-@Net.

VII. COMPARISON BETWEEN NSGA-II AND MOEA/D

Multiobjective evolutionary algorithms have been shown to be effective methods in solving multiobjective problems be-

TABLE V
COMPARISON OF NMI BETWEEN MOGA-@NET, *MOEA-SA*, $Louvain$ AND *K-means* ON THE CORA AND CITESEER CITATION NETWORKS.

Method	Cora NMI	Citeseer NMI
MOGA-@Net(f_S=modularity)	1	1
MOGA-@Net(f_S=community score)	1	1
MOGA-@Net(f_S=conductance)	1	1
MOEA-SA	0.46 (0.001)	0.35 (0.004)
$Louvain$	0.603	0.534
K-means	0.289	0.327

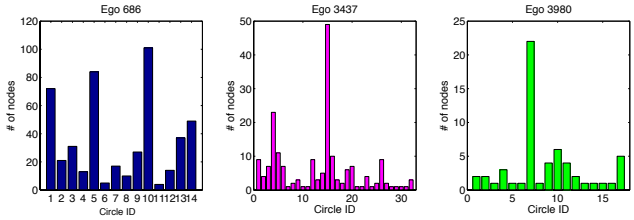


Fig. 6. Distributions of nodes within circles in Facebook Ego Networks.

cause of their population based nature which allows to obtain multiple Pareto optimal solutions in a single run [49]. There are two main frameworks to find the Pareto optimal solutions: the *dominated-based* framework and the *decomposition-based* framework. In the dominated-based framework a multiobjective optimization problem is optimized by simultaneously optimizing all the objectives relying on the Pareto-dominance principle [13]. The most famous multiobjective framework of this family is the *Nondominated Sorting Genetic Algorithm (NSGA-II)* proposed by Deb *et al.* in [43]. NSGA-II builds a population of competing individuals and ranks them on the basis of nondominance. In the decomposition-based framework a multiojective optimization problem is decomposed into a number of scalar optimization subproblems by using decomposition methods. The single-objective subproblems are then simultaneously solved by evolving a population of solutions. *MOEA/D* is the most famous representative of this category [50].

The experimentation presented in the previous section, as outlined, used the Global Optimization Toolbox of Matlab, which implements the NSGA-II framework of Deb *et al.* [43]. In this section we compare these results with those obtained by using the MOEA/D framework instead of the NSGA-II framework only when using as first objective function the modularity. The results for the other objectives are similar.

Figure 7(a) and Figure 7(b) show the Pareto Front obtained by using NSGA-II and MOEA/D on the synthetic network with mixing parameter $\mu = 0.5$ and attribute noise $\nu = 0.5$, before the execution of $LocalMerge$ method and after, respectively. The figures point out that the solutions obtained with MOEA/D have lower modularity and higher distance. Moreover, the NMI value is at most 0.42, while with NSGA-II the highest value is 0.984, corresponding to the solution with highest modularity. The behavior on the LFR networks with different values of μ and ν are similar, thus we do not report them for space problems.

Figure 8 shows the same information for the $Cora$ network. In such a case the modularity value of the solutions obtained with NSGA-II before the merge is much higher than those found with MOEA/D, though the cosine distance among the attributes is lower for the latter. However, after the merge, the NSGA-II solutions all converge to a unique solution which coincides with the ground-truth. The $LocalMerge$ procedure improves also the solutions obtained with MOEA/D, which reduces to 4, one of which is also that corresponding to the ground-truth.

Regarding the $Citeseer$ network, MOGA-@Net obtains a Pareto Front with only one solution, which is the ground-truth solution, independently of the multiobjective framework.

