

# Quantitative and Qualitative Analysis of Overlapping Community Detection Algorithms

Günce Keziban Orman

**Abstract**—Communities are described as functional modules in complex networks. Most of the existing algorithms are dedicated to find network partition as communities although in real-life, small groups can also represent functional modules. Recently, an overlapping community finder Ego Based Merged Overlapping Communities, a.k.a EMOC, is introduced as finding small groups. In this work, we concentrate on EMOC which seems to find highly overlapping communities at small size. In real-world social systems, these types of communities correspond to small social groups such as family members or coworkers sharing some members. We evaluate an analytical and empirical comparative analysis of EMOC with foremost algorithms in the literature. First, we perform a quantitative analysis by using well-known evaluation metric; NMI. Second, differently from ordinary performance evaluation methods, we compare the topological properties of the algorithms' estimated community structures in order to evaluate their community identification quality. Performance evaluation experiments have been done on LFR artificial network benchmark. Furthermore, we consult three real-world networks for representing the use cases of EMOC. The results show that EMOC successfully identifies small and highly overlapping communities. Its quantitative performance results are supported by qualitative results. Although some algorithms might seem highly performing according to quantitative analysis, they might estimate communities with non realistic topological properties.

**Index Terms**—Overlapping Community, Performance Evaluation, Community Detection

## I. INTRODUCTION

Complex networks, as graph modelling of complex systems, constitute a powerful mathematical representation for studying the dynamics of interacting objects. One of the top issues in this domain is finding communities in a complex network [1], [2]. Communities are the network substructures such that the nodes in the same community should be more similar to each other than the ones of different communities [1], [3]. As the definition analogy, communities can be seen as the clusters of complex networks. They correspond to functionally related objects in real-world systems. Finding them help the researcher to discover the relation between different modules, inferring missing attribute values and predicting unobserved connections, etc.[4]. As a natural result, there are numerous applications such as well-known recommendation systems [5], viral marketing [6] or sentiment analysis [7].

As any clustering problem, community detection is also studied widely in different domains with different perspectives [1], [8], [9], [10], [11]. Because there is no labelling or

reference community structures, the definition of communities are described in a flexible manner. Rather than building a consensus for formal definition, the needs of applications are prioritized in the domain. Nevertheless, the most common approach is finding network segments, a.k.a partitioning, in which the communities do not share any node in common [11], [8]. However, in [12], the authors underlined that real-world social group segments have intersections. Thus, some types of communities have some nodes in common. Those types of communities are called overlapping communities. We encounter them especially in social networks in which discovering overlapping communities let us to reveal the dynamics of social interactions[13]. There is a high interest to overlapping community discovery as well [14], [15], [16], [17], [18], [19], [20], [21], [22].

A rich comparative analysis of the prominent overlapping algorithms are proposed in [18]. The interesting result of this comparison is that the higher the communities intersections, the worse the algorithms performance. The accuracy of the algorithms is affected by the intersect node number and the intersect community number. Some other limitations of existing solutions are first, most of them put every node into at least one community [17], [23], [24], [25] and second, the results communities are large sized [17], [25]. However, in real world networks, some nodes can become community-less whereas some communities can be small sized. Indeed, finding community-less nodes can be quite interesting because they might correspond to noise or outliers. A flexible algorithm, EMOC (ego based merged overlapping communities) is proposed in [26] for overcoming mentioned limitations of previous solutions. EMOC let the user regulate the size and the cohesiveness of the communities. First it finds all small sized closed groups around each ego node and second it merges similar node groups. It is not sensitive to the overlapping level and also extracts a specific type of outlier or noise. In [27], the authors studied the performance evaluation of EMOC by comparing it with the foremost overlapping community finders. This work is extending the previous analysis. It is dedicated to both explain analytically the algorithmic procedure and time complexity of EMOC and propose an exhaustive empirical analysis based on the experiments on artificial benchmark. Our main purpose is reveal quantitative and qualitative performance limits of EMOC in comparison with foremost overlapping community detectors. We consider OSLOM [15], GCE [14], MOSES [16], COPRA [17] and EGO-BASED [19] in this comparative study.

In the literature, there are many works dedicated for quantitative performance analysis [18], [28]. They in general propose a quantitative analysis on a benchmark including the comparison of the numerical results obtained by measuring the similarity between estimated and reference community structures via an accuracy evaluation metric. For instance, in

Manuscript received November 1, 2021; revised November 9, 2021. This work was supported by the Galatasaray University Research Fund (BAP) within the scope of project number fba-2021-1063, and titled "Niteliklendirilmiş çift yönlü ağlarda bağlantı tahmini ile öneri sistemleri geliştirilmesi".

G.K. Orman is an Assistant Professor of Computer Engineering Department, Galatasaray University, Istanbul, Turkey, (korman@gsu.edu.tr).

[18] an extensive comparative analysis of foremost methods, except the ego-centred network based ones, is performed. They used LFR artificial network generation model to create the benchmark [29] and modified NMI [30] to measure the performance. These types of comparisons neglect the quality of the estimated community structure. In [31], [32], the authors underline the importance of the qualitative analysis for disjoint community structure detection methods. They have shown that although an algorithm takes a high score by using a metric, e.g. NMI, it still can find unrealistic community structure. Thus, in this work, we measure not only the quantity but also the quality by examining some topological properties of the resulting community structures which are found by different algorithms.

Briefly, the two main contributions of this article is first analytically describing EMOC algorithm and second evaluating the performance of foremost overlapping community detection algorithms quantitatively and qualitatively to reveal their limits and advantages for different types of networks. During this comparison, we analyse in detail the properties of EMOC. In the following section, we explain some prominent works about overlapping community detection. In section III, we give the definitions and methods related to the details of EMOC. In section IV, we describe the LFR model, time performance of EMOC and accuracy results of all algorithms with their quantitative and qualitative comparisons respectively. Finally, in the last section, we give a brief conclusion and explain future aspects of this work.

## II. RELATED WORKS

In this section, we focus on some prominent works about overlapping community detection. We categorize existing algorithms according to their detection strategy into four classes. At first we explain them. At the end of this section, we describe a comparative study of the algorithms.

**Clique-based Methods:** One of the most popular overlapping methods is k-clique percolation [23]. This method first finds all the cliques of size k. It then places the adjacent cliques in the same community. It works well if the network contains dense substructures. However, searching the cliques not only makes the algorithm very slow but also results in a very limited number of communities. Another algorithm, GCE, based on cliques is proposed in [14]. GCE at first searches the maximum cliques and uses them as the seeds of communities. It then tries to maximize a local fitness function proposed by Lancichinetti et al. [30] by using greedy optimization. Because this method does not search all cliques and the expansion phase is performed by greedy, it works faster.

**Node Expansion Methods:** Lancichinetti et al. developed the method LFM similar to GCE [30]. But they utilized nodes instead than cliques as the seeds of communities. In LFM, they use the same fitness function as in GCE. This function reflects the proportion of total inner links to all links to the power of  $\alpha$ , a resolution parameter, related to a community. The performance of this algorithm depends directly on  $\alpha$ . Another method developed by Lancichinetti et al. is OSLOM [15]. OSLOM takes into account statistically significance of the communities for expansion. More clearly, it selects the most significant community w.r.t global null model among all possibilities. One can run OSLOM on the results of any other

algorithm to adjust the community structure. It not only finds overlapping structure but also hierarchical one. Moreover, we can detect the outliers or singleton communities with OSLOM.

