Many-objective optimization for Community Detection in multi-layer networks

Clara Pizzuti, Annalisa Socievole
National Research Council of Italy (CNR)
Institute for High Performance Computing and Networking (ICAR)
Via Pietro Bucci, 7-11C
87036 Rende (CS), Italy
Email: {clara.pizzuti,annalisa.socievole}@icar.cnr.it

***Abstract*—A many-objective optimization algorithm for community detection in multi-layer networks is proposed. The method exploits the modularity concept as function to be simultaneously optimized on all the network layers to uncover multi-layer communities. In addition, three different strategies to choice the best solution from the set of solutions of the Pareto front are presented. Simulations on several synthetic networks reveal that our method is able to extract high quality communities. A comparison with state-of-the-art approaches shows that the method is competitive and, in many cases, it is also able to outperform existing community detection algorithms for multi-layer networks.**

***Keywords*-Community detection, multi-layer networks, many-objective optimization.**

I. Introduction

Many real-world complex systems are characterized by entities intertwined by multiple types of relationships, each representing a different aspect of interaction. In social networking environments, for example, a user usually has multiple relationships on different online social networking services. Facebook, Twitter, LinkedIn, are just some examples of the social dimensions an online user may have. A *multi-layer network*, consisting of a set of layers, each representing a kind of relation among entities, has gained increasing interest in the last years, as a valid formalism to model the multiple types of social connections between users with a graph structure. The concept of multi-layer network to model complex networks has not a unique terminology in the literature [1], and many different definitions have been proposed, such as *multiplex* [2], [3], [4], *multirelational* [5], *multidimensional* [6], [7].

An important problem in the study of multi-layer networks is community detection. A community refers to a set of nodes behaving differently from the rest of the other nodes. Focusing on the connections between nodes, within a community the number of edges is usually much higher than the number of edges with the remaining nodes of the network. In other words, the number of intra-community links is much higher than the number of inter-community links.

The problem of community detection in multi-layer networks is more challenging compared to the single-layer case. Many basic metrics that are commonly used by monoplex community detection algorithms (e.g. node centrality, node similarity, modularity, etc.) need to be reformulated and adapted when dealing with multi-layer networks. Moreover, how to define a ground-truth for such networks is still an open problem. As a consequence, assessing the quality of a partition is not as easy as for a single-layer network.

In recent years, a number of approaches have addressed the problem of community detection in multi-layer networks. The general strategy of existing algorithms is to reduce the problem to the classical setting of community detection in a single-layer network [8] [9] [10]. The choice of the monoplex community detection algorithm is often independent and in theory, any algorithm can be chosen. Another approach consists in extending an existing community detection algorithm to deal with multiplex networks. The approach adopted in [2], for example, extends the modularity quality function to the multi-layer case and then applies approaches optimizing modularity [11].

In this paper we propose to formulate the community detection problem in multi-layer networks as a many-objective optimization problem where a given objective is contemporarily optimized on all the network layers. The framework of Pareto optimality [12] is exploited to obtain a set of solutions corresponding to the best compromise objective values for all the layers. The method, named *MLMaOP* (*Multi-Layer Many-objective OPtimization algorithm*), obtains the set of competing objectives by evaluating an objective function on as many different decision domains as the number of layers. Since the many-objective algorithm obtains a family of solutions that represent the best trade-off between the objectives to optimize, a single solution must be selected and reported as the result of the method. Three different strategies to choice the best solution from the set of solutions of the Pareto front are proposed and compared. An extensive experimentation on several synthetic networks, for which the ground-truth division is known, reveals that our method is able to extract high quality communities. Moreover, a comparison with state-of-the-art approaches shows that the method is competitive and, in many cases, it is also able to outperform existing community detection algorithms for multi-layer networks.

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

This paper has been organized as follows. The next section provides some preliminaries on multi-layer networks and defines the problem of community detection. Section III introduces the many-objective optimization concept and formalizes the community detection problem in multi-layer networks. Section IV describes the proposed method. Section V gives a brief overview of the related multi-layer community detection methods. Section VI describes the experimental evaluation and shows the results. Finally, Section VII concludes the paper.