Figure 9 shows the Pareto Fronts for the other real world networks, again before and after the local merge. In these cases, the ground-truth is known for the Polbooks and Polblogs networks, however the class label has been used as the unique attribute characterizing these two networks, while for the other networks the true division is not known. Thus, for these real-world networks the density and entropy values, the former considers the internal density of the partitions, the latter the attribute homogeneity, are reported. From the results it can be observed that on these networks the behavior of the two MOEA frameworks seems similar. With NSGA-II, in general, the size of the Pareto Front is lower, the solutions have higher modularity, lower distance, higher density, except for Polbooks and Ego 3437, and lower entropy for Polbooks and Ego 686. In order to better understand the behavior of the two frameworks, we analyzed more in detail the Polbooks and Polblogs networks. As outlined, for these two test problems the true division is known. However, even if the class label has been considered an attribute, we computed the NMI of the solutions obtained by the two frameworks, with the highest value of modularity, by considering the ground-truth division determined by the class label. We obtained for Polbooks NMI=0.841 and NMI=0.627 for the NSGA-II and MOEA/D frameworks, respectively, while for Polblogs NMI=1 and NMI=0.25, respectively. Thus, again, when comparing the two frameworks with respect to the ground-truth division NSGA-II outperforms MOEA/D. From the results, it is clear that the two measures of density and entropy are not able to clearly discriminate between the two frameworks, thus it is rather difficult to state the superiority of an approach with respect to the other by taking into account only these two measures. However, we can conclude that the NSGA-II framework is more appropriate than MOEA/D for finding the true network divisions.

VIII. CONCLUSION

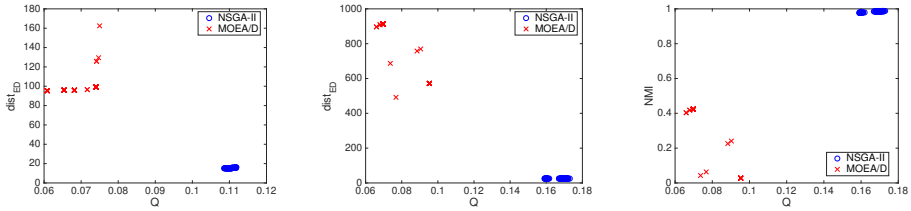
The paper proposed a multiobjective genetic algorithm for the community detection problem that integrates the structural and compositional dimensions contained in attributed networks. The multiobjective framework allows to balance the information contained into the actors composing the network and their topological connections to obtain divisions that are both well connected and with similar nodes. An extensive experimentation on synthetic and real world networks showed the very good performance of our approach when compared with state-of-the-art methods. MOGA-@Net is often capable

TABLE VI
COMPARISON OF DENSITY AND ENTROPY BETWEEN MOGA-@NET AND MOEA-SA ON AMAZON US POLITICS BOOKS AND POLITICAL BLOGS NETWORKS.

Dataset	Methods	D_{max}	D_{min}	D_{avg}	E_{max}	E_{min}	E_{avg}	n_C
Polbooks	MOGA-@Net (f1=modularity)	0.9433	0.7485	0.841 (0.0571)	0.161	0.1349	0.1518 (0.0105)	4-7
	MOGA-@Net (f1=community score)	0.8435	0.678	0.7843 (0.0507)	0.183	0.1589	0.1667 (0.0091)	5-9
	MOGA-@Net (f1=conductance)	0.9751	0.8072	0.8811 (0.0813)	0.1807	0.1663	0.1731 (0.0057)	3-7
	MOEA-SA	0.8934	0.8005	0.8463 (0.0253)	0.4888	0	0.2304 (0.1278)	5
Polblogs	MOGA-@Net (f1=modularity)	0.926	0.9229	0.9252 (0.1213)	0	0	0	5-6
	MOGA-@Net(f1=community score)	0.9258	0.9246	0.9256 (0.0003)	0	0	0	5-6
	MOGA-@Net (f1=conductance)	1	0.9256	0.9332 (0.023)	0	0	0	2
	MOEA-SA	0.9134	0.89	0.9062 (0.0059)	0.1827	0	0.0813 (0.0564)	3-11