**Information Propagation Methods:** COPRA, differently from previously described methods, detects the communities by using information propagation strategy [17]. Each node takes the most commonly appearing label among its neighbours synchronously at each expansion time interval. Algorithm continues until the labels do not change. COPRA uses the maximum community number to be found as an input parameter. This algorithm is faster than most other algorithms. However, COPRA does not produce robust results. More clearly, there can be a significant difference of the results after two different applications of COPRA to the same network. Another algorithm based on information propagation SLPA [33] has a strategy very similar to COPRA. Differently from COPRA, in SLPA, nodes do not choose the labels independently than previous expansion time interval. Every node has a memory to cache the label information of previous time intervals. The more frequent the labels in node memory, the more probable for the node to choose it. SLPA does not take any input parameters such as community number. Because of not forgetting the gained information of previous expansion states, SLPA results are more robust than COPRA.

**Other Methods:** The algorithm MOSES proposed in [16] uses links as seeds and expands them according to an objective function by using heuristic approximation. This objective function is based on the joint distribution over the set of communities, single connection probability of inner and outer of community. The authors claim that MOSES is successful at detecting highly overlapping structures. EGO-BASED which is based on merging friendship groups of ego-centred networks is proposed in [19]. This algorithm extracts at first ego-centred network of each node. It then detects the friendship groups in ego-centred networks as the connected components after removal of the ego node. Finally, it merges the friendship groups if intersecting node number of two groups is one less than the size of one of those groups. This algorithm uses first level neighbours of the ego nodes when creating ego-centred network. Thus, it is possible to ignore big friendship groups. Furthermore, when merging groups, the proposed criteria can be too restricted to find highly similar ones.

A descriptive state-of-art work about the most prominent algorithms, related benchmarks and quality measures are given in [18]. There is also an extensive comparative analysis of all methods we explained above, except the ego-centred network based one, regarding their performance on artificial networks generated with different (1) overlapping node number and (2) number of communities that overlapping nodes can belong. According to the results of modified NMI [30], the performance of all algorithms decreases by the increase of the value of those two parameters. In general, SLPA and GCE give the best results. However, the authors conclude that *the detection in networks with high overlapping density and high overlapping diversity still has space for improvements.*

## III. METHOD

In this section, we concentrate on analytical properties of EMOC. At first, we explain the preliminaries for further usage. Then we detail algorithmic properties and complexity of EMOC.

## A. Preliminaries

Given a plain network  $G = (V, E)$ ,  $V$  is a set of  $n$  nodes and  $E$  is a set of  $m$  links. Each node  $i \in V$  is also called an *ego*. For each ego  $i$ , its *ego-centred network at radius  $d_i$* ,  $G_i^{d_i} = (V_i^{d_i}, E_i^{d_i})$ , is a sub network of  $G$ .  $G_i^{d_i}$  is centred on ego  $i$  and is surrounded with  $d_i^{\text{th}}$  level neighbours of  $i$ .  $V_i^{d_i}$  is the set of ego  $i$  and its  $d_i^{\text{th}}$  level neighbours.  $E_i^{d_i}$  is the set of links between the nodes from  $V_i^{d_i}$ . Note that if  $d_i = 1$ , all the nodes in  $V_i^{d_i}$  are the direct neighbours of ego. If  $G$  is a connected network and  $d_i$  is equal to the diameter of  $G$ ,  $G_i^{d_i}$  is as same as  $G$ . Two *paths* between two nodes  $i$  and  $j$  are *node independent*, a.k.a *node disjoint*, if they do not have any common internal nodes except  $i$  and  $j$  [34]. It is said that  $i$  and  $j$  are *k-node-connected* if there is at least  $k \geq 0$  different node disjoint paths from  $i$  to  $j$  [35], [36]. Shortly, it is called *k-connected*.

## B. k-Connected Node Groups

Seeing real-world social phenomena from the perspective of each person, we might discover multiple cohesive groups around him. Some of those groups may represent his family members, his coworkers or his social friends. Those groups might have common members as well. For instance, a coworker might be a social friend as well. It is also possible that some connections are sole. It means, without creating a cohesive group, some people individually connects with some other people. We assume that the people in the same cohesive group have more chance to build new connections with each other than the people from different groups. In network modelling, these kinds of cohesive groups may correspond to the cliques or clique-like network objects. A clique is a small network pattern in which every node is connected with every other nodes. Actually, this definition is too restricted for representing cohesive groups around a person because it is not necessary that every pair of person in a group is connected with each other. Another way of representing those groups can be to regard simple connectivity. But this time the definition of group can be too relaxed for ensuring their cohesiveness. For instance, a large circular sub-network is also connected but the density of connection is too low to be named as cohesive. Between these two limits, e.g. clique versus simple connectivity, EMOC concentrates on the notion of *k-connection* of the nodes. It defines a cohesive group as a group of nodes in which each pairs of nodes are *k-connected*. For a given ego node  $i$ , it extracts each *ego-centred k-connected node groups*  $C_{ip} \subseteq V_i^{d_i}$  related to ego  $i$ . It is shortly said *k-connected group*. Note that if  $k$  is 1, which is its lowest value,  $\bigcup_p C_{ip} = V_i^{d_i}$ . However, if  $k$  is at its maximal value, several nodes may not belong to any group.

By definition of *k-connection*, an important property that let the EMOC finds communities arises. For *k-connected group* whose size is larger than one,  $\forall k > 1$ , the nodes stay

connected even if one node is removed from the group. This property is evident. According to the definition of *k-connection*, if a *k-connected group* has more than  $k$  nodes, it remains connected whenever fewer than  $k$  nodes are removed [36]. Thus, removing one node does not affect the connectivity of the rest of the groups when  $k > 1$ . On the basis of this property, each *k-connected group* remains connected even if one removes the ego node. If there are several different *k-connected groups* that are not interconnected via a node other than ego, removing the ego will separate them. EMOC uses this strategy for finding knit groups around egos. It proposes a simple procedure which is given in algorithm 1. This procedure considers ego-centred network of each ego. For an ego node, at first it eliminates the nodes which are not *k-connected* with the ego in studied ego-centred network. It then finds the node groups which stay connected even in case of absence of ego. Those types of groups correspond to the connected components of ego-centred network after removal of ego.

---

**Algorithm 1** Finding Ego-centred *k*-Connected Groups
 

---

**Require:**  $G_i^{d_i}$ ,  $k$ ,  $i$

**Ensure:**  $C_i = C_{i1}, \dots, C_{ip}$

```

1: for  $j \in V_i^{d_i}$  do
2:   if  $\text{nodeDisjointPathNumber}(i, j) < k$  then
3:      $G_i^{d_i} = \text{remove}(G_i^{d_i}, j)$ 
4:   end if
5: end for
6:  $G_i^{d_i} = \text{remove}(G_i^{d_i}, i)$ 
7:  $C_i = \text{extractConnectedComponents}(G_i^{d_i})$ 
8:  $C_i = \text{insertEgoToAll}(C_i, i)$ 
    
```

---

Let us try to explain the idea of finding ego-centred node groups for different  $k$  values with an example. We represent in figure 1-(A), ego-centred network of node  $i = 1$  (coloureds as black) at radius  $d_i = 3$ . Assume that we search 2-connected groups. After checking the number of node disjoint paths, nodes  $j = 2$  and  $j = 11$  will directly be eliminated. So, those nodes will not be placed in any group. Note that although node  $j = 2$  has a direct contact with ego, it will not appear in any ego-centred 2-connected group because of not connecting with any node else. We see in figure 1-(B-1), ego-centred network after elimination of not 2-connected nodes. As it can be seen, there are two different cyclic groups whose meeting point is ego. Removing ego results separation of those groups but each of them still stay connected inside. In figure 1-(C-1), we represent two 2-connected groups found after removing ego with different colours. Ego belongs to both of them. That is why; we paint it with two colours. In case of looking for 3-connected groups in the same ego-centred network, we obtain different results. After removing not 3-connected nodes with ego, nodes 5, 6, 8 are remaining (shown in figure 1-(B-2)). Removal of ego makes them stay connected so there is only one node group including all the nodes remaining (shown in 1-(C-2)).

