Future Generation Computer Systems 101 (2019) 221–235

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Multilayer overlapping community detection using multi-objective optimization

M.R. Shahmoradi [a], M. Ebrahimi [a,*], Z. Heshmati [a], M. Salehi [a,b]

[a] Department of Network Science and Technology, Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran
[b] School of Computer Science, Institute for Research in Fundamental Science (IPM), P.o.Box 19395-5746, Tehran,

HIGHLIGHTS

- Novel definition for multilayer communities and overlapping multilayer communities.
- A new method to detect multilayer overlapping and disjoint communities is proposed.
- Optimizing the multi-objective mathematical model which is based on the definition.
- Uses all Pareto front, instead of choosing one, to have robustness and flexibility.
- We proposed a novel two-phase method for optimizing and selecting communities.

ARTICLE INFO

Article history:
Received 23 January 2019
Received in revised form 12 May 2019
Accepted 23 May 2019
Available online 29 May 2019

Keywords:
Overlapping community detection
Multi-objective optimization
NSGA-II
Multilayer networks

ABSTRACT

Community detection in multilayer networks such as social or information networks, due to its vast practical applications has attracted many attentions in the past years. Although some researches have been done to develop monoplex methods to multilayers, but because of the complexity of multilayer networks they are in their infancy. In this study, initially, the definition of community in single layer networks is extended and a new definition for multilayer community is presented. Then, regarding the importance of overlapping communities in real networks, a comprehensive definition for overlapping multilayer community is presented. Furthermore, in order to address the problem of multilayer community detection, a two-phase approach has been adopted. In the first phase, a multi-objective mathematical model is developed, and optimized using a genetic algorithm based method (NSGA-II) to achieve a set of solutions (Pareto front). Each potential solution represents a partition of the multilayer network. In the second phase, the best solution is chosen from the Pareto fronts, using a novel algorithm. This proposed algorithm, handles the obstacles of genetic representation and is able to detect the optimal multilayer overlapping communities. Experiments on both synthetic mLFR networks with different parameters and six real data networks show the performance of the algorithm in terms of NMI and multilayer modularity.

1. Introduction

In the recent years, complex networks analysis due to their widespread applications has been extensively studied in the literature [1,2]. Networks representation of data generally provide a more accurate picture of what goes on in reality, because of its emphasis on the interaction among entities. However, most studies in this area usually consider one type of relation between nodes, which usually presents an incomplete picture of the reality [3]. Nevertheless, real world network data, due to the diverse and often parallel forms of relations between nodes, often contain different aspects of interactions among entities. This type of complex networks can be modeled as a multilayer network, where each layer represents one type of relation or interaction.

In addition, in the analysis of complex networks, community detection is one of the significant tools where memberships of a node in one or more communities is found based on either location density, similarity function and/or a null model. Community detection is used in many applications such as search engines, recommendation systems [4], fraud detection, link prediction [5] and community evolution [6], where it has attracted a lot of attention.

In the literature, there is no universally accepted definition of multilayer community. However most of the researches in

* Correspondence to: Department of Network Science and Technology, Faculty of New Science and Technologies, University of Tehran, North Kargar Street, Tehran, Iran.
E-mail addresses: re.shamoradi@ut.ac.ir (M.R. Shahmoradi), mo.ebrahimmi@ut.ac.ir (M. Ebrahimi), zheshmati@ut.ac.ir (Z. Heshmati), mostafa_salehi@ut.ac.ir (M. Salehi).

https://doi.org/10.1016/j.future.2019.05.061
0167-739X/

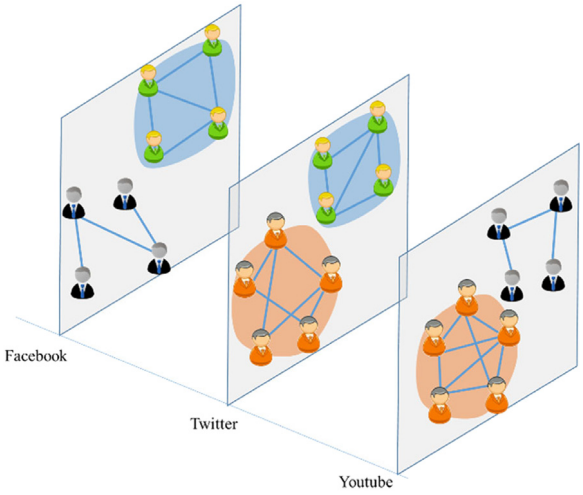

Fig. 1. Multilayer communities may form in different subset of layers.

this field, attempt to detect communities which are formed in all layers [7], but this assumption is not always true. Multilayer community detection should also be carried out with regards to just the layers in which the nodes have congested connections. In other words, we assume a multilayer community is a subset of nodes that have dense connections in a *subset of layers*.

Consider the ff-tw-yt multilayer network[1] [8], a network of people whose layers represent their interactions on social networks Facebook, Youtube and Twitter. Multilayer community detection should have the ability to recognize the community by taking into account all layers. Obviously, multilayer communities due to dense interactions in more than one layer, provides a more suitable analytical tool for further use (such as, link prediction, extracting network structure, community evolution, etc.) Because of the observation of the behavior of sets of nodes, which interact in different layers, we can get information about future interactions in other layers or studies for the reason of links formation.

On the network, it is possible to extract groups of people who have formed communities in different online networks (for example, Facebook and Twitter) and study their current/future relationships in other layers (such as Youtube), or the effect of the layers on each other. Obviously, determining the type of layers is important for future analysis and the recognition of community meanings.

It is important to take into account only the layers in which the community is composed of, whilst considering all the layers may distort the process of community detection and not yield the correct results. On the other hand, specifying layers in communities makes it easier to interpret the final multilayer communities. In this paper, we propose an algorithm for the detection of communities that are formed in either some or all the layers in the multilayer network. Fig. 1 shows a multilayer network with three layers and two communities in different layers.

Communities in the real world are not necessarily disjoint. For example, people in social networks may have multiple roles and community memberships. In many researches [9,10] it is shown that overlapping communities are a significant feature of many real world networks. In multilayer networks, there are limited studies in overlapping communities. Our proposed algorithm makes it possible to detect multilayer overlapping as well as disjoint communities.

[1] http://multilayer.it.uu.se/datasets.html.

The problem of community detection in a multilayer network can effectively be addressed by multi-objective optimization approach. In this paper, initially, a multi-objective mathematical model is proposed which takes into account the different aspects of a community and then using a genetic algorithm based multi-objective optimization method, tries to find the best partition of nodes with regards to the layers. Then, a novel algorithm is proposed to select the final overlapping multilayer communities using all the Pareto front solutions.

The proposed method is evaluated on synthetic mLFR networks with different mixing parameter, overlapping size and number of overlapping nodes as well as six real data networks. The evaluation is done in terms of two well-known evaluation metrics, namely NMI and multilayer modularity. The results obtained are then compared with three of the main algorithms in this field.

The main contributions of this paper are as below:

- A multi-objective mathematical model for multilayer community detection is given
- A novel algorithm is proposed to select the best multilayer overlapping communities from the Pareto optimal solutions.

The rest of this paper is organized as follows: Section 2 describes the related works on multilayer community detection. Section 3 gives the mathematical model of the problem in question, and the next section explains the proposed method. In Section 5 computational complexity is discussed. The performance of the proposed method is then evaluated on both synthetic and real networks as presented in Section 6, and finally conclusions are drawn and summarized in the last section of the paper.

2. Related works

In the literature, different definitions have been presented for a 'community'. With the emergent of multilayer networks, this heterogeneity in the definitions has become more complex. Since in multilayer networks community of a node is formed according to the nature of the layer. One of the early methods used is flattening, where the layers are combined into a single layer and the multilayer network is converted into a single layer weighted/unweighted network. Then, any single layer community detection algorithm can be applied. Nevertheless, in order to use as much as the information that is implicit in layers as possible, some of the single layer community detection methods have been extended to multilayer networks. For example, Berlingerio et al. in (Berlingerio, Coscia, & Giannotti, Finding and characterizing communities in multidimensional networks, [11]) states that a community must have a high rate of redundant interactions in different layers. The main idea is, the information flow between members of a community in different layers must be stable. From this point of view, obviously, a community can be evaluated locally and apart from the rest of the network. Hence, they defined the redundancy measure as:

$$\rho_c = \sum_{(u,v)\in\overline{\overline{p_c}}} \frac{|\{d: \exists(u, v, d) \in E\}|}{|D| \times |p_c|} \tag{1}$$

which $|D|$ is the number of layers, $|p_c|$ is the number of edges of community c and $\overline{\overline{p_c}}$ represents pairs of nodes in the community c, which are connected to each other, in more than one layer. As another example [12] present a method to extend the well-known clique percolation method (CPM) to multilayer networks.

In other studies, the structure of links in all layers are considered. For example Tang et al. in [7] believe that if there is a hidden community structure in a multilayer network, the corresponding subset of layers must also be discovered. From this point of view,

a multilayered community should also have high modularity in its layers. Therefore, the authors propose Principal Modularity Maximization (PMM) method which uses the structural feature of each layer to extract communities in the layer and obtains the final communities via modularity maximization. In [13] modularity is defined as the number of links within a community compared to the expected number of links based on a null model. With the emergence of multilayer networks, researchers have extended modularity to multilayer networks.