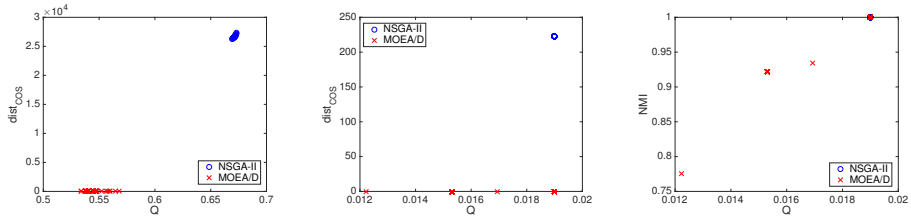
TABLE VII
COMPARISON OF DENSITY AND ENTROPY BETWEEN MOGA-@NET AND MOEA-SA ON FACEBOOK EGO NETWORKS.

Dataset	Methods	D_{max}	D_{min}	D_{avg}	E_{max}	E_{min}	E_{avg}	n_C
Ego 686	MOGA-@Net (f1=modularity)	0.9885	0.9408	0.9634 (0.0145)	0.0839	0.0635	0.0753 (0.0058)	2-6
	MOGA-@Net (f1=community score)	0.9758	0.9541	0.9637 (0.0064)	0.08	0.0699	0.0747 (0.0038)	4-5
	MOGA-@Net (f1=conductance)	0.9849	0.9504	0.962 (0.0102)	0.0909	0.07	0.0753(0.007)	3-4
	MOEA-SA	0.7101	0.5954	0.666 (0.0262)	0.2946	0.2737	0.2811 (0.0059)	-
Ego 3437	MOGA-@Net (f1=modularity)	0.8653	0.8015	0.8344 (0.0237)	0.1079	0.0997	0.1042 (0.0273)	20-23
	MOGA-@Net(f1=community score)	0.8499	0.8044	0.8305 (0.0151)	0.1079	0.0978	0.1034 (0.0031)	20-23
	MOGA-@Net (f1=conductance)	0.8532	0.814	0.8352 (0.0098)	0.1086	0.0998	0.1023 (0.0026)	20-22
	MOEA-SA	0.9522	0.6641	0.8417 (0.0989)	0.1089	0.1003	0.1042 (0.0029)	-
Ego 3980	MOGA-@Net (f1=modularity)	0.7756	0.5289	0.6565 (0.1003)	0.3252	0.3018	0.314 (0.0079)	4-8
	MOGA-@Net (f1=community score)	0.7101	0.4782	0.592 (0.071)	0.3275	0.2976	0.3144 (0.0081)	5-9
	MOGA-@Net(f1=conductance)	0.7028	0.4624	0.5442 (0.028)	0.3591	0.2699	0.3142 (0.031)	2-7
	MOEA-SA	0.7552	0.6294	0.6921 (0.0362)	0.2991	0.2719	0.2887 (0.0061)	-



(a) (b) (c)

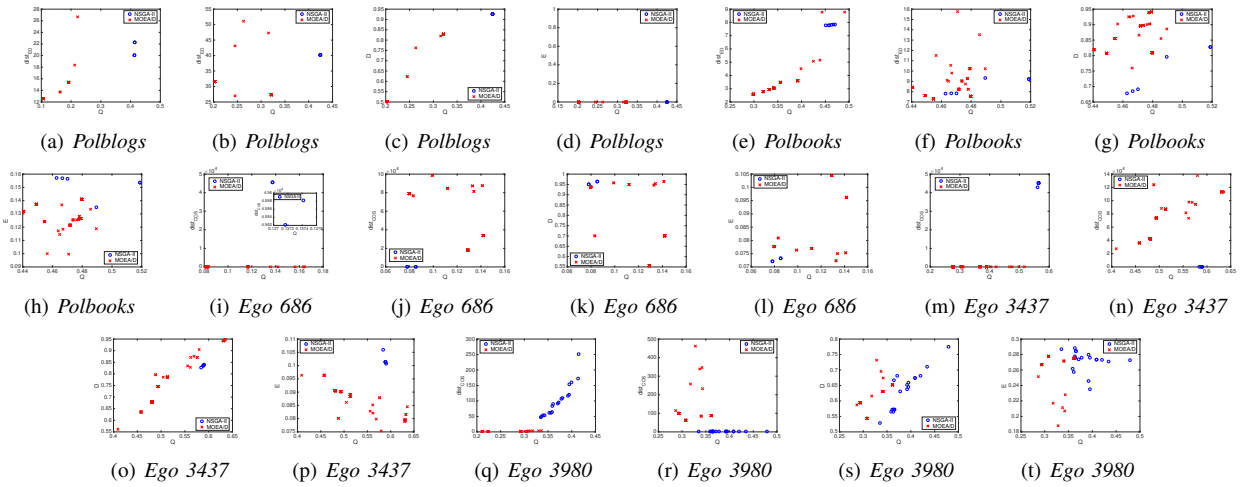
Fig. 7. Comparison of the Pareto Fronts obtained with NSGA-II and MOEA/D on the synthetic dataset with parameters $\mu = 0.5$ and $\nu = 0.5$. (a) Pareto Fronts returned by the method under the NSGA-II (blue circle) and the MOEA/D (red cross symbol) frameworks. (b) Pareto Fronts after applying the Local Merge procedure to each Pareto Front Solution. (c) NMI values of each solution.