II. Preliminaries

A **multi-layer network** [13] is a pair $\mathcal{M} = (\mathcal{G}, \mathcal{C})$, where $\mathcal{G} = \{G_\alpha, \alpha \in \{1, \ldots, d\}\}$ is a family of graphs $G_\alpha = (V_\alpha, E_\alpha)$, called layers of $\mathcal{M}$, and

$$\mathcal{C} = \{E_{\alpha\beta} \subseteq V_\alpha \times V_\beta, \alpha, \beta \in \{1, \ldots, d\}, \alpha \neq \beta\} \qquad (1)$$

is the set of interconnections between nodes of two different layers G_α and G_β. The elements of $\mathcal{C}$ are called *inter-layers* or *crossed layers*, while those of E_α are called *intra-layer* connections. The adjacency matrix of each layer G_α is denoted as A^α, where an element $A^\alpha_{ij} = 1$ if nodes i and j are connected in E_α. The *projection network* of $\mathcal{M}$ is the graph $proj(\mathcal{M}) = (V_\mathcal{M}, E_\mathcal{M})$, where $V_\mathcal{M}$ is the union of all the V_α and $E_\mathcal{M}$ is the union of intra-layer and inter-layer connections. The adjacency matrix of $proj(\mathcal{M})$ is denoted by $\overline{\mathcal{A}_\mathcal{M}}$. A *multiplex network* is a particular multi-layer network where $V = V_1 = \ldots = V_d$ and the interlayer connections are implicitly only between a node in a layer and the counterpart node of the other layers. This implies that $\mathcal{M}$ can be characterized only by $\mathcal{G}$, thus $\mathcal{M} = \mathcal{G} = \{G_\alpha = (V, E_\alpha), \alpha \in \{1, \ldots, d\}\}$.

In the following, we deal with multiplex networks and the terms multiplex and multi-layer will be used as synonymous.

Community in multiplex networks. The concept of *community* in a multiplex network, as outlined in [14], does not have a generally accepted formal definition, rather it depends on the problem domain. Loe and Jensen [14] classify the different ideas of communities into three groups: *local definitions, global definitions* and *vertex similarity*. A *local definition* relies on the interactions among the members of a community and considers a community of high quality when the information flow among nodes is not interrupted even if a network layer is no more available. To this end, the concept of *redundancy* of a community $C \subseteq V$ has been proposed in [8] as a quality measure of C. Let $P \subseteq C \times C$ be the set of vertex pairs that are connected in at least a layer. The set of redundant pairs $P' \subseteq P$ are those connected in at least two layers, then the redundancy of a set of nodes C is the number of vertex pairs of C that are adjacent in more that two layers, normalized by the the maximum number of possible connections:

$$redundancy(C) = \frac{1}{d \times |P|} \sum_{G_\alpha \in \mathcal{G}} \sum_{\{u,v\} \in P'} \delta(u, v, E_\alpha) \quad (2)$$

where $\delta(u, v, E_\alpha) = 1$ if $(u, v) \in E_\alpha$, zero otherwise. As outlined in [14], this definition can give high score to communities with low density, which does not correspond to the intuitive idea of community.

The *global definition* of community takes into account both the internal and external interactions among communities. The *modularity function*, introduced by Newman and Girvan [15], is a very popular global measure to evaluate the goodness of a network partitioning in monoplex networks. Modularity Q is defined as:

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(C_i, C_j) \qquad (3)$$

where A is the adjacency matrix of the graph, m is the number of edges of the graph, k_i and k_j are the degrees of nodes i and j respectively, and $\delta(C_i, C_j)$ yields one if i and j are in the same community, zero otherwise.

Fixed a partition $\mathcal{P} = \{P_1, \ldots, P_k\}$ of the set V of vertices, the modularity $Q_\alpha(\mathcal{P}), \alpha = 1, \ldots, d$ on each of the layers is different. $\mathcal{P}$ is a good community structure of $\mathcal{M}$ if the modularity values on all the layers are high [6], [7].

The *vertex similarity* definition of community assumes that nodes belonging to the same community should be similar with respect to some measure. To this end, the *Cross-Layer Edge Clustering Coefficient (CLECC)* has been introduced by Bródka et al. [16] to compute the number of common neighbors between two nodes on all the layers. It is defined as

$$CLECC(u, v, \alpha) = \frac{\mid N(u, \alpha) \cap N(v, \alpha) \mid}{\mid N(u, \alpha) \cup N(v, \alpha) - \{u, v\} \mid} \quad (4)$$

where $N(u, \alpha)$ are the neighbors of node u in the layer α. In this paper we adopt the modularity function as criterion to optimize because for monoplex networks it proved to find solutions of high quality [11]. However, any other function could be used. Our objective can be defined as follows.