A well-known generalization of the popular modularity is introduced by Mucha et al. [14]. The authors represent each layer as a slice with an adjacency matrix as well as coupling links between nodes in slices, and the node-layer tuples are assigned to a community. So the communities are detected with regards to the layers. The following formula (2) is used for multi-slice modularity:

$$Q_{multi-slice} = \frac{1}{2\mu}\sum_{ijsr}\left\{\underbrace{\left(A_{ijs}-\gamma_s\frac{k_{is}k_{js}}{2m_s}\right)\delta_{sr}}_{\substack{\text{intra-slice adjacency}\\ \text{and null model}}}+\underbrace{\delta_{ij}C_{jsr}}_{\substack{\text{inter-slice identity}\\ \text{coupling links}}}\right\}\underbrace{\delta(g_{is},g_{jr})}_{\text{Kronecker delta}} \tag{2}$$

where μ is a normalization factor, and γ_s is a resolution factor, s and r indicates indices of slices, i and j, are indices of nodes, A is adjacency matrix, k is degree of node, m is number of edges, C is coupling edges matrix between slices and finally δ is Kronecker delta.

However, the computational complexity of the multi-slice modularity causes the optimization of the $Q_{multi-slice}$ to become a hard problem and restricts the application of modularity maximization based methods.

Another approach is introduced in (Berlingerio, Pinelli, & Calabrese, Abacus: frequent pattern mining-based community discovery in multidimensional networks, [15]), where the authors propose a new method using association rule mining, a traditional data-mining problem to find frequent item sets. In this method, using any community detection algorithm, communities in each layer, are detected. Then, it groups together the nodes that share memberships in single layer communities across the different layers. It then uses a traditional data-mining algorithm to find the maximum number of nodes that have appeared in common layers.

In [16] a random walk based probabilistic method is presented called Locally Adaptive Random Transitions (LART). LART initially, runs some random walkers in each layer, and then calculates a dissimilarity measure between nodes. This measure is then used to obtain the multilayer communities. Also, in [17] the authors propose a network reduction based method, which after simplifying the network using layer reduction, tries to detect multiplex communities.

Some authors have also utilized multi-objective optimization approaches. This category of methods have been primarily developed for single layer networks [18–21], and with the advent of multilayer networks, some have extended to this kind of networks. For example, Amelio and Pizzuti [22] defined two objective functions, which in a predetermined and ordered network layers, uses modularity maximization to partition the nodes in each layer and simultaneously using a temporal smoothness method, minimizes the difference between the results on previously considered layers.

Also in [23], the authors use a set of measures as objective functions and by optimizing them, expect to obtain an appropriate Pareto front. They then propose three strategies to select the best solution from the Pareto front.

3. Mathematical model of the problem

In order to carry on with the discussion, it is important to clarify and define the terms being used in the literature. Most importantly, definitions of multilayer network as well as strong and weak multilayer communities, as will be used in this research.

Definition 3.1 (*Multilayer Network*)**.** In this study we present a multilayer network as $\mathcal{M} = (\mathcal{V}, \mathcal{D}, \mathcal{G})$, where $\mathcal{D}$ is the layer set, $X^1, X^2, \ldots.., X^{\mathcal{D}} \subseteq \mathcal{V}$ and $\mathcal{V}$ is the set of nodes, $\mathcal{G} = \{G^\alpha : \alpha \in \{1, \ldots, \mathcal{D}\}\}$ is the set of graphs. In addition, the layer α is represented as $G^\alpha = (X^\alpha, E^\alpha)$ in which $E^\alpha \subseteq X^\alpha \times X^\alpha$ and $X^\alpha = \{x^\alpha_1, \ldots, x^\alpha_{N_\alpha}\}$, also each set of edges is represented as:

$$e^\alpha_{ij} = \begin{cases} 1 & if (x^\alpha_i, x^\alpha_j) \in E^\alpha, \\ 0 & o.w. \end{cases}$$

In addition, the results will be a set of communities $\mathcal{S} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$, in which $\mathcal{C} = (g, L)$, $g \subseteq \mathcal{V}$ and $L \subseteq \mathcal{D}$.

In order to have a formal definition of a multilayer community, inspired by the definition of community in simple networks by Radicchi et al. in [24] we extend the definition to multilayer strong community and multilayer weak community as given below:

Definition 3.2 (*Multilayer Strong Community*)**.** A set of nodes such that in every layer in which the community is present, the internal degrees of nodes are higher than the external degrees. In other words,

$$\begin{aligned} &\forall l' \in L : \forall i \in g : k^{in}_{i,l'}(g) > k^{out}_{i,l'}(g) \\ &k^{in}_{i,l'}(g) = \sum_{j\in g} A^{l'}_{ij} \\ &k^{out}_{i,l'}(g) = \sum_{j\notin g} A^{l'}_{ij} \end{aligned} \tag{3}$$

where $A^{l'}_{ij}$ is the i and jth element of adjacency matrix A of the layer l' and $k^{in}_{i,l'}(g)$ represent the input degrees of node i in layer l' to community g and subsequently $k^{in}_{i,l'}(g)$ represent the output degrees of node i in layer l' from community g.

Definition 3.3 (*Multilayer Weak Community*)**.** A set of nodes such that in every layer in which the community is present, the sum of the internal degrees of nodes is higher than the sum of the external degrees. This is better described as:

$$\sum_{l'\in L}\left(\sum_{i\in g} k^{in}_{i,l'}(g)\right) > \sum_{l'\in L}\left(\sum_{i\in g} k^{out}_{i,l'}(g)\right) \tag{4}$$

With the definitions given in formulas (3) and (4), we outline a multi-objective mathematical model for community detection in multilayer networks. In the mathematical model, we have two groups of decision variables: first $i \in \mathcal{V}$, the nodes and the second, $L \subseteq \mathcal{D}$, the network layers (see Fig. 2).

3.1. Objective functions

In this paper, two objective functions are proposed. Taking into account the definition of weak community, in the first function we evaluate the ratio between the internal and external degrees of the communities in each layer. Therefore, the first function is defined as:

$$f_1(g, l') = \sum_{i\in g}\frac{k^{out}_{i,l'}(g)}{\left(k^{in}_{i,l'}(g) + k^{out}_{i,l'}(g)\right)^\alpha}, \tag{5}$$

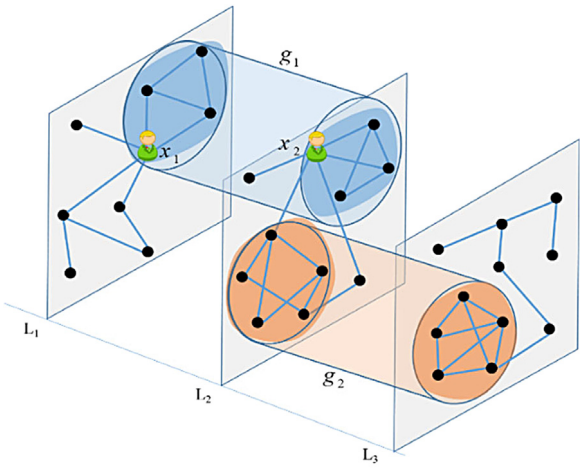

Fig. 2. Multilayer weak community (g_1) and multilayer strong community (g_2). Community g_1 is a weak community because nodes x_1 and x_2 have 3 out-degree and 2 in-degree with respect to community g_1.

in which $f_1\left(g, l'\right)$ shows the quality of the community $\left(g, l'\right)\cdot\alpha$ is the community size control, which is normally equal to 1, which means we simply calculate the ratio of external degrees. A small number for α leads to dense and small communities and vice versa. Considering Eq. (5), to evaluate a multilayer community $(g, \{L\})$ the first objective function is:

$$FS_1\left(g, \{L\}\right) = \frac{\sum_{l'\in L} f_1\left(g, l'\right)}{|L|}, \tag{6}$$

And finally the first objective function to evaluate partition $\mathcal{S} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$, in which $\mathcal{C}_i = (g_i, \{L\}_i)$ will be:

$$FT_1\left(\mathcal{S}\right) = \sum_{j=1}^{m} FS_1\left(\mathcal{C}_j\right). \tag{7}$$

Obviously $FS_1\left(g, \{L\}\right)$ reaches its minimum when all nodes in g, in all of its layers, have less external degrees than internal degrees which is one of the important aspects of communities, therefore minimizing $FT_1\left(\mathcal{S}\right)$ satisfies this aspect.

The second objective function should instead focus on maximizing the internal degrees. This is equal to determining the membership strength of a node to its community. The membership strength of node i to community g, is proportional to the maximum edges of node i to community g. So the average membership of node $i \in g$ in layer l' is:

$$f_2\left(g, l'\right) = \frac{\sum_{i\in g}\left(\frac{k^{in}_{i,l'}(g)}{|g-1|}\right)^{\beta}}{|g|} \times e_g, \tag{8}$$

Obviously, the nodes with more internal edges, are more important for the community because of their strong membership, and they accelerate the process of community detection. Hence, β in Eq. (8) magnifies the weight of the nodes with more links to the community. Also, e_g represents the number of edges in the community g. As mentioned above, $f_1\left(g, l'\right)$ evaluates community g in a single layer l', but to evaluate g in a subset of layers $\{L\}$, as a multilayer community we define $FS_2\left(g, \{L\}\right)$. Therefore, the second objective function for multilayer community detection $(g, \{L\})$ is:

$$FS_2\left(g, L\right) = \frac{\sum_{l'\in L} f_2\left(g, l'\right)}{|L|}, \tag{9}$$

To evaluate the partition $\mathcal{S} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$ as $\mathcal{C}_i = (g_i, \{L\}_i)$ in a multilayer network, will be:

$$FT_2\left(S\right) = \sum_{j=1}^{m} FS_2\left(C_j\right), \tag{10}$$

Therefore, optimizing both objective functions, will lead to communities which will have dense internal links and sparse external links in all the layers they are present.