(a) (b) (c)

Fig. 8. Comparison of the Pareto Fronts obtained with NSGA-II and MOEA/D on the Cora network. (a) Pareto Fronts returned by the method under the NSGA-II (blue circle) and the MOEA/D (red cross symbol) frameworks. (b) Pareto Fronts after applying the Local Merge procedure to each Pareto Front Solution. (c) NMI values of each solution.

to obtain the ground-truth division, like for instance, the Cora and Citeseer real world networks, or a partitioning very close to the real division. Moreover, the local search procedure properly reduces the number of solutions of the Pareto Front, and sensibly improves the quality of the communities. The method has been experimented with two MOEAs frameworks. The results show that MOGA-@Net performs well for both the NSGA-II and MOEA/D multiobjective evolutionary frameworks, even if NSGA-II seems more suitable for this kind of problem. It is worth pointing out that several studies, mainly on many-objective optimization [51], [52], [53], comparing multiobjective frameworks on problems with different characteristics, have observed there is not a MOEA method which outperforms all the others in all the types of problems. For instance, Li et al. [51], have outlined that "none of the approaches has a clear advantage over the others, although some of them are competitive on most of the problems". Analogous considerations are done in [52]. Understanding the behavior and mechanisms of MOEAs on the different kinds of domains is still an open problem, as outlined in [54].



(a) *Polblogs* (b) *Polblogs* (c) *Polblogs* (d) *Polblogs* (e) *Polbooks* (f) *Polbooks* (g) *Polbooks*

(h) *Polbooks* (i) *Ego 686* (j) *Ego 686* (k) *Ego 686* (l) *Ego 686* (m) *Ego 3437* (n) *Ego 3437*

(o) *Ego 3437* (p) *Ego 3437* (q) *Ego 3980* (r) *Ego 3980* (s) *Ego 3980* (t) *Ego 3980*

Fig. 9. Comparison of the Pareto Fronts obtained with NSGA-II and MOEA/D: (a)-(d) Polblogs, (e)-(h) Polbooks, (i)-(l) Ego 686, (m)-(p) Ego 3437, (q)-(t) Ego 3980. (a), (e), (i), (m), (q): Pareto Fronts returned by the method under the NSGA-II (blue circle) and the MOEA/D (red cross symbol) frameworks. (b), (f), (j), (n), (r): Pareto Fronts after applying the Local Merge procedure to each Pareto Front Solution. (c), (g), (k), (o), (s) Density values of each solution. (d), (h), (l), (p), (t): Entropy values of each solution.