The procedure which is given in algorithm 1 takes three input parameters; (1) minimum number of node disjoint paths ( $k$ ), (2) ego-centred network ( $G_i^{d_i}$ ) and (3) ego ( $i$ ) itself. It first eliminates the nodes whose node disjoint path numbers to the ego are less than  $k$  (lines 1, 2, 3, 4 and 5). Here, function

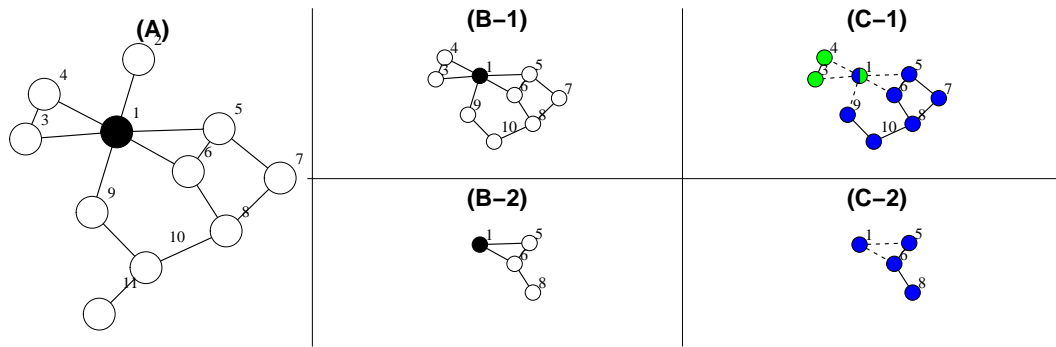


Figure 1. (A) Ego-centred network of Node 1 at radius 3. Black node is ego and white nodes are its neighbours of radius 3. (B-1) Same ego-centred network after removing nodes which are not 2-connected to ego. (B-2) Same ego-centred network after removing nodes which are not 3-connected to ego. (C-1) 2-connected groups in same ego-centred network. Each different colour shows belonging to a different node group. Ego belongs to all groups. That is why, it has two colours. (C-2) 3-connected group in same ego-centred network.

nodeDisjointPathNumber(...) in line 2 calculates the number of disjoint paths between ego  $i$  and scanned node  $j$ . EMOC uses push relabel max flow algorithm [37] for finding the number of different paths between two nodes. It is one of the fastest methods that is based on mathematical optimisation. The complexity of this step is  $O(|V_i^{d_i}|^3)$ . It is calculated for all nodes, so it makes  $O(|V_i^{d_i}|^4)$ . Functions `remove(...)` and `insertEgoToAll(...)`, given in lines 6 and 8 respectively, removes  $i$  from and inserts  $i$  to ego-centred network respectively. In case of using edge list data structure, remove operation demands one scan in the edge list of ego-centred network for updating link and node structure of network. So, it can be done in linear time  $O(|E_i^{d_i}|)$ . Insert operation works on the node sets of connected components. It does not need to scan the connected component node sets but it only visits each connected component node sets for adding ego. In total, this task demands a linear time operation in  $O(|C_i|)$ . Extracting connected components after removal of ego (given in line 7 as function `extractConnectedComponents(...)`) demands traversing studied network once. This task can be done with Breath-first Search approach. Running time of this step is  $O(|V_i^{d_i}| + |E_i^{d_i}|)$ . So overall running time of this algorithm is  $O(|V_i^{d_i}|^4 + |E_i^{d_i}| + |C_i| + |V_i^{d_i}| + |E_i^{d_i}|) \sim O(|V_i^{d_i}|^4)$ . As the longest process is to find disjoint path numbers between two nodes, the overall complexity of the algorithm is  $O(|V_i^{d_i}|^4)$ . **For extracting connected components  $O(V + E)$  (line6)**

### C. Ego Based Merged Overlapping Communities

Once all  $k$ -connected groups in the entire network have been found, EMOC merges the similar ones in order to eliminate the separation of same or similar groups. It considers Jaccard coefficient,  $sim(S_1, S_2) = |S_1 \cap S_2| / |S_1 \cup S_2|$ , for measuring the similarity of two groups  $S_1$  and  $S_2$  respectively. The groups whose Jaccard coefficient are above the given *threshold* are merged.

The pseudo code of merging process is shown in algorithm 2. Algorithm takes two input parameters; (1) the set  $S$ , whose members are the node groups found for entire network, and (2) the minimum ratio of the similarity (*threshold*). It outputs overlapping community structure as the updated set  $S$ . It firstly sorts  $S$  according to the size of node groups in increasing order (line 1). Computing size of each group can

---

#### Algorithm 2 Merging Node Groups

---

**Require:**  $S$ , *threshold*

**Ensure:**  $S$

```

1: while  $S_i \in S$  do
2:   while  $S_j \in S$  do
3:     if  $sim(S_i, S_j) \geq threshold$  then
4:       union( $S_i, S_j$ )
5:       update( $S$ )
6:     end if
7:   end while
8: end while
    
```

---

be done in constant time. We apply simple radix sort which requires  $O(|S|)$  where  $|S|$  is the total number of node groups. Then, The procedure checks, for each couple of node groups,  $S_i$  and  $S_j$ , if their similarity is greater than given *threshold* (line 3). Similarity can be computed in  $O(\min(|S_i|, |S_j|))$  with the usage of hash table for storing the elements of one of the node groups. In case of being sufficiently similar, these two groups are merged (line 3) and the groups  $S_i$  and  $S_j$  are updated. Merging phase can also be done in  $O(\min(|S_i|, |S_j|))$  with hash table. The algorithm continues until each couple of node groups is processed. The average size of a node group can be given  $n/|S|$ . So, overall time complexity of merging phase is  $O(2 \times (n/|S|) \times |S|^2) \sim O(n \times |S|)$ .

EMOC at first, extracts ego-centred network of each node  $i$  at radius  $d_i$ . This can be done in linear time by using edge list representation of the graph. It then finds  $k$ -connected groups related to each ego by using algorithm 1 and creates the global set  $S$  of node groups over entire network. Finally, it applies algorithm 2 and creates the overlapping community structure. Total time complexity of these two steps is  $O(|E| + |V_i^{d_i}|^4 + n \times |S|)$ . We assume that  $|V_i^{d_i}|^4 \gg n \times |S|$  and  $|V_i^{d_i}|^4 \gg |E|$ . It results  $O(|V_i^{d_i}|^4)$ . This complexity highly depends on the density of network and the chosen radius value  $d_i$  to create ego-centred networks. If network is sparse and  $d_i = 1$ , size of ego-centred network will be  $\log n$  [19]. So, time complexity will be  $O((\log n)^4)$ . In case of dense network, or high values of  $d_i$ , complexity can be  $O(n^4)$ . Indeed, considering each node as an ego gives the opportunity to make the computation in parallel programming. Thus, in practice, time complexity can be reduced effectively.