Nevertheless, if we just consider the above functions, the algorithm will be biased towards detecting the communities in just one layer, which is similar to the aggregation methods. But we need a solution to find more layers for contribution. In order to guide the search process to find communities in more than one layer, we use the concept of *entropy* in the way used in [25]. Entropy, measures the difference in the weights of a weight vector, in a way that, the greater the difference of the weights, the higher the entropy and vice versa. Assuming that w_{lg} is the weight of layer l in community g, the entropy of l in community g will be:

$$Entropy\left(W_g\right) = -\sum_{l=1}^{L} w_{lg}\log\left(w_{lg}\right), \tag{11}$$

Clearly, layers have different contributions in community formation, so if the weight of contribution of each layer in a community is considered, minimizing it leads us to communities in the layers with more contribution and less differences. The use of entropy can be considered as a motivation to detect communities in more layers. In addition, the average strength of node memberships in a community in layers represents the contribution of the layer in the community. Therefore, by adding the entropy, we can rewrite Eq. (10) as:

$$FT_2\left(S\right) = \sum_{j=1}^{m}\left(FS_2\left(C_j\right) + \lambda\sum_{l'\in\{L\}_j} F_2\left(g_j, l'\right)\log\left(F_2\left(g_j, l'\right)\right)\right). \tag{12}$$

The above function (12) has two parts, the first for measuring the internal density and the second part for the rate of entropy, where λ is the impact factor of entropy and is normally set to 1. Maximizing the $FT_2(S)$, can detect communities with high internal density and contributions from more layers.

With these explanations, the mathematical model for the multilayer community detection problem will be:

$$\begin{aligned} &min \quad FT_1(S) = \sum_{j=1}^{m}\left(\frac{\sum_{l'\in\{L\}_j} f_1(g_j, l')}{|\{L\}_j|}\right)\\ &max \quad FT_2\left(S\right)\\ &\quad = \sum_{j=1}^{m}\left(\frac{\sum_{l'\in\{L\}_j} f_2\left(g_j, l'\right)}{|\{L\}_j|} + \lambda\sum_{l'\in\{L\}_j} f_2\left(g_j, l'\right)\log\left(f_2\left(g_j, l'\right)\right)\right)\\ &subject\ to \qquad x_i \in \sum_{l\in L}\Gamma^l_i,\\ &\qquad\qquad\quad L \subseteq D. \end{aligned} \tag{13}$$

Also for the constraint in formula (13):

$$\begin{aligned} N_{x_i} &= \left(\sum_{l\in D}\Gamma^l_i \cap \Gamma^l_{x_i}\right)^{\mu},\\ p(x_i) &= \frac{N_{x_i}}{\sum_{x_i} N_{x_i}} \end{aligned} \tag{14}$$

By taking into account both functions at the same time, the search process leads to the appropriate direction. It is worth noting that

considering just one of the functions will cause wrong results. For example, optimizing only FS_1 minimizes the external links, which biases the search process to find connected components, which may lead to the entire network. On the other hand, optimizing just FS_2, because of the focus on the internal links, can return part of a clique as a community, therefore ending up with many small communities. Hence, considering both the objective functions simultaneously and optimizing them, due to the tradeoff between the objective functions, will result in more appropriate and realistic communities.

4. Proposed method

In this paper, we have defined a novel mathematical model as presented above, that is optimized in order to find overlapping multilayer communities. The proposed method for the detection of such communities, involves using all the Pareto optimal solutions gained from the optimization process. This will also speed up the overall algorithm and improve the quality of results. Although, for faster convergence of the optimization process, a probabilistic model for initialization of the population has been suggested.

4.1. Multi-objective optimization process

After defining the mathematical model, a multi-objective optimization method must be selected. There are various methods to optimize a multi-objective problem, such as, NSGA II [26], MOPSO [27], PESA II [28] etc. All of the methods are population-based solutions, which means a population of potential solutions are formed and initialized, then the algorithm repeatedly optimizes the population and leads to the optimal point(s).

4.1.1. Genetic representation

First of all, we need to represent an optimal solution as an individual. Different ways to encode community structure in individuals are proposed by researchers [29–31]. Here, we used the locus-based representation [30], because of the several advantages it has over the other method [19]. The advantages are:

1. The process of encoding/decoding communities can be done using a fast breadth first search
2. The genetic operators can be applied easily
3. It is flexible enough to apply any constraint to the search space
4. It is suitable to non parametric methods meaning, using locus-based there is no need to determine the number of communities in advance.

In locus-based representation there is an array of values for decision variables, where x_i is the value of the decision variable i and that i and its allele value x_i are in the same community. Locus-based representation and the process of encoding/decoding is explained in Fig. 3:

4.1.2. Initialization

Regarding the importance and relation of initialization to convergence speed, in order to determine the allele value of i, we try to use other information of node i, such as topological position of the node and its first and second hop neighbors. In this way, to restrict the search space, we apply the constraint $x_i \in \Gamma_i$, which means the value of the decision variable i must be chosen from the neighbor set of i. This constraint prevents the search process from diverging from the optimal point. Yet for faster convergence, and in order to choose a suitable neighbor, we take into account a lightweight local search, formally:

$$\mathcal{N}(x_i) = \left(\Gamma_i \cap \Gamma_{x_i}\right)^{\mu}, i = 1, 2, \ldots, n,$$
$$p(x_i) = \frac{\mathcal{N}(x_i)}{\sum_{x_i} \mathcal{N}(x_i)} \tag{15}$$

which increases the probability of choosing the neighbors with more neighbors in common to staying in same community. To magnify the effect of the common neighbors, the factor μ is used, which is normally set to one.

Although there are different ways to do local searches between neighbors [7,20,32–34], the method used here is simple and efficient.

4.1.3. Cross-over

Cross-over is one of the operations of genetic algorithms which exchanges the information of two or more selected individuals/nodes. It is worth noting that the cross-over does not violate community structure. In this case, double point cross-over has been chosen, where two parents are chosen randomly as well as two random numbers i,j, and all the informational parts from these points are exchanged together.

4.1.4. Mutation

Another key operation of genetic algorithms used as a population-based optimization method, is mutation, which randomly changes some of the values and tries to generate new individuals by exploring new spaces, in search of optimal points. Obviously, mutation must be done with regards to the constraints in the formula (15). However, in order to prevent useless exploration, we performed some heuristics such as node memberships in communities.

4.2. Multilayer overlapping communities

Most of the methods presented for community detection, partitions the network under study, which means they aim to assign a node to just one community. But in real networks, communities can overlap and a node is often shared between communities [35]. In other words, a node can simultaneously be a member of several groups. For example, a person can be a member of the community g_1, which is formed in layers Facebook and YouTube, and at the same time is a member of the community g_2 that is made up in layers Facebook and Twitter (based on our definition, overlapping communities can be formed in the same layers or not). Therefore, communities can overlap in nodes and layers. So, for overlapping community detection, we search for a cover instead of partitioning the network. In [36] and [37] authors emphasize that overlapping communities are one of the most important features in real world networks analysis. Overlapping community detection is a challenging problem because the overlapping part of networks are often denser and algorithms are trapped in these areas [38], so they usually return the dense part as a community due to its high density compared to the surroundings. By reviewing the literature in this field, it has become clear that there are no well-known definitions for overlapping communities in multilayer networks. So initially, a definition of an overlapping community in a multilayer network must be given. In Section 2, we presented a community as $\mathcal{C} = (g, L)$ which $g \subseteq \mathcal{V}$ and $L \subseteq \mathcal{D}$. In a cover, two communities can share their nodes, but to be considered as an overlapping community, they must have the same set of layers.

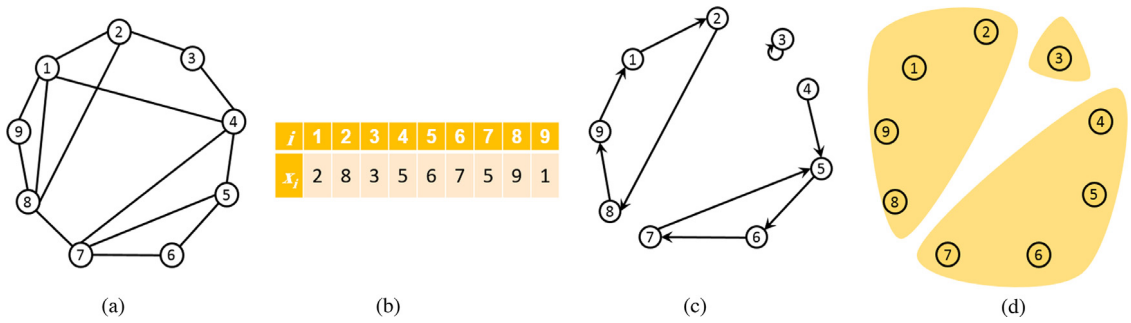

Fig. 3. (a) A simple 9-node network, (b) Locus-based representation, (c) Corresponding graph, (d) Detected Communities.

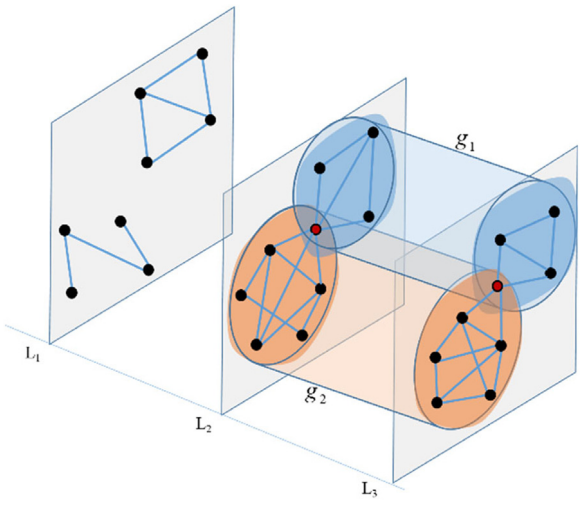

Fig. 4. Multilayer overlapping communities. Subset of nodes, which have dense connections in the same subset of layers and have node x_1 in common.