Recently, Li *et al.* [10] proposed the multiobjective method *MOEA-SA* for attributed networks. However, the differences between MOGA-@Net and *MOEA-SA* are numerous and noteworthy. First of all, *MOEA-SA* does not consider continuous attribute values and thus defines a similarity function between two feature vectors based on the cosine similarity, and the modularity function [2] as first objective. We do not restrict the attribute type and allow to experiment different topological fitness functions. The representation adopted by this method is the label-based, while MOGA-@Net uses the locus-based one. Consequently, genetic operators are different because depending on the adopted representation. MOGA-@Net, moreover, employs a local merge procedure that avoids to obtain very small communities densely connected with neighboring larger communities. A comparison on seven real world networks highlights that MOGA-@Net obtains community structures more accurate than those obtained by *MOEA-SA*. Future work will explore other structural and similarity measures, and will try to extend the approach to dynamic networks and with multiple layers.

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

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

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[4] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[5] T. La Fond and J. Neville, "Randomization tests for distinguishing social influence and homophily effects," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10, 2010, pp. 601–610.

[6] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova, "Clustering attributed graphs: models, measures and methods," *Network Science*, vol. 3, no. 03, pp. 408–444, 2015.

[7] J. D. Cruz, C. Bothorel, and F. Poulet, "Entropy based community detection in augmented social networks," in *International Conference on Computational Aspects of Social Networks (CASON), 2011*. IEEE, 2011, pp. 163–168.

[8] D. Combe, C. Largeron, E. Egyed-Zsigmond, and M. Géry, "Combining relations and text in scientific network clustering," in *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*. IEEE, 2012, pp. 1248–1253.

[9] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "A model-based approach to attributed graph clustering," in *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. ACM, 2012, pp. 505–516.

[10] Z. Li, J. Liu, and K. Wu, "A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 7, pp. 1963–1976, 2018.

[11] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.

[12] C. Pizzuti, "Evolutionary computation for community detection in networks: a review," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 464–483, 2018.

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

[14] I. Falih, N. Grozavu, R. Kanawati, and Y. Bennani, "Community detection in attributed network," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18. International World Wide Web Conferences Steering Committee, 2018, pp. 1299–1306.

[15] C. Jia, Y. Li, M. B. Carson, X. Wang, and J. Yu, "Node attribute-enhanced community detection in complex networks," *Scientific Reports*, vol. 7, no. 2626, pp. 1–15, May 2017.

[16] M. E. J. Newman and A. Clauset, "Structure and inference in annotated networks," *Nature Communication*, vol. 7, no. 11863, pp. 1–11, 2016.

[17] J. Neville, M. Adler, and D. Jensen, "Clustering relational data using attribute and link information," in *Proceedings of the text mining and link analysis workshop, 18th international joint conference on artificial intelligence*, 2003, pp. 9–15.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[19] K. Steinhaeuser and N. V. Chawla, "Community detection in a large real-world social network," in *Social computing, behavioral modeling, and prediction*. Springer, 2008, pp. 168–175.

[20] T. Dang and E. Viennet, "Community detection based on structural and attribute similarities," in *International Conference on Digital Society (ICDS)*, 2012, pp. 7–12.

[21] H. Li, Z. Nie, W.-C. Lee, L. Giles, and J.-R. Wen, "Scalable community discovery on textual data with relations," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 1203–1212.