EMOC considers three parameters: (1)  $d_i$ : Radius of ego-centred network to adjust the size of the communities. As this parameter can be constant for network, it can also change from one node to another depending on the topology of the network, (2)  $k$ : Number of node disjoint paths, to regulate the cohesiveness of node groups, (3) *threshold*: Minimum ratio of node groups similarity to merge them. Adjusting values of these parameters requires topological analysis of the studied network. Nevertheless, in the most basic form, one can set constant  $d_i = 1$ ,  $k = 2$  and *threshold* = 0.8 for considering first-level ego-centred network, minimum cohesiveness inside the groups and high similarity of the groups. Note that decreasing the value of *threshold* may result high overlapping of the communities. Depending on the values of parameters, EMOC might result a specific type of outliers or noise nodes. Considering the network on figure 1-(A) as a whole network, setting  $d_i = 1$ ,  $k = 2$  and *threshold* > 0.5, EMOC results two communities, {1,3,4} and {1,5,6,7,8,9,10}. Two nodes, 2 and 11 remain. They stay community-less. Those nodes have single connection and are not in a cyclic group. Hence, they are not put into any community. Changing the values of the parameters, the topological types of those nodes change.

#### IV. PERFORMANCE EVALUATION

We conduct experiments on artificial networks to see the performance and the limits of overlapping community detection algorithms. Current section is dedicated to explain these experiments. We generate artificial networks with predefined overlapping community structure by using LFR model [29].

##### A. LFR Artificial Network Generator

LFR model allows generating random networks with power-law degree distribution and with predefined community structure having power-law community size distribution [29]. This is the most realistic model in the literature [28]. One can change the topology of generated networks with various parameters, e.g. number of nodes  $n$ , desired average  $\langle k \rangle$  and maximum degrees  $k_{max}$ , exponent  $\gamma$  for the degree distribution, exponent  $\beta$  for the community size distribution,  $c_{min}$  and  $c_{max}$  for minimum and maximum community sizes respectively and mixing coefficient  $\mu$  for the desired average proportion of outer community links. It provides a rich environment for evaluating the performance and the limits of community detection algorithms. It is previously used for this issue in many works [28], [18], [31]. The version of LFR we consult at this work generates plain networks with predefined overlapping community structure. In this version, we can also control total number of nodes  $O_n$  which belong to more than one community and maximum number of communities  $O_m$  that a node can belong in overlapping structure.

LFR determines at first the possible number of inner links and membership numbers of each node by using  $\mu$ ,  $O_m$ ,  $O_n$ ,  $\langle k \rangle$ ,  $k_{max}$  and  $\gamma$  and the possible sizes of communities with  $c_{min}$ ,  $c_{max}$  and  $\beta$ . Once the initial network is generated by using configuration model [38], the assignment of the nodes to the communities is done by a rewiring process managing generation of a bipartite network whose two sides are the nodes and the communities of main network. In this bipartite network, each node has as many links as its membership

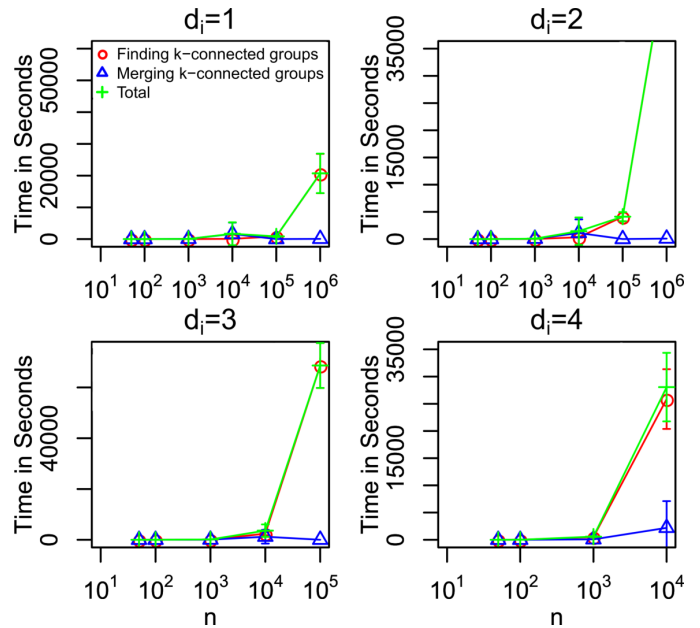


Figure 2. Execution time of EMOC on different network sizes for different  $d_i$  values. Top-left, top-right, bottom-left and bottom-right plots correspond to the performance for  $d_i = 1, 2, 3$  and  $4$  respectively. Given that the execution time span for larger networks are too much, they are not represented.

number and each community has as many links as its size. We evaluate the performance of EMOC by comparing it with 5 different overlapping algorithms: GCE[14], OSLOM[15], COPRA [17], MOSES[16] and EGO-BASED [19]. We implement EGO-BASED by considering its pseudo-code given in [19]. In this section, we first explain quantitative performance results of the algorithms in terms of their NMI scores. [30]. NMI is a well-known metric which is used before for performance analysis of community detection algorithms [28], [32], [30]. The version we use here is the modified one for overlapping community structures. Finally, we perform a qualitative analysis by explaining the changes of some topological properties of estimated community structures in comparison with the ones of reference structure.

##### B. Time Performance of EMOC

In section III-C, we underline that time performance of EMOC depends on the density of network or the value of parameter  $d_i$ . To see these effects, we generate LFR networks with increasing  $n$  values. The values of the generation parameters are as follow:  $n = \{5 \times 10^1, 10^2, 10^3, 10^4, 10^5, 10^6\}$ ,  $\langle k \rangle = 5$ ,  $k_{max} = 25$ ,  $\gamma = 3$ ,  $\beta = 2$ ,  $c_{min} = 5$ ,  $\mu = 0.1$ ,  $O_n = 50$  and  $O_m = 2$ . We set the values of  $c_{max}$  depending on the values of  $n$ . So, it takes 10, 25, 25, 250, 2500 and 25000 respectively. In figure 2, we see execution time of EMOC for the networks with increasing numbers of nodes in logarithmic plots. Each plot in the figure represents time performance for different  $d_i$ . Different colours of the curves symbolise the execution time of different steps or total time of EMOC. According to this, the larger the network, the higher the execution time for any  $d_i$  value. The execution time for the computation of merging process can be ignored when comparing it with the computation of  $k$ -connected groups.

Almost all execution time of EMOC is spent for finding  $k$ -connected groups for any  $n$  and  $d_i$ . For  $n > 10^5$ ,

Table I  
LFR NETWORK GENERATION PARAMETER VALUES

	Parameters	Values
1	$\mu$	{0.1, 0.3, 0.5}
2	$(c_{min}, c_{max})$	{(5, 25), (10, 50), (20, 100)}
3	$O_n$	{50, 100, 500}
4	$O_m$	[2, 10]

execution time increases suddenly too much when  $d_i = 1$  and 2. We observe a similar trend when  $d_i = 3$  and 4 for  $n > 10^4$  and  $n > 10^3$  respectively. Indeed, the analytic time complexity calculation is confirmed with experimental results. However, as we indicated before, EMOC is suitable for parallel programming because of handling each node separately when finding  $k$ -connected groups. This can be a solution for decreasing execution time.