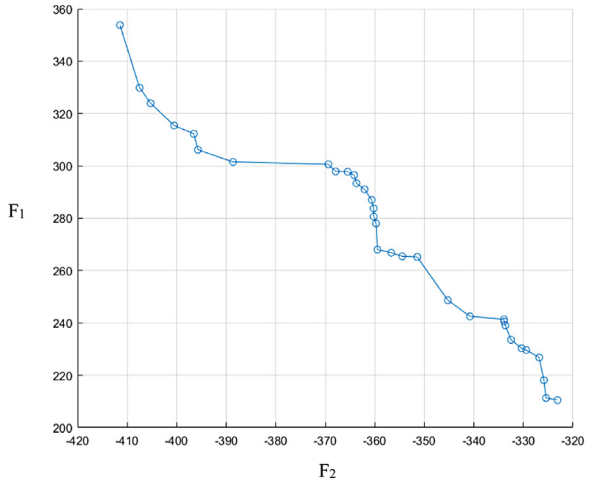

Fig. 5. Pareto front of the optimization process that contains pareto optimal solutions.

Definition 4.1 (*Multilayer Overlapping Communities*)**.** The communities $\mathcal{C}_1 = (g_1, L_1)$ and $\mathcal{C}_2 = (g_2, L_2)$ are multilayer overlapping communities if: $\{L_1 = L_2 \wedge g_1 \cap g_2 \neq \varnothing\}$. Therefore, even if two communities have nodes in common, but in different layers because of the conceptual difference, they are not overlapping communities.

Two simple overlapping communities in a multilayer network are shown in Fig. 4.

As discussed in Section 4.1.1, nodes are considered as decision variables and use locus-based genetic representation, which assigns a community membership for each node. Clearly, because of the space limitation, we cannot assign two community memberships to a single node, and we are not able to detect overlapping communities because of the constraint. In single layer networks, Zhao et al. [21] proposed a link clustering method so they considered links as decision variables. we believe in this method due to the multiplying the number of decision variables, the algorithm convergence and running time would be a problem. To resolve this problem, we extend our algorithm in [19] to multilayer networks, and a new algorithm to choose the appropriate communities from the Pareto optimal solutions.

In multi-objective optimization, due to the trade-off between objective functions, rarely we have a complete solution. Usually the optimization results in a Pareto front that consists of two or more Pareto optimal solutions, which cannot dominate each other in terms of different objective functions. Fig. 5 shows the Pareto front, which is the result of the multi objective optimization process. Each point in the figure represents a partition of the network.

4.3. Proposed algorithm for overlapping community detection

In the second phase of the proposed algorithm, we take into account the communities instead of the partitions (optimal solution). Therefore, we define the concept of dominance in the communities' level.

Definition 4.2 (*Dominated Multilayer Community*)**.** Assume communities $\mathcal{C}_1 = (g_1, L_1)$ and $\mathcal{C}_2 = (g_2, L_2)$, we say $\mathcal{C}_1$ dominates $\mathcal{C}_2$ and we show it as $\mathcal{C}_1 \prec \mathcal{C}_2$ if and only if: $g_1 \preceq g_2$, $L_1 = L_2$ and $\forall i; f_i(\mathcal{C}_1) \leq f_i(\mathcal{C}_2)$

Using the definition, we design a three-step algorithm to refine and select the communities from the Pareto optimal solutions. The proposed algorithm has four steps:

1. **Refinement**: In this phase, all the repeated or dominated communities need to be eliminated. Different optimal solutions will possibly produce some redundant communities. Using the definition above, we can eliminate the redundant communities.
2. **Selection**: select communities in an ordered manner to get a cover.
3. **Post processing**: eliminating the communities, which have no effect on the coverage.

The algorithm, takes the Pareto front as input and by applying the above mentioned steps, produces communities which will be a cover for the network under study.

The pseudo code of the algorithm is shown below with some details:

Algorithm 4-1- Overlapping multilayer community selection

Parameters: *MaxGen,PopSize,Pcrossover,Pmutation*
Input: *The adjacency matrix (A) of the multilayer network*
Output: *overlapping multilayer communities (C)*

Phase 1: multi-objective optimization
Initialization:
Layers initialization : L={l_1,l_2,...,l_{pop}}
Create neighbors matrix
Position initialization regards to neighbors matrix: P={p_1,p_2,...,p_{pop}}
Calculate cost function: F={??}
[pop]=Nondominated Sorting(pop)
[pop]=Calculate Crowding Distance(pop)
[pop]=Sort population(pop)
Cycling:
*For i=1:popsize*pcrossover*
    Randomly select two parent as p_1 and p_2
    Popc=double point crossover(p_1,p_2)
    Pop=pop+popc;
End
*For i=1:popsize*pmutation*
    Randomly select an individual as p_m
    Popm=roulettweelmutation(p_m)
    Pop=pop+popm
End
[PFs,pop]=nondominated sorting(pop)
[PFs,pop]=Calculate Crowding Distance (pop)
Pop=pop(1:npop); truncate
ParetoFront=PFs(1);
end cycling

phase 2: Select Communities
Input: ParetoFront
Output: final multilayer overlapping communities
    AllCommunities= all unique communities from ParetoFront
Calculate cost function for each community
    For every communities $\mathcal{C}_1 = (g_1, L_1)$ *and* $\mathcal{C}_2 = (g_2, L_2)$
    if ($g_1 \subseteq g_2$ *and* $l_1 = l_2$ *and* $\mathcal{C}_2 \prec \mathcal{C}_1$*)*
        Delete community $\mathcal{C}_1$
    End
    Sort AllCommunities based on F_2 and F_1 –Ascend
    While all nodes to be covered
        Select Communities from AllCommunities and add to FinalCommunities
    End
    Delete all communities from FinalCommunities, which have Minimum effect on F_2 and do not violate coverage
End

4.4. Computational complexity

Our approach has three main parts. First, initialization, if n, l and Pn denote the number of nodes, number of layers and population size respectively, for the first phase, the computational complexity is O(nlP_n). In the second part, for a complete analysis of the computational complexity, we can say that a significant part of the computational complexity is related to the calculation of objective functions, which depends on the number of nodes (decision variables). Its upper bound is of order $O(n^2)$ and lower bound is of order $O(n)$.

Computational complexity of NSGA-II in [39] has been considered as $O(n_g n_o P_n^2)$, in which n_g is number of generations, n_o is number of objective functions and P_n is population size. According to the mathematical model $n_o = 2$. Also, due to the fact that in this case the stop condition is fixed and does not determine dynamically, the number of generations is fixed and $n_g = 5$. It should be noted that depending on the conditions of the problem under study, n_g can be NP. However, due to the operation of the second phase of the algorithm (community selection); the stop conditions in the proposed algorithm are considered constant and in Section 5.4 the efficiency in different values for n_g is discussed. Therefore, the computational complexity of the algorithm, in optimization phase, in the worst case is of order $O(n^2{P_n}^2)$ and in the best case is of order $\Omega\ (n{P_n}^2)$.

For the third part, community selection, if we denote the number of communities as m, since we map the communities in a tree and two pass traverse are needed for preprocessing and community selection, then it's computational complexity is $O(mlogm)$. So the total computational complexity will be $O(nlP_n + n^2{P_n}^2 + mlogm)$. Since m << n this can be reduced to $O(nlP_n + n^2{P_n}^2)$ and $\Omega\ (nlP_n + n{P_n}^2)$.

Also to evaluate the efficiency, we run the proposed algorithm on a laptop with core i5 CPU and 8 GB of RAM, on a mid-size network with 1000 nodes and more than 15,000 edges, the algorithm takes even less than 13 s to detect the communities.

5. Experimental results

To evaluate the proposed method, the effectiveness of the proposed algorithm is compared with different methods on both synthetic and real network datasets.

5.1. Evaluation methods

In order to evaluate the results, we used two well-known measures, normalized mutual information (NMI) [40] and multiplex modularity [14]. To have a comprehensive evaluation on the proposed algorithm, we consider two types of networks. First, is the network with ground truth communities. The NMI measure is used when the ground truth of network is available. NMI, measures the similarity of the two sets of nodes. Formally:

$$I(P_1, P_2) = \frac{-2\sum_{i=1}^{C_{S_1}}\sum_{j=1}^{C_{S_2}} N_{ij}\log\left(\frac{N_{ij}n}{N_i N_j}\right)}{\sum_{i=1}^{C_{S_1}} N_i \log\left(\frac{N_i}{n}\right) + \sum_{j=1}^{C_{S_2}} N_j \log\left(\frac{N_j}{n}\right)}. \quad (16)$$

where there are two partitions P_1 and P_2 in the network, and N_{ij} represents the nodes in common between community i of partition P_1 and community j of partition P_2. Also $N = [N_{ij}]$ is the confusion matrix and N_i and N_j are sums of the row i and column j of the confusion matrix N. Furthermore, C_{S_1} and C_{S_2} are the number of partitions of P_1 and P_2 and n is the number of nodes.

The second measure is multiplex modularity. The modularity measure given by Newman and Girvan [13], is one of the most accepted measures of evaluation for community detection methods. In fact, modularity, comparing to a random network, as a null model, measures the significance of community structure, which implies how dense the detected communities are. Mucha et al. [14] extended this modularity to multilayer networks which is also discussed here in Section 2.

5.2. Synthetic networks

To evaluate our method, we first run the algorithm on synthetic networks. The results of algorithm on synthetic networks are of great importance because synthetic networks are generated based on some adjustable parameters and thus we can produce a range of different networks with different characteristics. This can provide a better understanding of how the algorithm behaves on different networks.