[22] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 718–729, 2009.
[23] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos, "PICS: parameter-free identification of cohesive subgroups in large attributed graphs," in *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.*, 2012, pp. 439–450.
[24] H. Elhadi and G. Agam, "Structure and attributes community detection: comparative analysis of composite, ensemble and selection methods," in *Proceedings of the 7th Workshop on Social Network Mining and Analysis*. ACM, 2013, p. 10.
[25] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Trans. Evolutionary Computation*, vol. 16, no. 3, pp. 418–430, 2012.
[26] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Physica A*, vol. 391, no. 15, pp. 4050–4060, 2012.
[27] C. Shi, Z. Yan, Y. Cai, and B. Wu, "Multi-objective community detection in complex networks," *Appl. Soft Comput.*, vol. 12, no. 2, pp. 850–859, 2012.
[28] P. Wu and L. Pan, "Multi-objective community detection based on memetic algorithms," *PLOS One*, vol. 10, no. 5, p. e0126845, 2015.
[29] X. Zhang, K. Zhou, H. Pan, L. Zhang, X. Zeng, and Y. Jin, "A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks," *IEEE Transactions on Cybernetics*, pp. 1–14, 2018.
[30] M. Gong, L.-J. Zhang, J.-J. Ma, and L. Jiao, "Community detection in dynamic social networks based on multiobjective immune algorithm," *Journal of Computer Science and Technology*, vol. 27, no. 3, pp. 455–467, 2012.
[31] F. Folino and C. Pizzuti, "An evolutionary multiobjective approach for community discovery in dynamic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1838–1852, 2014.
[32] 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*, vol. 44, no. 12, pp. 2274–2287, 2014.
[33] A. Amelio and C. Pizzuti, "An evolutionary and local refinement approach for community detection in signed networks," *International Journal on Artificial Intelligence Tools*, vol. 25, no. 4, pp. 1–44, 2016.
[34] ——, "Evolutionary clustering for mining and tracking dynamic multilayer networks," *Computational Intelligence*, vol. 33, no. 2, pp. 181–209, 2017.
[35] T. He and K. C. C. Chan, "Evolutionary community detection in social networks," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, July 6-11, 2014, Beijing, China*, 2014, pp. 1496–1503.
[36] Y. Park and M. Song, "A genetic algorithm for clustering problems," in *Proc. of 3rd Annual Conference on Genetic Algorithms, Morgan Kaufmann Publishers*, 1989, pp. 2–9.
[37] M. Gong, B. Fu, L. Jiao, and H. Du, "A memetic algorithm for community detection in networks," *Physical Review*, vol. E84, p. 056101, 2011.
[38] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *In the Eighth Conference on Parallel Problem Solving from Nature (PPSN VIII). Lecture Notes in Computer Science*. Springer-Verlag, 2004, pp. 722–731.
[39] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 181–213, Jan. 2015.
[40] C. Pizzuti, "Ga-net: A genetic algorithm for community detection in social networks," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2008, pp. 1081–1090.
[41] U. Brandes, M. Gaertler, and D. Wagner, "Engineering graph clustering: Models and experimental evaluation." *ACM Journal of Experimental Algorithmics*, vol. 12, no. 1.1, pp. 1–26, 2007.
[42] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.
[43] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[44] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.
[45] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
[46] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.
[47] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 u.s. election: Divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery*, ser. LinkKDD '05. New York, NY, USA: ACM, 2005, pp. 36–43.
[48] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 539–547.
[49] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd, Chichester, England, 2001.
[50] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[51] M. Li, S. Yang, X. Liu, and R. Shen, "A comparative study on evolutionary algorithms for many-objective optimization," in *Evolutionary Multi-Criterion Optimization*, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 261–275.
[52] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective oprimization: A comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26 194–26 214, Sep. 2018.
[53] L. Chen, H. Liu, K. C. Tan, Y. Cheung, and Y. Wang, "Evolutionary many-objective algorithm using decomposition-based dominance relationship," *IEEE Transactions on Cybernetics*, pp. 1–11, 2018.
[54] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 13:1–13:35, Sep. 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. Her research interests include knowledge discovery in databases, data mining, data streams, bioinformatics, social network analysis, evolutionary computation. She is serving as program committee member of international conferences, and as reviewer for several international journals.

Annalisa Socievole is a Post-doctoral Research Fellow at the National Research Council of Italy (CNR), Institute for High Performance Computing and Networking (ICAR). She received a PhD in Systems and Computer Science Engineering in February 2013 and a master's degree in Telecommunications Engineering in July 2009, both from the University of Calabria. From October 2011 to April 2012 she has been a visiting PhD student in the Systems Research Group of Cambridge Computer Laboratory (UK). From October 2013 to June 2014 she also spent a Post-doc period in NAS Group at TU Delft (The Netherlands) to carry out the research project CONTACTO. Her research interests include Opportunistic Networks, DTNs, complex networks and community detection algorithms.