### C. Quantitative Comparison

We evaluate the performance of six algorithms by using normalised mutual information, NMI. NMI is a well-known metric which is based on information theory. It is frequently used to assess the accuracy of estimated communities [28]. A modified version of NMI for overlapping communities is proposed by Lancichinetti et al. [30]. As the traditional one, modified NMI takes 0 if two compared overlapping structure is totally dissimilar and it takes 1 if they are exactly same. Previously, the modified version is used for comparing the accuracy of the foremost overlapping community detection algorithms in [18].

In our experiments, we mainly concentrate on the performance changes w.r.t overlapping level change of the network. That is why; we fix the values of some LFR parameters and modify only few of them which are related to our aim. The fixed parameter values are based on previous works [18], [28]. Finally, we generate networks with  $n = 1000$ ,  $\langle k \rangle = 10$ ,  $k_{max} = 50$ ,  $\gamma = 3$  and  $\beta = 2$ . The values of other parameters are given in table I. The increasing values of  $\mu$  in table I allow us decreasing the level of community separation. We assume that we generate networks with well, medium and few separated communities for  $\mu$  values given in table I respectively. The different values of the second parameter  $(c_{min}, c_{max})$ , minimum and maximum community size pair, in table I allow us generating small, medium and large communities respectively. We can adjust the overlapping level with  $O_n$  and  $O_m$ . We assume that we generate networks with small, medium and large numbers of overlapping nodes respectively for the values of  $O_n$  given in table I. The values of last parameter  $O_m$  allow us generating networks with increasing numbers, from 2 to 10, of overlapping communities that a node can belong. We mainly interpret the results of the algorithms performances according to their behaviour to the increase of  $O_m$ . We generate 10 networks for the combination of each parameter values given in table I. We run all 6 algorithms on each network and consider the average performance on 10 networks. Amongst these algorithms, we run COPRA 10 times for each network to ensure its consistency. We set three parameters of EMOC as  $d_i = 1$ ,  $k = 2$  and  $threshold = 0.8$ . The explication we give in this section is valid for all the results we have found although we only present some examples of NMI plots.

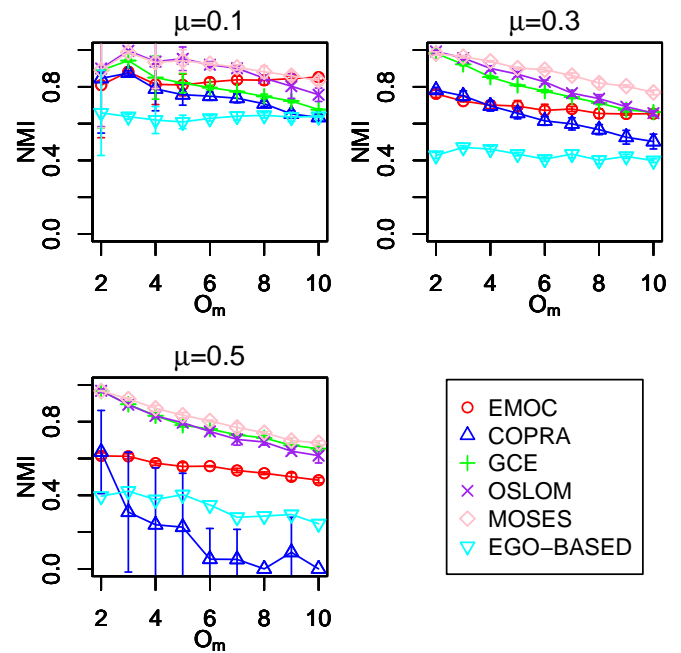


Figure 3. NMI Result comparison of six algorithms for  $(c_{min}, c_{max}) = (5, 25)$ ,  $O_n = 50$ . Top-left plot is for well-separated communities ( $\mu = 0.1$ ). Top-right plot is for medium-separated communities ( $\mu = 0.3$ ). Bottom-left plot is for few-separated communities ( $\mu = 0.5$ )

1) *Sensitivity to Community Cohesiveness*: In figure 3, we represent NMI results of the algorithms when we increase  $O_m$  for different  $\mu$ . For  $\mu = 0.1$ , all the algorithms take  $NMI > 0.8$  (figure 3 top-left). When  $O_m \leq 5$ , OSLOM and MOSES are the most performing ones. GCE, EMOC and COPRA follow them. EGO-BASED seems the least performing one. We observe a few linear decrease of the performance with the increase of  $O_m$ . The two ego based methods are less sensitive to this fact. Amongst them, EMOC seems more performing than EGO-BASED. Considering the increase of  $\mu$ , we notice that especially the performance of EMOC, EGO-BASED and COPRA decreases. However, EMOC and EGO-BASED still can keep their performance stable to the increase of  $O_m$ . COPRA does not exhibit a robust behaviour especially when  $\mu = 0.5$  (figure 3 bottom-left).

2) *Sensitivity to Community Size*: We show NMI results of all algorithms according to different community sizes on different plots in figure 4. Apparently, for all algorithms, the easiest case is the case that the network has small communities whose sizes change between 5 and 25 (figure 4 top-left). When community sizes are between 10 and 50, the algorithms performance is still good (figure 4 top-right). However, we observe a linear decrease for all algorithms except ego based ones with the increase of  $O_m$ . Two ego based methods keep a stable performance. Amongst them, EMOC results are as good as other algorithms while EGO-BASED is one step backward than them. Especially, when  $O_m > 8$ , EMOC and OSLOM perform the best. Here, we want to remind that running parameters of EMOC is set for finding small communities ( $d_i = 1$ ). So, it is not surprising that its performance for large communities is not as good as the ones for small communities.

3) *Sensitivity to Overlapping Density*: To see the behaviour of the algorithms for increasing overlapping den-

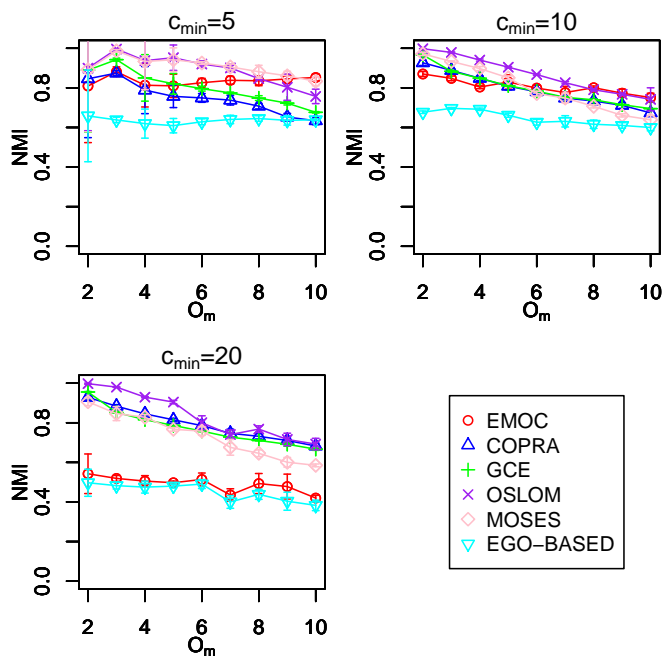


Figure 4. NMI Result comparison of six algorithms for  $\mu = 0.1$ ,  $O_n = 50$ . Top-left plot is for small communities ( $(c_{min}, c_{max}) = (5, 25)$ ). Top-right plot is for medium communities ( $(c_{min}, c_{max}) = (10, 50)$ ). Bottom-left plot is for large communities ( $(c_{min}, c_{max}) = (20, 100)$ )