5.2.1. Multilayer benchmark networks

(Lancichinetti & Fortunato, Benchmark for testing community detection algorithm on directed and weighted graphs with overlapping communities, [41]) introduced LFR synthetic networks to solve the problem of simple GN networks [13]. LFR networks are power law in degree distribution and community size; also, it allows communities to overlap, which makes LFR networks a closer representation of real networks in terms of structural parameters. Lancichinetti also extends the well-known LFR networks to multilayer [42,43] and introduced a method which runs the original LFR benchmark algorithm several times and slices the original graph in multiple layers. To evaluate the performance of the proposed algorithm, we generated a multilayer network with 3 layers and 1000 nodes. Other parameters for the generated synthetic networks are listed below in Table 1:

Table 1
Parameters for the generation of synthetic networks.

Parameter	Value	Description
-Orig	1	Number of original graphs
L	3	Number of layers
N	1000	Number of nodes
K	15	Average node degree
MaxK	50	Max node degree
μ	0-0.8:0.05	Mixing parameter
τ_1	2	Negative exponent for the degree sequence
τ_2	1	Negative exponent for the community size distribution
MinC	20	Minimum community size
MaxC	50	Maximum community size
ON	0-200:40	Number of overlapping nodes
OM	1–4	Number of memberships of the overlapping nodes

By varying μ, *OM* and *ON*, we generate 408 different networks, yet due to space limitations here, we only present a few selection of the corresponding results. The results for the rest of the networks are similar and the charts given below are completely representative.

We compared the results with three well-known multi-layer community detection methods: GenLouvain [44], mInfomap [42, 43] and PMM [7], lcd [45] and MultiMOGA [22]. The last algorithm was executed for only the OM = 1 due to its long runtime. As can be seen in Figs. 6 and 7, generally, when $\mu \leq 0.65$, our method dominates the other methods.

5.3. Real networks

Recently, researchers have collected some real multilayer networks data as well as some useful applications for multilayer analysis (De Domenico, Porter, & Arenas, MuxViz: a tool for multilayer analysis and visualization of networks, [42,43]). To show the performance of the proposed algorithm on real data, the following real multilayer network[2] datasets have also been chosen.

Florentine: Describes the relations among politically prominent families in the city of Florence [46]. There are two layers (relations): business ties and marriage alliances. Two factions appear in the data, with families strongly related to the Medici or to the Strozzi family, which makes the data useful for community detection methods.

Monastery: Top three choices of a group of monks on four pairs of positive/negative relations was asked [47]. Therefore, the layers are esteem/disesteem, liking/disliking, positive/negative influence, and finally, praise/blame.

Bankwiring: The data describes employees that work in the bank wiring room [48],where they have different roles. The layers are participation in horseplay; participation in arguments about open windows; friendship; antagonistic (negative) behavior; helping others with work, and the number of times workers traded job assignments.

Tailorshop: This data represents work and friendship interactions among workers in a tailor shop [49]. Two versions of the social network that are recorded in different times are available.

Aucs: This data was collected at a University research department [50] And looks at the relationships and interactions among

[2] The data are available at: http://multilayer.it.uu.se/datasets.html.

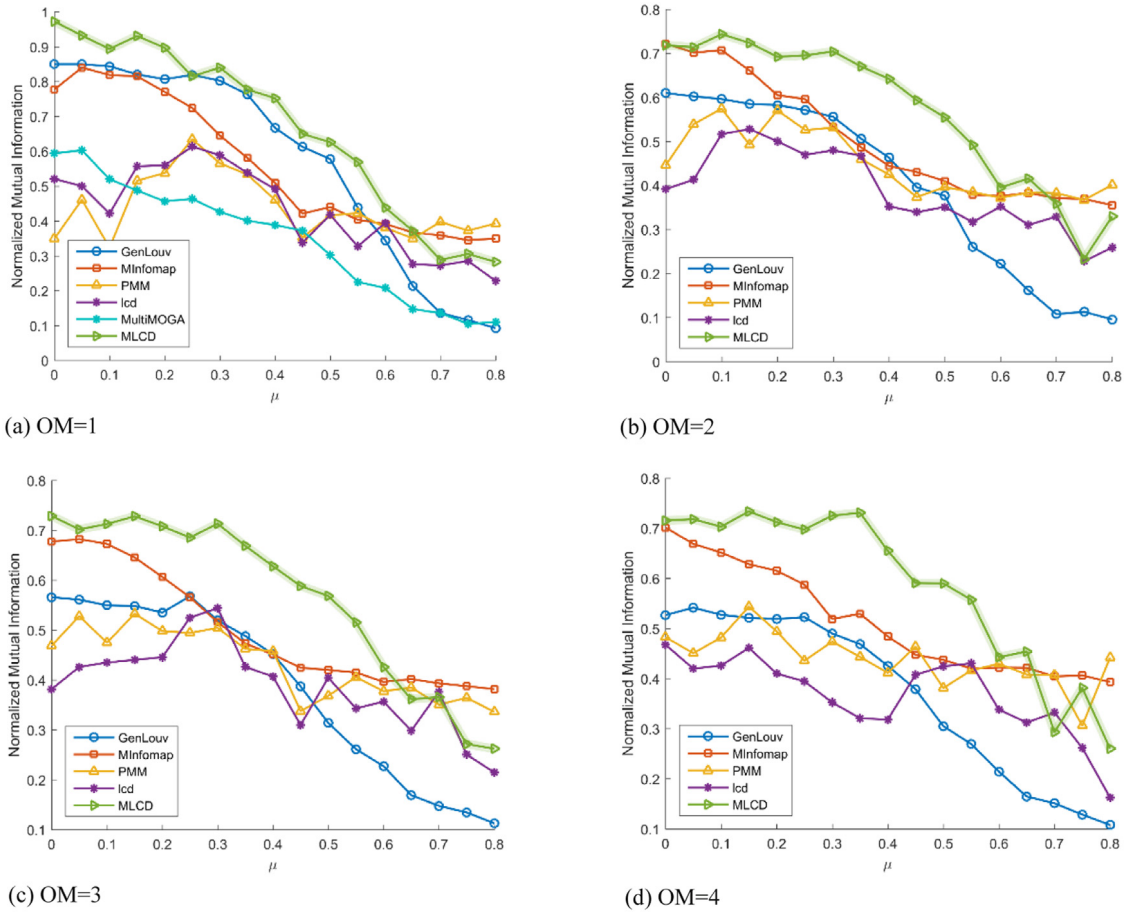

Fig. 6. Comparison of quality of the results in terms of NMI on synthetic networks, each chart represents the results for different number of memberships (OM = [1–4]) in different mixing parameters.

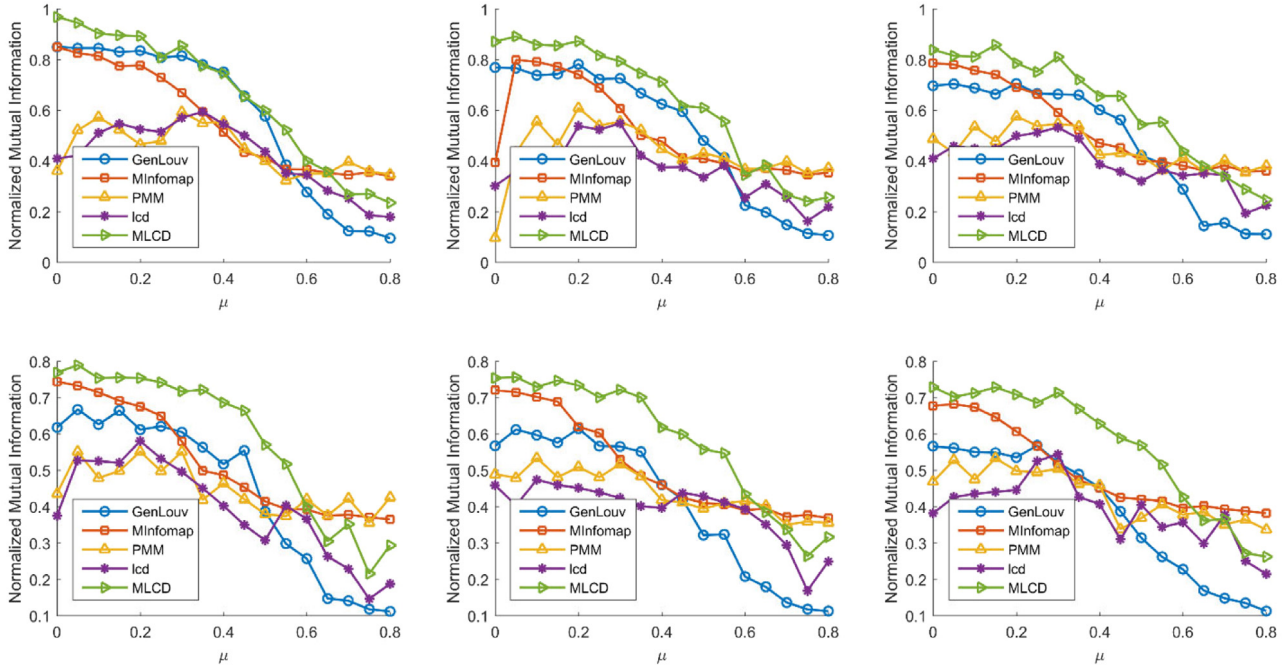

Fig. 7. Comparison of quality of the results in terms of NMI on synthetic networks, each chart represents the results for different number of memberships (ON = [1–6]) and OM = 3 in different mixing parameters μ.