Problem definition: *Find a division of a multiplex network $\mathcal{M}$ such that the modularity of all the graphs $G_\alpha \in \mathcal{G}$ is maximized.*

To solve this problem, we propose to formalize the detection of a multiplex network partition as a modified *Many-Objective Optimization Problem* [17], where the objective space is obtained by evaluating the modularity function on each of the d layers. The next section describes in detail the proposal.

III. Many-Objective Optimization for Multiplex Networks

A multi-objective optimization problem (MOP) is defined as [18]

$$min_{\mathbf{x}} \;\; \mathbf{F}(\mathbf{x}) = (\mathbf{f_1}(\mathbf{x}), \mathbf{f_2}(\mathbf{x}), \ldots, \mathbf{f_d}(\mathbf{x})) \\ \textit{subject to } \mathbf{x} \in X$$

where d is the number of objective functions, $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in X$ is the decision vector with a domain of definition $X \subseteq R^n$, $\mathbf{F} : X \rightarrow \mathcal{Z}$ is the mapping from the decision space X to the objective space $\mathcal{Z}$.

When the number of objectives is more than 3, an MOP is referred to as Many Objective Optimization Problem ($MaOP$) [17]. In $MaOPs$ conflicting objectives, that must be simultaneously optimized, prevent to have a unique optimal solution to the problem, rather a set of solutions representing the trade-offs between the criteria are obtained. Since for many-objective problems a complete ordering of the solutions is difficult to define, in order to compare solutions, Pareto-dominance relation is used to define a partial ordering in the objective space [12]. Given two decision vectors $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}$ is said to *dominate* $\mathbf{y}$, denoted as $\mathbf{x} \prec \mathbf{y}$, if and only if

$$\forall i \in 1, 2, \ldots, d : f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \;\wedge\; \exists \; i \; s.t. \; f_i(\mathbf{x}) < f_i(\mathbf{y})$$

A solution $\mathbf{x} \in \mathbf{R^n}$ is said *Pareto optimal* if and only if $\nexists \mathbf{y} \in R^n \; s.t. \; \mathbf{y} \prec \mathbf{x}$

The *Pareto optimal set* is the set of all the Pareto optimal solutions, while the *Pareto front* is the set of the objective function values corresponding to the solutions in the Pareto optimal set.

The goal of optimizing a many-objective optimization problem is analogous to that of optimizing an MOP, i.e. find an approximation set of the Pareto front such that the solutions in this set are diverse in the objective space and as close as possible to the Pareto front [19].

The problem of detecting a community structure of a multiplex network $\mathcal{G} = \{G_\alpha, \alpha \in \{1, \ldots, d\}\}$ such that the modularity function is optimized on each G_α layer can be formalized as an $MaOP$, as follows.

Let $\Omega = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ be the set of feasible partitions of the multiplex network $\mathcal{G}$ and $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_d\}$ a set of d single criterion functions.

A many-objective community detection problem $MaCDP = (\Omega, \mathcal{F})$ on a multiplex network $\mathcal{G}$ is defined as

$$\begin{aligned} \textit{min } \mathcal{F}(\mathcal{P}) =& (\mathcal{F}_1(\mathcal{P}), \mathcal{F}_2(\mathcal{P}), \ldots, \mathcal{F}_d(\mathcal{P})) \\ \textit{subject to } & \;\; \mathcal{P} \in \Omega \end{aligned} \tag{5}$$

Each $\mathcal{F}_\alpha : \Omega \rightarrow \mathcal{R}$ computes the value of the objective function only on the layer G_α. For our aim, since Q must be maximized, $\mathcal{F}_\alpha(\mathcal{P}) = -Q_\alpha(\mathcal{P})$. Note that, the main difference between a classical $MaOP$ and an $MaCDP$ is that for the former the competing objectives $\mathcal{F}$ are all different and they are evaluated on the same domain, while in an $MaCDP$ it is possible to have a single criterion and the d objective values are obtained by evaluating each $\mathcal{F}_\alpha$ on a different domain of decision.

Since $\mathcal{F}$ is a vector of competing objectives that must be simultaneously optimized, this implies that the obtained nondominated solutions of the Pareto front optimize at the best the modularity of each layer. In the next Section a detailed description of the method is given.