sity, we experiment on increasing  $O_n$  values. Here, we also increase  $O_m$  values till 15 to realise the level of the stability of algorithm performances. In figure 5, we represent NMI results of the algorithms for networks generated with  $O_n = 50, 100$  and  $500$  in plots top-left, top-right and bottom-left respectively. Here, the easiest case for all algorithms is  $O_n = 50$ . When the numbers of overlapping nodes increase to 100, we observe a visible linear decrease on the performance of all algorithms except ego based ones. EMOC's performance is as similar as the case of  $O_n = 50$ . Amongst the expansion based methods, MOSES exhibit better performance than the others even for high  $O_m$  values. It is claimed that MOSES is successful for detecting highly overlapping structures [16]. As seen in top-right plot, EMOC performs as well as MOSES for  $O_m \geq 6$ . Its performance is higher than all algorithms when  $O_m > 9$ . For these networks, EMOC has a stable and robust performance even if we increase  $O_n$  and  $O_m$ . In case that the half of the nodes in the network overlaps (figure 5 bottom-left), the decrease of the performance of all algorithms with the increase of  $O_m$  becomes more visible. The performance of the algorithms OSLOM, GCE and MOSES decrease logarithmic. Amongst them, MOSES has a smoother decreasing trend. COPRA has a sudden decrease. Even for low  $O_m$  values, its performance is worse than the others. Two ego based methods result similar performance trends and NMI values.

By overall observation of the results for every parameter combination, we see that the performance of the algorithms GCE, OSLOM and MOSES are similar and good when communities do not overlap too much. However, their performances are affected by overlapping density and diversity. In contrast to this fact, two ego based methods seems more stable than the others for changes on overlapping level. Amongst them EMOC exhibits better results than EGO-

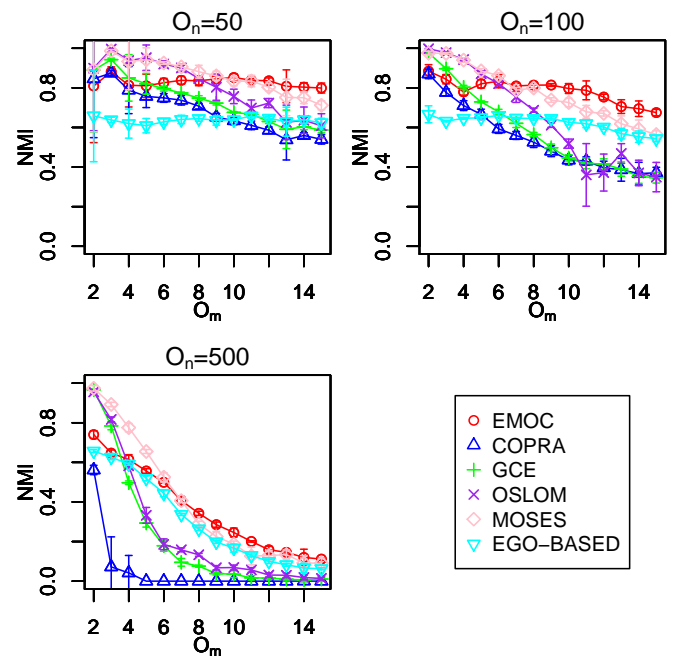


Figure 5. NMI Result comparison of six algorithms for  $\mu = 0.1$ ,  $(c_{min}, c_{max}) = (5, 25)$ . Top-left plot is when there is few number of nodes which belong to more than one community ( $O_n = 50$ ). Top-right plot is when there is medium number of nodes which belong to more than one community ( $O_n = 100$ ). Bottom-left plot is when there is large number of nodes which belong to more than one community ( $O_n = 500$ )

BASED in many cases. Especially, if the communities are well-separated and their sizes are small, EMOC results are as good as the most performing algorithms. Nevertheless, a performance decrease is observed for large or non-cohesive communities. In the next section, complementary qualitative analysis is explained.

#### D. Qualitative Comparison

To decide the quality of the estimated community structure of the algorithms, we analyse their topological properties and compare them with the reference structure. In [31], the authors focus on the distribution of community-level topological properties (e.g. hub dominance, scaled link density, community size). Here, we work on overlapping community detectors. Our aim is revealing their behaviour to the changing overlapping state of the networks. Thus, we mainly concentrate on the topological properties reflecting their overlapping status. We study six properties: (1) number of communities, (2) median of community size, (3) community size distribution, (4) maximum number of communities that a node overlaps, (5) number of total overlapping nodes and (6) number of community-less nodes. Amongst those properties, (4) and (5) are in fact determined by the generator parameters  $O_m$  and  $O_n$  of LFR respectively. We use those properties for assigning the compatibility of estimated structures with the reference one. Moreover, we also want to reveal how the algorithms differentiate amongst each other in terms of finding overlapping nodes. The properties (1), (2) and (3) that we consult give us an idea about if an estimated community structure looks like the reference one in the perspective of their sizes. By its construction, community size distribution of reference structure follows power-law distribution. That is

the reason why we are oriented at median rather than mean. In the following part of this section, the readers find graphical representation and the interpretation of those topological properties for the algorithms and LFR reference structure. The different coloured lines at each plot represent the value of the mentioned topological properties trends for increasing  $O_m$  values. Because for  $d_i = 3$  and 4 execution time span of larger networks than  $10^5$  and  $10^4$  respectively is too much, they are not represented.

In figure 6, number of communities found by each algorithm and generated by LFR is shown when increasing the values of  $O_m$ . Each plot represents the results for different  $c_{min}$  and  $c_{max}$  (see table I). Accordingly, we notice that the number of communities in reference structure increases from 50 to 100 with the increase of  $O_m$  when community size is small ( $c_{min} = 5$ ). Except EGO-BASED, all estimated structures have a similar numbers of communities to the reference one. EGO-BASED seems to find much more communities than reference and the rest of the algorithms. We observe a similar trend when community size is medium ( $c_{min} = 10$ ). However, when community size is large ( $c_{min} = 20$ ), EMOC behaves similarly with EGO-BASED. It finds much more communities than reference. Because those two algorithms have a similar strategy of detection and they work with small node groups at radius 1 around each ego node, it is not surprise that they find small but more communities than the reference. One of the assumptions of EMOC is that it finds small communities which might correspond to the small friendship groups or family members in real-world social networks.

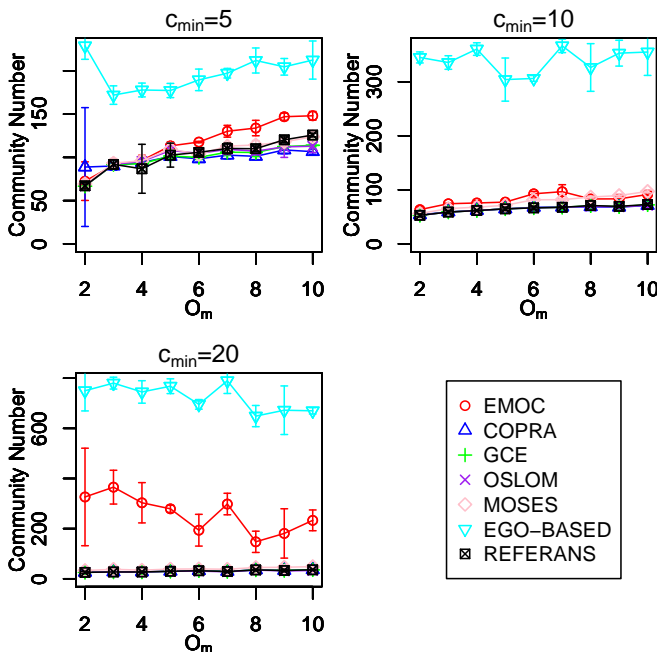


Figure 6. Number of estimated communities found by each algorithm and LFR generated reference structure for  $\mu = 0.1$ ,  $O_n = 50$ . Top-left plot is for small communities ( $(c_{min}, c_{max}) = (5, 25)$ ). Top-right plot is for medium communities ( $(c_{min}, c_{max}) = (10, 50)$ ). Bottom-left plot is for large communities ( $(c_{min}, c_{max}) = (20, 100)$ )