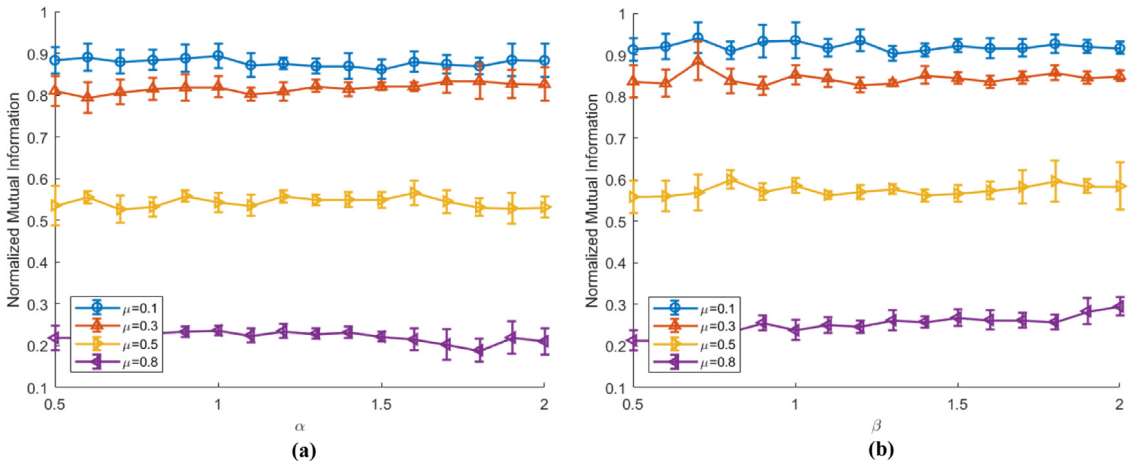

Fig. 8. The effect of α (a) and β (b) on *NMI* in mLFR benchmark networks. Error bars show the standard deviation for 10 runs of the algorithm. The parameters are $N = 1000$, $L = 3$, $OM = 2$, $ON = 40$.

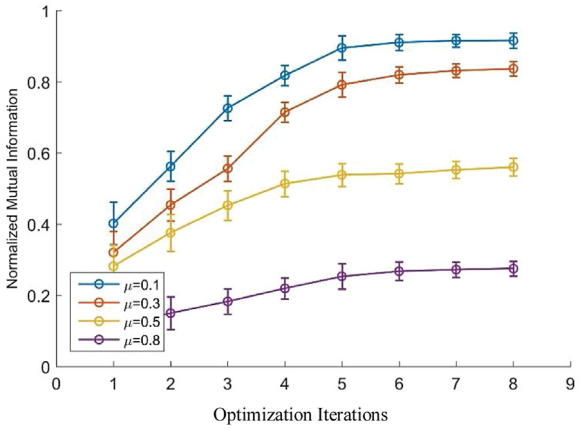

Fig. 9. NMI and optimization iterations for 10 runs and error bars.

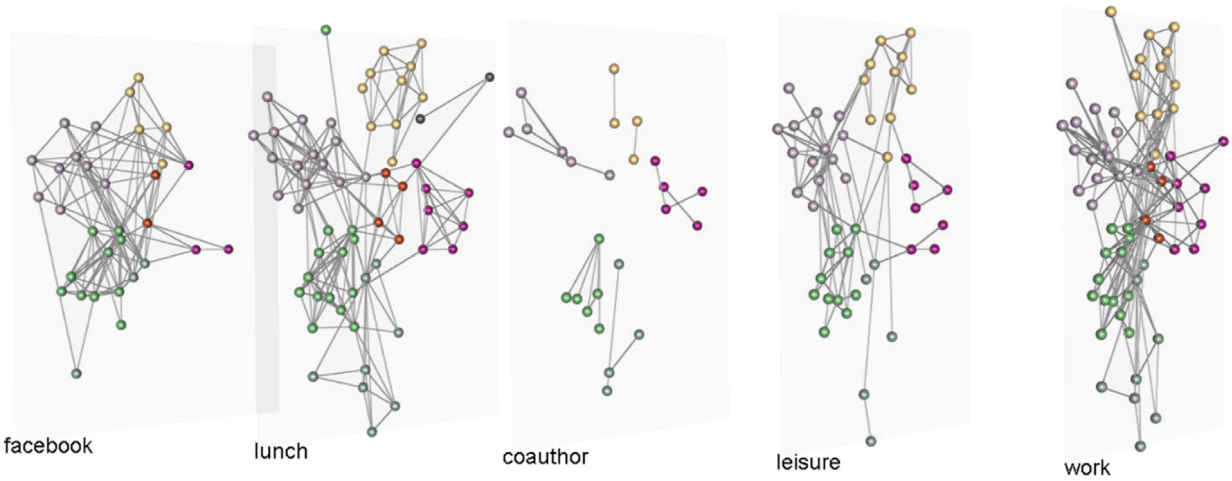

Fig. 10. Layers, nodes and multilayer communities in Aucs dataset.

professors, postdoctoral researchers, Ph.D. students and administration staff. The layers are Facebook, lunch, coauthor, leisure, work.

Ff-Tw-Yt: This anonymized dataset has been obtained starting from Friendfeed, a social media aggregator [51]. In this system, users can directly post messages and comments on other messages much like in Facebook and other similar OSNs. From these, a multilayer network was retrieved, with users who their Twitter and YouTube accounts was associated to exactly one account.

Friendfeed: Friendfeed is a large anonymized real dataset containing public interactions among users of Friendfeed collected

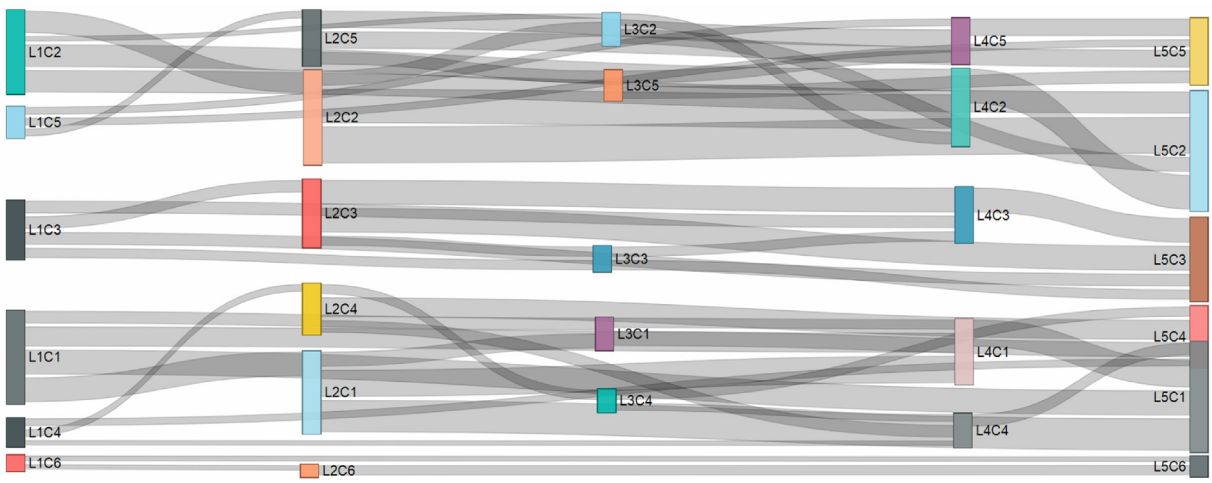

Fig. 11. This diagram represents the layers, number of communities, size of communities, multilayer communities as well as the flow of nodes' memberships between multilayer communities in Aucs dataset.

Table 2
Real datasets and results of multilayer community detection methods in terms of multilayer modularity.

DataSet	Layers	Nodes	Edges	MLCD	GenLouv	mInfomap	PMM
Florentine	2	15	35	**0.591**	0.530	0.548	0.456
Monastery	10	18	510	**0.416**	0.411	0.241	0.241
Bankwiring	6	14	110	0.526	0.526	**0.919**	0.40
Tailorshop	4	39	552	**0.510**	0.358	0.174	0.174
Aucs	5	61	620	0.590	0.633	**0.826**	0.372
Ff-Tw-Yt	3	6 407	74 862	**0.890**	0.497	0.286	0.238
Friendfeed	3	510,896	20,330,701	**0.629**	0.582	0.482	na

Table A.1
Structural measures and their formula for multilayer networks.

Measure	Formula	Description
NodeSetCov	$NodeSetCov\ (\mathcal{N}) = \frac{\sum_{i\in\|\mathcal{L}\|}\left(\left\|V_{L_i}\right\|/\|\mathcal{V}\|\right)}{\|\mathcal{L}\|}$	Average ratio of nodes in layers
EdgeSetCov	$EdgeSetCov(\mathcal{N}) = \frac{\sum_{i\in\|\mathcal{L}\|}\left(\|E_i\|/\sum_i\left\|E_{L_i}\right\|\right)}{\|\mathcal{L}\|}$	Average ratio of edges in layers
AvgDeg	$Av5gDeg(\mathcal{N}) = \frac{\sum_{L\in\mathcal{L}}\left(\frac{\sum_{i\in V_L}\deg_i}{\|V_L\|}\right)}{\|\mathcal{L}\|}$	Average degree of nodes in layers
AvgPathLength	$AvgPathLen(\mathcal{N}) = \frac{\frac{1}{N(N-1)}\sum_{i\neq j}d(v_i,v_j)}{\|\mathcal{L}\|}$	Average path length in layers
AveClustCoeff	$AvgCC(\mathcal{N}) = \frac{\|\mathcal{L}\|\left\|\left\{e_{jk}:v_j,v_k\in N_i,e_{jk}\in E\right\}\right\|}{\|N\|k_i(k_i-1)}$	Average clustering coefficient in layers
AveDensity	$AveDensity(\mathcal{N}) = \frac{\sum_{L_i\in\mathcal{L}}\frac{2\left\|E_i\right\|}{(N\times(N-1))}}{\|\mathcal{L}\|}$	Average edges density in layers
FlattedDensity	$FlattedDensity(\mathcal{N}) = \frac{\left\|\cup_{L_i}E_{L_i}\right\|}{\|\mathcal{V}\|\times\|\mathcal{V}-1\|}$	Density of the flattened network
AveNodeDegInLayers	$AveNodeDeg(\mathcal{N}) = \sum_{j\in\mathcal{V}}\frac{\left(\sum_{L_i}\deg(v_j^{L_i})\right)/\|\mathcal{L}\|}{\|\mathcal{V}\|}$	Average degree of a node in layers
AveNodePresInLayers	$NodePres\ (\mathcal{N}) = \sum_{i\in\mathcal{V}}\frac{\sum_{l\in\|\mathcal{L}\|}(l\|v_i)}{\|\mathcal{L}\|}$	Average number of Layers in which a node is present

over two months. It contains three layers, which are commenting, liking and following, so it defines a contact networks based on different user interactions and forms a multilayer social network.