IV. Method description

The *MLMaOP* method uses the locus-based adjacency representation [20], where an individual I of the population consists of $n =\mid V \mid$ genes $I = (g_1, \ldots, g_n)$. Each gene can assume a value j in the range $\{1, \ldots, n\}$. A value j assigned to the ith gene means that there is a link between the nodes i and j in at least a layer G_α. A decoding step identifies all the components, i.e. communities, of the multiplex graph. The kind of crossover operator is a standard uniform crossover, where a binary mask of length equal to the number of nodes is randomly generated, and an offspring is obtained by selecting from the first parent the genes where the mask is a 0, and from the second parent the genes where the mask is a 1. The mutation operator randomly changes the value j of a i-th gene to one of its the neighbors.

Though crossover and mutation operators, along with individual representation, can be those used from popular methods for community detection in monoplex networks, particular attention must be paid to the initialization process. In fact, since a couple of nodes can be connected in a layer but not in another one, and some nodes could be isolated in some layers, the initialization process assigns a value j to the i-th gene if there exists an edge $(i, j) \in E_{\mathcal{M}}$, i.e. if nodes i and j are connected in one of the layers. This type of initialization allows the exchange of information among the layers and avoids solutions with isolated nodes, even if they are present in some layer.

The pseudo-code of the *MLMaOP* algorithm is reported in Figure 1. The method initializes the population as described (step 1), and 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, it evaluates the modularity function of the current solution of all the layers to obtain $\{\mathcal{F}_1, \ldots, \mathcal{F}_d\}$ objective values (steps 3-7). A rank is then assigned to solutions based on Pareto dominance (step 8), and a new population is created by applying genetic operators to the best selected points from the combined parent and offspring populations (steps 10-11). At the end of the computation the method returns the set of Pareto-optimal solutions (step 13). Though each of these solutions corresponds to a different tradeoff of the modularity values on each layer, a criterion should be

```
The MLMaOP Method:
Input:  A multiplex network $\mathcal{M} = \{G_\alpha = (V_\alpha, E_\alpha), \alpha \in \{1, \ldots, d\}\}$
Output: A partitioning $\mathcal{P}$ of the nodes of $\mathcal{M}$ in communities

Initialize a population of random individuals by using the adjacency
matrix $\overline{\mathcal{A}_{\mathcal{M}}}$ of $proj(\mathcal{M})$
while termination condition is not satisfied do
  for each individual $I = \{g_1, \ldots, g_n\}$ of the population
    Decode $I$ to generate a partitioning $\mathcal{P}$ of $\mathcal{M}$
    for $\alpha = 1 : d$
      evaluate the objective function on $G_\alpha$ to obtain a value $\mathcal{F}_\alpha$
      end for
    Assign a rank based on Pareto dominance
  end for each
  Combine parents and offspring and partition into fronts;
  Select the best points, and apply the variation operators
  to create the next population;
end while
choose a solution from the Pareto front and return it
```

Figure 1. The pseudo-code of the *MLMaOP* algorithm.

established to automatically select one solution with respect to another. In the following, three approaches are described.

Model Selection. Evolutionary many-objective algorithms obtain a family of solutions that represent the best trade-off between the objectives to optimize. However, a single solution must be selected and reported as the result of the method. In [21] Purshouse et al. classify multiple criteria decision making techniques into three categories: *a priori*, *interactive*, and *a posteriori*. In an *a priori* approach preferences are included prior to the search process. In an *interactive* method preferences are added progressively during the optimization process. In an *a posteriori* approach first an approximation of the optimal Pareto front is obtained, and then a solution from the trade-off solutions is selected. Since as MOEA method [18] we use $NSGA-II$, which tries to obtain a well-converged and well-distributed set of solutions, we need to define a strategy to choose one of the solutions of the approximated optimal Pareto front. To this end, we propose and compare three strategies.

The former takes the solution having the maximum modularity value computed on $\overline{\mathcal{A}_{\mathcal{M}}}$, the adjacency matrix of $proj(\mathcal{M})$. In this case the method will be referred as *MLMaOP-proj*.

Let $y_i = (f_1, f_2, \ldots, f_d)$ be the fitness values of the i-th solution of the Pareto front. The second strategy, referred as *MLMaOP-mf*, computes the mean value $mf_i = (\sum_i f_j)/d$ for each y_i, and the solution having the highest value of mf is chosen as final.