Regarding median of estimated and reference community sizes (see figure 7), we notice that two ego-centring methods find communities whose median ranges between 10 and 20

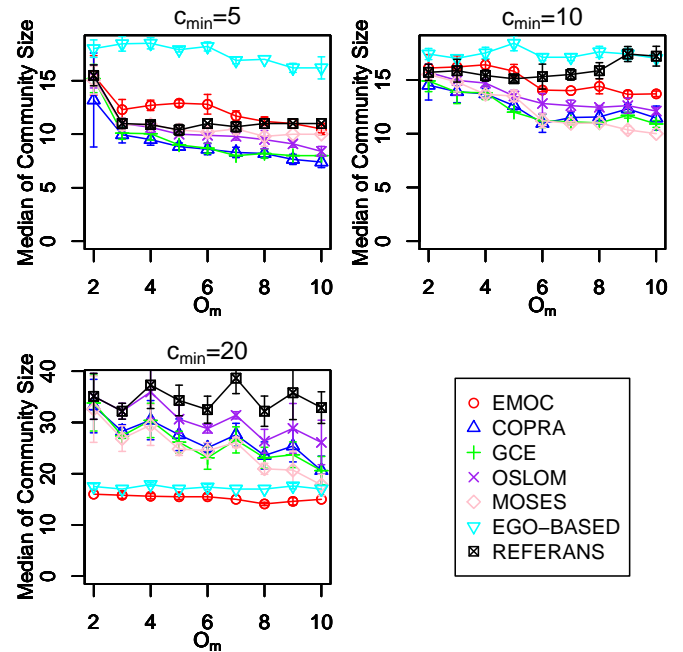


Figure 7. Median of the size of estimated communities found by each algorithm and LFR generated reference structure for  $\mu = 0.1$ ,  $O_n = 50$ . Top-left plot is for small communities ( $(c_{min}, c_{max}) = (5, 25)$ ). Top-right plot is for medium communities ( $(c_{min}, c_{max}) = (10, 50)$ ). Bottom-left plot is for large communities ( $(c_{min}, c_{max}) = (20, 100)$ )

independently from LFR generation parameter values. For other 4 algorithms' results and also for the reference structure, median values depend on LFR generation parameter values. In general, we observe that the higher the community sizes, the less similar the estimated communities to the reference one for all algorithms. Amongst them, EMOC has three parameters that might help the user to adjust the result community sizes. In these experiments, we execute EMOC with its default parameter values. However, increasing the radius ( $d_i$ ) or decreasing the similarity *threshold* of merging node groups may result larger communities. In order to have a deeper knowledge about the sizes of estimated communities, we examine the size of all generated communities. In figure 8, we represent probability density for different  $\{c_{min}, c_{max}\}$  values both for estimated structures and the reference one. Note that LFR uses power-law distribution for community size. Nevertheless it respects all its parameter values when generating networks. As a result, final community size distribution has still heavy tail (as it is in power-law distribution) but it looks more like Poisson distribution (see black line plot in figure 8). We see that EMOC, EGO-BASED, GCE and MOSES community sizes range between 5 and 30 independently than  $\{c_{min}, c_{max}\}$  LFR generation parameters. Thus, even if the reference communities are large, these algorithms find smaller ones. We notice that the most similar community sizes to the reference is the one of OSLOM.

In figure 9, we show maximum number of communities that a node overlaps when  $O_m$  increases for different  $O_n$ . Here, we want to notice that  $O_m$  is the control parameter when we look for estimated  $O_m$  for different algorithms. Thus, it is a natural trend that reference structure has a linear increase with the increase of  $O_m$ . For any algorithm,



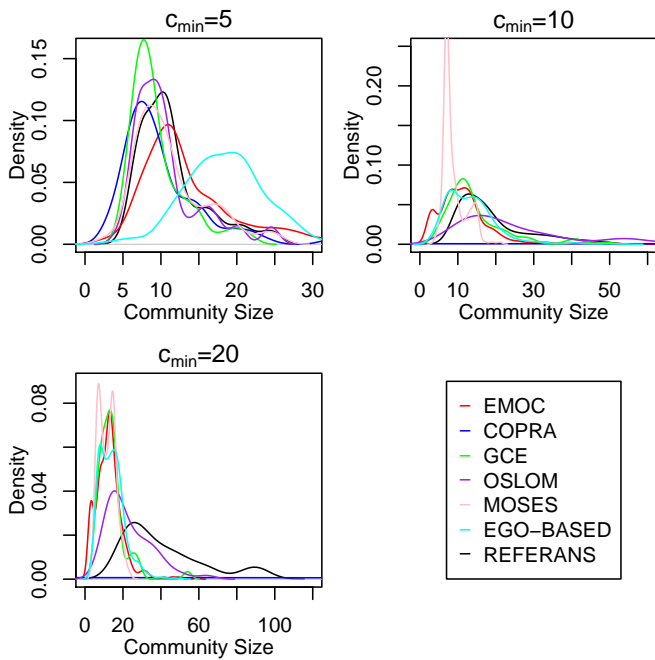


Figure 8. Density of community sizes found by each algorithm and LFR generated reference structure for  $\mu = 0.1$ ,  $O_n = 50$ ,  $O_m = 6$ . Top-left plot is for small communities ( $(c_{min}, c_{max}) = (5, 25)$ ). Top-right plot is for medium communities ( $(c_{min}, c_{max}) = (10, 50)$ ). Bottom-left plot is for large communities ( $(c_{min}, c_{max}) = (20, 100)$ )

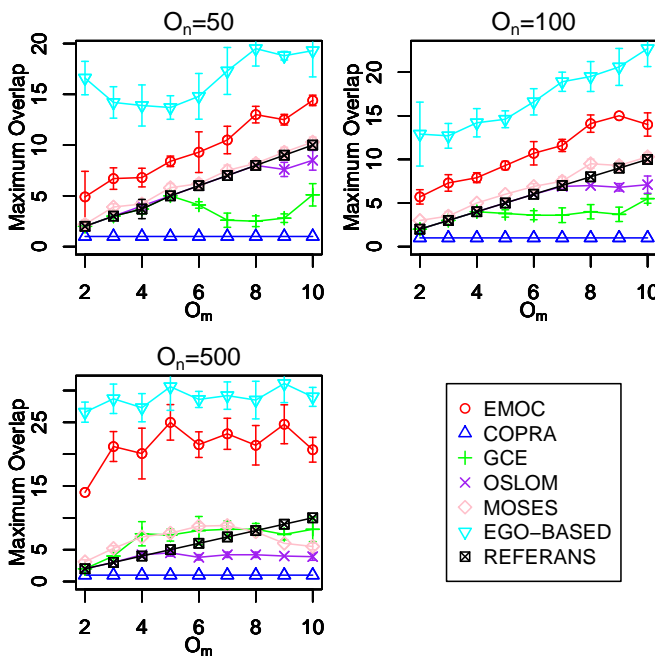


Figure 9. Maximum number of overlapping communities found by each algorithm and LFR generated reference for  $\mu = 0.1$ ,  $(c_{min}, c_{max}) = (5, 25)$ . Top-left plot is when there is few number of nodes which belong to more than one community ( $O_n = 50$ ). Top-right plot is when there is medium number of nodes which belong to more than one community ( $O_n = 100$ ). Bottom-left plot is when there is large number of nodes which belong to more than one community ( $O_n = 500$ )