In Appendix, some structural measures and their formula for better understanding of the datasets are described (see Tables A.1 and A.2).

In Table 2, the results of four multilayer community detection algorithms in terms of multilayer modularity are compared and the best results are shown in **bold**. As can be seen, in four out of the six datasets, our proposed method MLCD (i.e. multilayer community detection) performs better.

5.4. Parameter analysis

We study the effect of various α and β on the performance of the proposed method. In Fig. 8(a) and (b) average *NMI* of 10 runs of the method, for various α and different μ in *mLFR* networks has been shown. As can be seen by varying α and β, the algorithm keeps its stability. As we know because of the nature of meta heuristic methods (such as *NSGA II*) the results of multiple runs are not necessarily the same, but the second phase of the algorithm makes the results more stable.

In Fig. 9, the mean NMI is shown for 10 runs as well as error bars in various repetitions of the optimization method, NSGA-II for different networks. As can be seen, in about five repetitions,

Table A.2
Main characteristics of the multilayer networks.

Name	Layers	Nodes	Edges	Nod-Set-Cov	Nod-Set-Cov-StD	Edge-Set-Cov	Edge-Set-Cov-StD	Ave-Deg	Ave-Deg-StD	Ave-Path-Len	Ave-Path-Len-StD	Ave-Clust-Coeff	Ave-Clust-Coeff-StD	Ave-Density	Ave-Density-StD	Flatted-Density	Ave-Nod-Deg-In-layer	Ave-Nod-Deg-In-Layer-StD	Ave-Node-Pres-In-Layers	Ave-Node-Pres-In-Layers-StD
Florentine	2	15	35	0.8667	0.2886	0.5	0.101	0.9444	0.786	2.45	0.05	0.0611	0.0079	0.4306	0.0196	0.1286	2.3333	1.9881	1.2667	0.8837
Monastery	10	18	510	0.9722	0.054	0.1	0.0131	2.8331	0.397	1.99	2.385	0.0685	0.0278	0.4753	0.1797	0.6895	28.3333	10.6826	8.7222	1.4061
Bankwiring	6	14	110	0.7381	0.133	0.1667	0.0684	1.5071	0.8687	1.9	0.3	0.0833	0.0732	0.9471	0.7046	0.3626	7.8571	5.7893	3.2857	1.7728
Tailorshop	4	39	552	0.9615	0.0491	0.25	0.1207	3.695	1.6539	2.125	0.2586	0.0487	0.0306	0.2628	0.1066	0.224	14.1538	8.4216	3.3333	0.8983
Aucs	5	61	620	0.7344	0.2621	0.2	0.1184	5.2149	2.4564	2.44	0.6468	0.5028	0.1707	0.124	0.0725	0.1929	20.3279	11.3118	3.6721	0.9785
Ff-tw-yt	3	6 407	74 862	0.6194	0.447	0.3333	0.29	5.4029	4.1574	4.18	1.27	0.0474	0.0235	0.0062	0.0023	0.0016	11.6844	56.165	1.2933	0.8051
Friendfeed	3	21,006	573,600	0.4305	0.4914	0.3333	0.4614	23.9563	16.6429	10.59	3.432	0.1071	0.1191	0.0123	0.0016	0.0012	27.3065	107.9762	0.6619	0.9117

the results are converged and the error of the results is reduced. Therefore, the number of optimization iterations (n_g) is set to five for all networks.

6. Discussion and results analysis

To generate mLFR networks with overlapping communities, there are two main parameters: *ON* represents the number of overlapping nodes and *OM* specifies the number of memberships that a node can have (i.e. the number of communities a node can belong to). In addition, another important parameter in mLFR networks generation is the mixing parameter μ, which is the ratio of a node's in-degree to out-degree to the community it belongs. Obviously, increasing μ, communities becomes more complex and interwoven so the process of community detection becomes more complicated. In other words, in networks with $\mu > 0.5$ there are no multilayer strong communities as given by Definition 3.2. This is why in defining the objective functions (Section 3.1), we have considered multilayer weak community as the base.

With these explanations, Figs. 6 and 7 shows that, increasing OM and ON, the performance of the algorithm is improved with greater distance, indicating that increasing the amount of overlaps, the algorithm's operating distance is increased. This difference is due to the definition of overlapping multilayer communities and its focus on designing a mathematical model and the proper operation of the second part of the algorithm, which selects communities from the Pareto front.

At $\mu < 0.2$, because the community structure of the networks is quite clear and the density of communications between the communities is low, naturally, the algorithms have proper results, but with increasing μ the difference between the results of the algorithms is more pronounced.

Also, in all mLFR networks with $\mu < 0.7$ the proposed algorithm outperforms the other methods, and when $\mu > 0.5$, the performance of all the algorithms drop.

At $\mu > 0.7$, due to the enormous complexity of the network and the entanglement of the communities as well as and the lack of community structure, the algorithms show more fluctuations. This is because in these circumstances, often only overlapping sections of communities are identified as dense communities, while they are merely part of larger communities.

In addition, when the communities are not overlapping, i.e. $OM = 1$, although our proposed method (MLCD) still performs better, but the results are close to other methods like *GenLouvain* and *mInfomap*. Instead, in multilayer overlapping communities ($OM = [2..4]$), because of the emphasis on overlapping communities, the proposed algorithm again performs much better. This shows that the proposed novel method works correctly in trying to find overlapping communities in multilayer networks. It should also be noted that, uniform reduction of the NMI with increasing μ in Figs. 6 and 7 shows that the results can also be generalized to other networks with different *OM* and *ON*.

The stability of the results against the changes over the parameters as well as the convergence of results in low repetitions of optimization is another advantage of the proposed method. The reason for the stability of the results is the second phase of the algorithm (community selection). An *ideal* Pareto solution contains *appropriate combination* of *dense communities*. On the other hand, each Pareto optimal solution contains a number of dense communities, but does not necessarily have the proper combination of them. Therefore, the major part of the attempt of the optimization process is to find the *proper combination* of communities. When we focus on the choice of communities rather than choosing a Pareto solution (partition), the effect of parameter changes as well as the difference in repetitions is largely neutralized. This causes the selection of the proper communities regardless of which Pareto solution they belong to.

In addition, in Figs. 10 and 11, the communities of the dataset *Aucs* are visualized. In order to have a comprehensive understanding of the results, as can be seen in Fig. 10, the multiyear communities include different number of nodes in the contributing layers. Also, the layers in a multilayer network can have different number of nodes. In Fig. 11, layers, nodes, size of the communities and nodes' flow between layers (to organize a multilayer community) are explicitly shown. This completely matches our definition for multilayer community that emphasizes communities form in a subset of layers.

7. Conclusion

In addition, in this paper a new definition of multilayer community is presented which provides more accurate and meaningful results in comparison with other methods. According to Definition 3.1, despite approaches like flattening, ensemble based and so on, communities are formed in different layers, which eliminates the limitation of presence in all layers for a community. It is important to note that all nodes in a multilayer community not necessarily have dense connections in all layers, and they may even not have any connections in some layers. In this case, if we consider all the layers, it will not only distort the community detection process, but also produces results which are not necessarily meaningful and correct.

Finally, it is worth noting that the proposed idea can be applied to many other applications apart from community detection, as it is a general approach to solve problems that consist of sub-problems. In other words, in problems where the final solution should include the appropriate combination of the following sub-problems, the proposed idea can be applied. Although multi-objective optimization methods generally produce a set of Pareto optimal solutions (Pareto front), here, we get one solution which has been created from the proper combination of the sub-problems (however, because they have been chosen independently, they may also overlap). For example, in the case of community detection in networks, in addition to finding dense communities, the results should also contain the proper combination of communities. Finally, our study may also be extended by fuzzy based method to determine node's memberships in multilayer networks and it is also interesting to evaluate the performance of the proposed algorithm on larger networks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Mostafa Salehi was supported in part by a grant from IPM, Iran through project No. CS1398-4-161.

Appendix

To have a better insight of multilayer networks, different measures are proposed by researchers. In this appendix, some structural measures of real networks, which have been used in this paper, are presented (see Tables A.1 and A.2).