The third strategy, referred as *MLMaOP-cspa*, relies on the concept of *consensus clustering* introduced by Strehl and Gosh [22] in the context of cluster ensembles, where different clusterings of a set of data, obtained by multiple learning models, are combined *to improve the quality and robustness of results*. In [22] the authors propose three consensus functions. We adopt the so-called *Cluster-based Similarity Partitioning Algorithm (CSPA)*, that induces a similarity measure among pairs of partitions, used then to re-cluster objects. For our purposes, we consider the solutions of the Pareto front returned by *MLMaOP* as the clusterings to combine. Let $\lambda_q \in N^n$, $q = 1, \ldots, r$, be the label vector of the r Pareto front clustering solutions, then a binary indicator matrix $\mathbf{H}^{(q)}$ of size $n \times k$ is built for each λ_q, where k is the number of different labels of λ_q. Each row i contains a 1 value in correspondence of the column j, if the i-th node has been assigned to the j-th cluster, and zero in all the other positions. Then, based on the idea that two nodes have similarity 1 if they belong to the same community, and 0 otherwise, an $n \times n$ similarity matrix $\mathbf{S}$ can be created from the concatenated block matrix $\mathbf{H} = (\mathbf{H}^{(1)}\mathbf{H}^{(2)} \cdots \mathbf{H}^{(r)})$ as $\mathbf{S} = \frac{1}{r}\mathbf{H}\mathbf{H}^T$. The entries of $\mathbf{S}$ are the fraction of partitionings in which two nodes belong to the same community. $\mathbf{S}$ is used to recluster the nodes, by executing a single objective genetic algorithm that optimizes the modularity function having as input network that corresponding to the matrix $\mathbf{S}$. The genetic algorithm gives a partitioning that takes into account the agreement level of all the solutions of the Pareto front.

V. Related work

In the last years the interest in multi-layer networks has generated a consistent number of algorithms to detect community structure [23], [14]. Many approaches either consider the projection matrix and optimize a criterion function on this matrix, or apply an existing method on each layer and then combine the results.

Because of the high number of proposals, in the following we review only the methods reported in [14], which will be compared with our algorithm.

A naive strategy that projects the multi-layer network into a single graph and then applies a monoplex-community detection algorithm has been proposed by several authors. More specifically, the *Projection-Average (PA)* strategy considers an average adjacency matrix [6] for the multi-layer network given by $\bar{A} = \frac{1}{d}\sum_{i=1}^{d} A^i$, where d is the number of layers, while *Projection-Binary (PB)* [14] and *Projection-Neighbors (PN)* [16] are variants that consider the unweighted adjacency matrix, the former, and the adjacency matrix weighted with the ratio of common neighbors, to take into account stronger ties, the latter. The *Louvain algorithm* [11] is then applied for detecting communities within the projected graphs.

The *Cluster-based Similarity Partition algorithm (CSP)* computes the similarity between node pairs in terms of occurrences within communities at different layers. Then, the k-means clustering method is applied to extract the principal cluster. k is chosen as the value that maximizes the modularity of the multi-layer network [14].

In the *Generalized Canonical Correlations (GCC)* algorithm [14], communities on single layers map each vertex of the multi-layer network within an l-dimensional Euclidean space, where the shorter the path between two vertices is the closer they are in the Euclidean space. Given m graph layers, for each layer a structural feature matrix S^i of size $l \times m$, where each column is the position of a vertex in the l-dimensional Euclidean space, is defined. Then, these matrices are averaged trough a set of linear transformations maximizing the pairwise correlations of the S^i and defining a principal structural feature matrix $\bar{S}$. Finally, the principal partition of the multi-layer network is determined applying the k-means clustering to $\bar{S}$.

CLECC Bridge Detection (CLECC-BD) [16] reduces the graph to a weighted graph and then computes the CLECC score for each pair. Since pairs with low CLECC score are considered weak ties and strong ties are desirable within a community, at each iteration, nodes having low CLECC values are disconnected until a predefined number of communities is obtained.

VI. Experimental Evaluation

In this section we test the effectiveness of *MLMaOP* on two types of synthetic networks, for which the partitioning in communities is known. For the former benchmark the performance of the method is evaluated by varying the number of layers and the network structure. The second benchmark, instead, is used to compare *MLMaOP* with the methods described in the previous section. The results of these methods are those reported by Loe and Jensen in [14]. *MLMaOP* has been written in MATLAB 2015b by using the Global Optimization Toolbox. The results of *MLMaOP* are the average values obtained by running the method 10 times, population size 200, 100 generations, mutation rate 0.2, and crossover fraction 0.8. These parameter values have been selected as those giving the best results by employing a trial-and-error procedure on the benchmark data sets.

A. Synthetic networks

mLFR-128. The first collection of synthetic networks has been generated by using the benchmark[1] proposed by Bródka and Grecki [24], which is an extension of the LFR benchmark by Lancichinetti *et al.* [25]. The parameters used for generating the mLFR datasets are shown in Table I. Each layer of a multi-layer network consists of 128 nodes partitioned in different non-overlapping communities having variable sizes. Specifically, we generated multi-layer networks of 2, 3 and 4 layers, managing the number of inter-community edges and the difference in terms of a node's degree between layers by varying the *mixing parameter* μ and the *degree change chance (DCC)*, respectively, as reported in Table I. The mixing parameter μ is the fraction of links shared by a node with the nodes of its community. When $\mu < 0.5$ the number of neighbors of a node inside its group are more than the number of neighbors belonging to the other groups. The degree change chance controls how much different the network layers are in terms of node degree. The higher the DCC of a network, the more the nodes may have different degree values within different layers, and thus the more diverse the networks on different layers.

Table I
mLFR-128 parameters setting.

Parameter	Value
Number of nodes	128
Node average degree	8
Node maximal degree	16
Mixing parameter	[0.1, 0.2, 0.3, 0.4, 0.5]
Exponent for power law creating degree sequence	2
Exponent for power law creating community sizes	1
Overlapping nodes	0
Overlapping memberships	0
Maximal community size	32
Minimal community size	8
Number of layers	[2, 3, 4]
Degree change chance	[0.2, 0.4, 0.6, 0.8]
Membership swap chance	0
Exponent for power law of nodes through layers	2

SSRM. A second type of synthetic dataset named *Structured Synthetic Random Multiplex (SSRM)* [14] contains a two-layer network with 128 nodes on each layer. SSRM starts considering a set of four high-quality multi-layer communities (i.e., good communities on both layers), namely c_1, c_2, c_3, c_4, that are successively recombined in order to create three new partitions that satisfy the quality indexes of multi-layer network described in Section II, i.e. redundancy, modularity, and CLECC. The first partition of the two-layer network in two communities, $P1 = \{[c_1, c_2], [c_3, c_4]\}$, has high redundancy, low modularity and low CLECC. The second partition $P2 = \{[c_2, c_3], [c_1, c_4]\}$ has high CLECC, but low values for the remaining indexes. Finally, the third partition $P3 = \{[c_1, c_3], [c_2, c_4]\}$ has high modularity with the other indexes with low values.

B. Evaluation measure

Since for both benchmarks the ground truth partitioning is known, to assess the quality of the solutions we use the well known evaluation measure of *Normalized Mutual Information (NMI)* [26].

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.

$$NMI(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} C_{ij} log(C_{ij}n/C_{i.}C_{.j})}{\sum_{i=1}^{c_A} C_{i.} log(C_{i.}/n) + \sum_{j=1}^{c_B} C_{.j} log(C_{.j}/n)} \quad (6)$$

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 $A = B$,

[1] mLFR Java code is available at https://www.ii.pwr.edu.pl/∼brodka/mlfr.php

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

C. Results

Fig. 2 shows the normalized mutual information results for the mLFR-128 networks against different values of the mixing parameter μ. Each point in the plot corresponds to values averaged over 10 runs. For this first experiment, we focused on comparing the three *MLMaOP* strategies, varying the number d of layers of the multi-layer network and the degree change chance.

The first observation is that for low μ values ($0.1 \leq \mu \leq 0.4$), *MLMaOP-cspa* outperforms the other two strategies. For these values, there are few links between communities and the well defined structure of communities in distinct groups benefits the consensus-based strategy, which is able to produce higher NMI values. When the mixing parameter is 0.5, thus the inter-links between communities increase, the NMI degrades, as expected, but *MLMaOP-mf* and *MLMaOP-proj* seem to better capture communities with respect to *MLMaOP-cspa*.

We further observe that as the number of layer increases, *MLMaOP* performs better. For $d = 4$, for example, the NMI values are always greater than 0.75, thus showing that also high mixing parameter values do not degrade the performance of our strategies.

We finally highlight that within the considered network scenarios, the DCC parameter does not significantly influence our community detection schemes. In other words, even if the DCC increases making the network layers more different, *MLMaOP* is always able to uncover communities with a high NMI.