References

[1] S. Wasserman, F. Katherine, Social Network Analysis: Methods and Applications, Cambridge University Press, Cambridge, 1994.
[2] M.E. Newman, Networks: An Introduction, Oxford University Press, Oxford, 2010.
[3] C. De Bacco, E. Power, D. Larremore, C. Moore, Community detection, link prediction, and layer interdependence in multilayer networks, Phys. Rev. E 95 (4) (2017) 042317.
[4] P. Moradi, S. Ahmadian, F. Akhlaghian, An effective trust-based recommendation method using a novel graph clustering algorithm, Physica A 436 (2015) 462–481.
[5] Z. Wang, Y. Wu, Q. Li, F. Jin, W. Xiong, Link prediction based on hyperbolic mapping with community structure for complex networks, Physica A 450 (2016) 609–623.
[6] W. Zhixiao, Z. Li, G. Yuan, Y. Sun, X. Rui, X. Xiang, Tracking the evolution of overlapping communities in dynamic social networks, Knowl.-Based Syst. 157 (2018) 81–97.
[7] L. Tang, X. Wang, H. Liu, Uncoverning groups via heterogeneous interaction analysis, in: IEEE International Conference on Data Mining ICDM'09.. 9, 503-512. Ninth IEEE International Conference on, 2009.
[8] M.E. Dickison, M. Magnani, L. Rossi, (2016). Multilayer social networks. Cambridge University Press.
[9] S. Kelley, M. Goldberg, K. Mertsalov, M. Magdon-Ismail, W. Wallace, Overlapping communities in social networks, Int. J. Soc. Comput. Cyber-Phys. Syst. 1 (2) (2011) 135–159.
[10] F. Reid, N. Hurley, Diffusion in networks with overlapping community structure, in: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (pp. 969-978), IEEE Computer Society, 2011, pp. 969–978.
[11] M. Berlingerio, M. Coscia, F. Giannotti, Finding and characterizing communities in multidimensional networks, in: Advances in Social Networks Analysis and Mining, IEEE, 2011, pp. 490–494.
[12] N. Tehrani, M. Magnani, Partial and overlapping community detection in multiplex social networks, in: International Conference on Social Informatics (pp. 15-28), Springer, Cham, 2018.
[13] M. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[14] P. Mucha, T. Richardson, K. Macon, M. Porter, J. Onnela, Community structure in time-dependent, multiscale, and multiplex networks, Science 328 (5980) (2010) 876–878.
[15] M. Berlingerio, F. Pinelli, F. Calabrese, Abacus: frequent pattern mining-based community discovery in multidimensional networks, Data Min. Knowl. Discov. 27 (3) (2013) 294–320.
[16] Z. Kuncheva, G. Montana, Community detection in multiplex networks using locally adaptive random walks, in: Advances in Social Networks Analysis and Mining, IEEE\ACM, 2015, pp. 1308–1315.
[17] X. Wang, J. Liu, A layer reduction based community detection algorithm on multiplex networks, Physica A 24 (2017) 4–252.
[18] B. Amiri, L. Hossain, J. Crawford, R. Wigand, Community detection in complex networks: Multi–objective enhanced firefly algorithm, Knowl.-Based Syst. 46 (2013) 1–11.
[19] M. Ebrahimi, M. Shahmoradi, Z. Heshmati, M. Salehi, A novel method for overlapping community detection using multi-objective optimization, Physica A 505 (2018) 825–835.
[20] M. Gong, L. Ma, Q. Zhang, L. Jiao, Community detection in networks by using multiobjective evolutionary algorithm with decomposition, Physica A 391 (15) (2012) 4050–4060.
[21] Y. Zhao, S. Li, J. Feng, Overlapping community detection in complex networks using multi-objective evolutionary algorithm, Comput. Appl. Math. 36 (1) (2017) 749–768.
[22] A. Amelio, C. Pizzuti, Community detection in multidimensional networks, in: Tools with Artificial Intelligence, Vol. 26, IEEE, 2014, pp. 352–359.
[23] C. Pizzuti, A. Socievole, Many-objective optimization for community detection in multi-layer networks, in: IEEE Congress on Evolutionary Computation (CEC) (pp. 411-418). IEEE Congress on, 2017.
[24] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Natl. Acad. Sci. USA 101 (9) (2004) 2658–2663.
[25] J. Friedman, J. Meulman, Clustering objects on subsets of attributes, J. R. Stat. Soc. Ser. B Stat. Methodol. 66 (4) (2004) 815–849.
[26] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Ccomput. 6 (2) (2002) 182–197.
[27] C. Coello, G. Pulido, M. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.
[28] D. Corne, N. Jerram, J. Knowles, M. Oates, PESA-II: Region-based selection in evolutionary multiobjective optimization, in: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. 3, 283-290, Morgan Kaufmann Publishers Inc., 2001.
[29] A. Verma, K. Bharadwaj, Identifying community structure in a multi-relational network employing non-negative tensor factorization and GA k-means clustering, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 7 (1) (2017) 1–22.
[30] Y. Park, M. Song, A genetic algorithm for clustering, in: Proceedings of the Third Annual Conference on Genetic Programming, 1998, 1998, pp. 568–575.
[31] H. Xia, J. Zhuang, D. Yu, Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data, Pattern Recognit. 46 (9) (2013) 2562–2575.
[32] A. Lancichinetti, F. Radicchi, J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, PLoS One 6 (4) (2011) e18961.
[33] U. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
[34] W. Zhi-Xiao, L. Ze-chao, D. Xiao-fang, T. Jin-hui, Overlapping community detection based on node location analysis, Knowl.-Based Syst. 105 (2016) 225–235.
[35] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174.
[36] J. Xie, S. Kelley, B. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Comput. Surv. (csur) 45 (4) (2013) 43.
[37] X. Yu, J. Yang, Z.Q. Xie, A semantic overlapping community detection algorithm based on field sampling, Expert Syst. Appl. 42 (1) (2015) 366–375.
[38] J. Yang, J. McAuley, J. Leskovec, Community detection in networks with node attributes, in: IEEE International Conference on Data Mining (ICDM) (pp. 1151-1156), IEEE, 2013.
[39] D.M. Curry, C.H. Dagli, Computational complexity measures for many-objective optimization problems, Procedia Computer Science 36 (2014) 185–191.
[40] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. 2005 (09) (2005) P09008.
[41] A. Lancichinetti, S. Fortunato, Benchmark for testing community detection algorithm on directed and weighted graphs with overlapping communities, Phys. Rev. E 80 (1) (2009) 016118.
[42] M. De Domenico, A. Lancichinetti, A. Arenas, M. Rosvall, Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems, Phys. Rev. X 5 (1) (2015) 011027.
[43] M. De Domenico, M. Porter, A. Arenas, Muxviz: a tool for multilayer analysis and visualization of networks, J. Complex Netw. 3 (2) (2015) 159–176.
[44] L.G. Jeub, M. Bazzi, I.S. Jutla, P.J. Mucha, A generalized Louvain method for community detection implemented in MATLAB, 2011-2017.
[45] W. Liu, T. Suzumura, H. Ji, G Hu, Finding overlapping communities in multilayer networks, PloS one 13 (4) (2018) e0188747.
[46] D. Kent, The Rise of the Medici: Faction in Florence, Oxford University Press, Oxford, 1978.
[47] R. Breiger, S. Boorman, P. Arabie, An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling, Math. Psych. 12 (1975) 328–383.
[48] F. Roethlisberger, W. Dickson, Management and the Worker, Cambridge University Press, Cambridge, 1939.
[49] B. Kapferer, Strategy and Transaction in an African Factory, Manchester University Press, Manchester, 1972.
[50] L. Rossi, M. Magnani, Towards effective visual analytics on multiplex and multilayer networks, Chaos Solitons Fractals 72 (2015) 68–76.
[51] M. a. Magnani, The ML-model for multi-layer social networks, in: Advances in Social Networks Analysis and Mining (ASONAM) (pp. 5-12). International Conference on, 2011.

M.R.Shahmoradi received his B.S. degree in Computer Engineering from Tabriz University, Tabriz, Iran, in 2008 and M.S. degree in Information Technology from the Shiraz University, Shiraz, Iran, in 2010. He is currently PhD student of IT in University of Tehran, Tehran, Iran and he is a member of LEADS lab. His research interests include Complex networks, Data mining, Community detection in social networks, Decision support systems and Applications of MADM/MCDM in networks.

M. Ebrahimi is an Assistant professor of Applied Mathematics in Faculty of New Sciences and Technologies, University of Tehran, Iran. His research expertise is in mathematical modeling of engineering problems that have arisen in the fields of networked systems, information technology, decision sciences and health. He has been active in the combined numerical-probabilistic algorithm based on Monte Carlo methods for solving optimization problems. He has been active in research fields including multi-criteria decision-making methods and mathematical modeling for diffusion processes. He has published about 60 research papers. He has set up the Laboratory of Engineering and Analysis of Decision Support Systems (LEADS).

Z.Heshmati is an Assistant professor of Computer Engineering in Faculty of New Sciences and Technologies, University of Tehran, Iran. She received her B.S. degree in Electronics Telecommunications and Computer Engineering, University of Bradford, UK, 2003 and PHD Telecommunication Engineering, University of Leeds, Leeds, UK in 2008. Her research expertise is in Complex Networks, information technology. She has been active in the mathematical modeling for diffusion processes. She has published many papers in these fields.

M. Salehi is an Assistant professor of Computer Engineering in Faculty of New Sciences and Technologies, University of Tehran, Iran. He received his B.S. degree in Computer Engineering from Iran University of Science and Technology (IUST), Tehran, Iran, in 2005 and M.S. degree in Computer Engineering, Amirkabir University, Tehran, Iran, in 2008 and PHD in Computer Engineering, Sharif University of Technology, Tehran, Iran. His research expertise is in mathematical modeling of engineering problems that have arisen in the fields of networked systems, information technology, Crowd Sourcing and Collective Activities. He has been active in the General Mathematics and Statistics and complex networks. He has published about 65 research papers. He has set up the Computer and Information Networks (COIN) laboratory at University of Tehran.