As second experiment, we considered the SSRM network and compared *MLMaOP* with the multi-layer community detection algorithms described in Section V. As already observed, the aim of this experiment is to check the capability of a method to capture community structure of a multi-layer network satisfying different concepts of high quality multiplex partitions, represented by the three ground-truth partitions $P1$, $P2$ and $P3$. Table II shows the NMI values obtained for each ground truth and each algorithm. Every value corresponds to the average value over 10 different instances of the $SSRM$ network. As regards P1, where redundancy is high, our three strategies outperform all the contestant methods, and are able to find communities with higher NMI value. For P2, where $CLECC$ is high, the methods based on projection and the *CLECC-BD* algorithm, which optimizes the $CLECC$ measure, obtain a much higher NMI value than the other methods. However, it is worth noting that our approach is much better than *Cluster-based Similarity Partition* and *Generalized Canonical Correlations* to capture good communities.

In P3, where the modularity is high, our approach outperforms all the other methods when the consensus strategy is applied. However, also *MLMaOP-proj* and *MLMaOP-mf* obtain higher values than all the projection methods and *CLECC-BD*. These results show that *MLMaOP* is very competitive with respect to the other approaches.

Table II
THE NMI VALUES FOR THE DIFFERENT ALGORITHMS IN COMPARISON AND THE DIFFERENT GROUND-TRUTHS IN SSRM DATASET.

Algorithm	P1	P2	P3
MLMaOP-proj	0.053	0.271	0.082
MLMaOP-cspa	0.051	0.338	0.169
MLMaOP-mf	0.054	0.282	0.075
Projection Average	0.002	0.94	0.017
Projection Binary	0	0.983	0
Projection Neighbors	0	0.978	0
Cluster-based Similarity Partition	0.019	0.14	0.083
Generalized Canonical Correlations	0.004	0.002	0.158
CLECC-BD	0.006	0.964	0.006

VII. Conclusions

The paper proposed a method for uncovering community structure in multi-layer networks based on many-objective optimization. The framework obtains the set of competing objectives by simultaneously evaluating an objective function on as many different decision domains as the number of layers. Moreover, three different strategies to choice the best solution from the set of solutions of the Pareto front are proposed and compared. Experimentation on several synthetic networks, for which the ground-truth division is known, showed that the method is able to extract high quality communities, also when compared with state-of-the-art approaches. It is worth to observe that the consensus clustering approach for model selection, generally, obtains better results than the other strategies. The motivation could be that the execution of a genetic algorithm on the similarity matrix obtained by the concatenated block matrices tries to find the *latent community structure* shared, as much as possible, among all the already obtained partitionings on each level. This is similar to the objective of Tang et al. [6] of finding a shared latent community structure among the entities of a multi-layer network.

The current implementation of *MLMaOP* uses the *NSGA-II* of Deb et al. [27], optimized for at most three objectives. As pointed out in [28], a main difficulty that Pareto-based evolutionary algorithms have to face is the loss of selection pressure due to the high number of nondominated solutions generated in a population of limited size, when the number of objectives increases. This leads to lower convergence rate to the Pareto front. Thus, several extensions have been proposed to multi-objective methods to overcome performance deterioration of MaOPs [29], [19]. Even if *MLMaOP* obtains good results until four layers, future work will experiment the method on MOEA frameworks suitably defined for many-objective optimization, such as *NSGA-III* [30], [31], to efficiently deal with many layers, typical in many real-world complex systems.

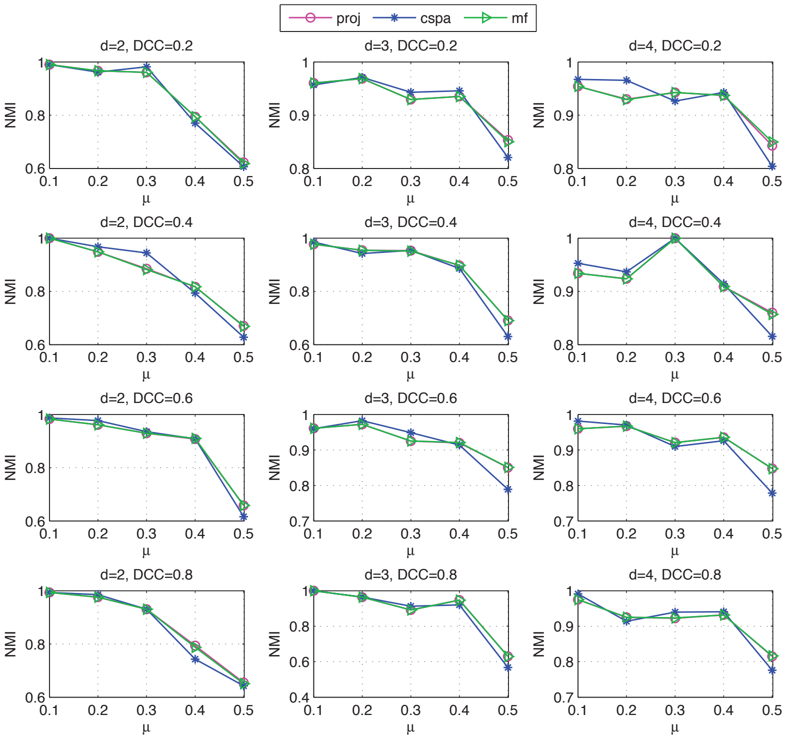

Figure 2. NMI results for mLFR-128 networks with increasing μ, number of layers d, and different degree change chance (DCC).

References

[1] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *arXiv:1309.7233v3*, 2014.

[2] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[3] M. D. Domenico, A. Sole, S. Gómez, and A. Arenas, "Random walks on multiplex networks," *arXiv:1306.0519*, 2013.

[4] F. Battiston, V. Nicosia, and V. Latora, "Metrics for the analysis of multiplex networks," *arXiv:1308.3182v2*, 2013.

[5] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Community mining from multi-relational networks," in *9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005)*, 2005, pp. 445–452.

[6] L. Tang, X. Wang, and H. Liu, "Uncoverning groups via heterogeneous interaction analysis," in *The Ninth IEEE International Conference on Data Mining (ICDM'09)*, 2009, pp. 503–512.

[7] ——, "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, 2012.

[8] M. Berlingerio, M. Coscia, and F. Giannotti, "Finding and characterizing communities in multidimensional networks," in *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2011, Kaohsiung, Taiwan, 25-27 July 2011*, 2011, pp. 490–494.

[9] D. Suthers, J. Fusco, P. Schank, K.-H. Chu, and M. Schlager, "Discovery of community structures in a heterogeneous professional online network," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, 2013, pp. 3262–3271.

[10] L. Tang and H. Liu, "Community detection and mining in social media," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 1–137, 2010.

[11] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefevre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. P10008, 2008.

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

[13] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. G.-G. nes, M. Romance, I. S. na Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1–122, 2014.

[14] C. W. Loe and H. J. Jensen, "Comparison of communities detection for multiplex," *Physica A*, vol. 431, pp. 29–45, 2015.

[15] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review*, vol. E69, p. 026113, 2004. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0308217

[16] P. Bródka, T. Filipowski, and P. Kazienko, "An introduction to community detection in multi-layered social network," in *Information Systems, E-learning, and Knowledge Management Research - 4th World Summit on the Knowledge Society, WSKS 2011, Mykonos, Greece, September 21-23, 2011. Revised Selected Papers*, 2011, pp. 185–190.

[17] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proc. of the NAFIPS-FLINT Int. Conference*, 2002, pp. 233–238.

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

[19] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Survey*, vol. 48, no. 1, pp. 13:1–13:35, 2015.

[20] 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.

[21] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang, "A review of hybrid evolutionary multiple criteria decision making methods," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, 2014, pp. 1147–1154.

[22] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.

[23] J. Kim and J. Lee, "Community detection in multi-layer graphs: A survey," *SIGMOD Record*, vol. 44, no. 3, pp. 37–48, 2015.

[24] P. Bródka and T. Grecki, "mLFR benchmark: Testing community detection algorithms in multi–layered, multiplex and multiple social networks," to appear.

[25] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

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

[27] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[28] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review." in *IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 2419–2426.

[29] C. von Lücken, B. Barán, and C. A. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Comp. Opt. and Appl.*, vol. 58, no. 3, pp. 707–756, 2014.

[30] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[31] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb, "EliteNSGA-III: An improved evolutionary many-objective optimization algorithm," in *IEEE Congress on Evolutionary Computation, CEC 2016, Vancouver, BC, Canada, July 24-29, 2016*, 2016, pp. 973–982.