we also expect such a linear increase. As it can be seen in figure 9, COPRA does not find any overlapping community for any case. It seems MOSES and OSLOM find community structure with the most similar maximum overlapping community to the reference one for any case. However, we also

notice that when half of the nodes overlap ( $O_n = 500$ ), they differentiate from the reference. Especially, OSLOM seems not to find as many overlapping communities as the reference one for this case. About GCE, we notice that it cannot find as many overlapping communities as the reference one when  $O_m > 5$  for  $O_n = \{50, 100\}$ . Two ego-centring methods find much more overlapping communities than the reference one for any case. Amongst them, we observe a more linear trend for EMOC. EMOC is designed for finding small and highly overlapping communities. These graphical results confirm its compatibility for this aim.

In figure 10, we see estimated  $O_n$  values by the algorithms. COPRA does not find any overlapping nodes for any  $O_n$  control parameter values. It seems all the algorithms result different  $O_n$  from the reference one when half of the nodes overlap. For other two cases ( $O_n = \{50, 100\}$ ), OSLOM finds as much overlapping nodes as the reference one. MOSES finds little more while GCE finds little less. Two ego-centring methods result much more overlapping nodes than the reference one. Amongst them EGO-BASED puts all nodes into more than one community for any case while EMOC results %30 of overlapping nodes in the network.

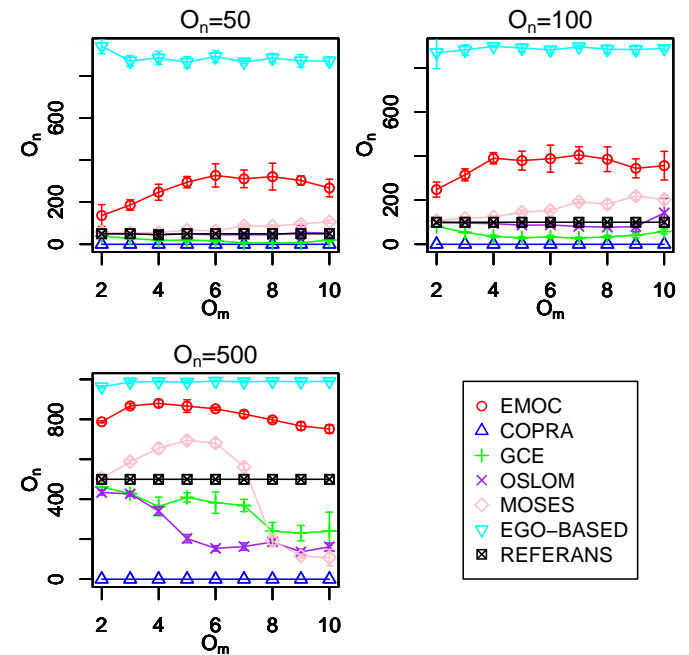


Figure 10. Number of overlapping nodes found by each algorithm and LFR generated reference for  $\mu = 0.1$ ,  $(c_{min}, c_{max}) = (5, 25)$ . Top-left plot is when there is few number of nodes which belong to more than one community ( $O_n = 50$ ). Top-right plot is when there is medium number of nodes which belong to more than one community ( $O_n = 100$ ). Bottom-left plot is when there is large number of nodes which belong to more than one community ( $O_n = 500$ )

We also examine the number of community-less nodes found by each algorithm. In the reference community structure, LFR puts all nodes into at least one community. Thus, we expect that the algorithms should not result any community-less nodes. You can find some examples of the found community-less node numbers for different  $O_n$  and  $O_m$  values in figure 11. Although we present here only some examples of the figures, the results are valid for all cases. Accordingly, we notice that except GCE and MOSES, all the algorithms put all the nodes into at least one community.

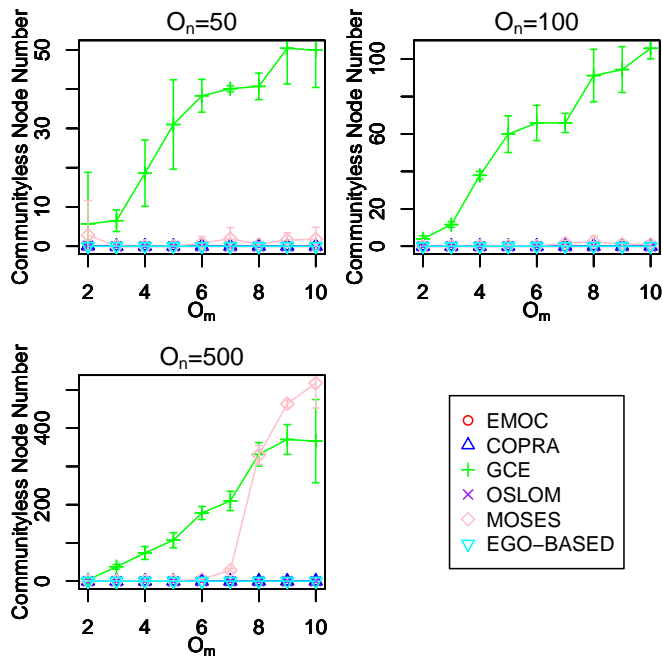


Figure 11. Number of overlapping nodes found by each algorithm and LFR generated reference for  $\mu = 0.1$ ,  $(c_{min}, c_{max}) = (5, 25)$ . Top-left plot is when there is few number of nodes which belong to more than one community ( $O_n = 50$ ). Top-right plot is when there is medium number of nodes which belong to more than one community ( $O_n = 100$ ). Bottom-left plot is when there is large number of nodes which belong to more than one community ( $O_n = 500$ )

However those two algorithms output some community-less nodes. Especially, GCE finds many community-less nodes. Their numbers are affected both by  $O_m$  and  $O_n$ . For MOSES, it seems when there is too much overlapping node with many overlaps between communities ( $O_m > 6$  and  $O_n = 500$ ), it cannot place many nodes into the communities. We want to remind here that those two algorithms' NMI score are too high to be compared with some other algorithms. However, here we discover that they might result many community-less nodes. Hence, the reliability of quantitative comparison of the algorithms might be open to criticism. One single measure is not efficient to explain the performance of the algorithms but it should be supported by qualitative analysis which enlightens the properties of the algorithms. We also want to underline that algorithms OSLOM and EMOC might find outliers or noise nodes (see Section III for EMOC and [30] for OSLOM). If it exists, they result community-less nodes. However those community-less nodes correspond to a specific type of outliers. Because LFR does not generate networks with those types of nodes, those two algorithms do not find community-less nodes.

Interpreting quantitative and qualitative results together, we summarise the behaviour of the algorithms for different kinds of networks. Here, we explain shortly our results for each algorithm. EMOC is convenient for finding small and medium sized communities. It can find highly overlapping structures. Because it is ego-based, it can be used for finding different communities that a node can belong. When the communities are not well-separated, EMOC may not find appropriate communities. It can detect specific types of outliers. COPRA cannot find overlapping structure. It is not robust. There is a risk that COPRA puts all nodes into one

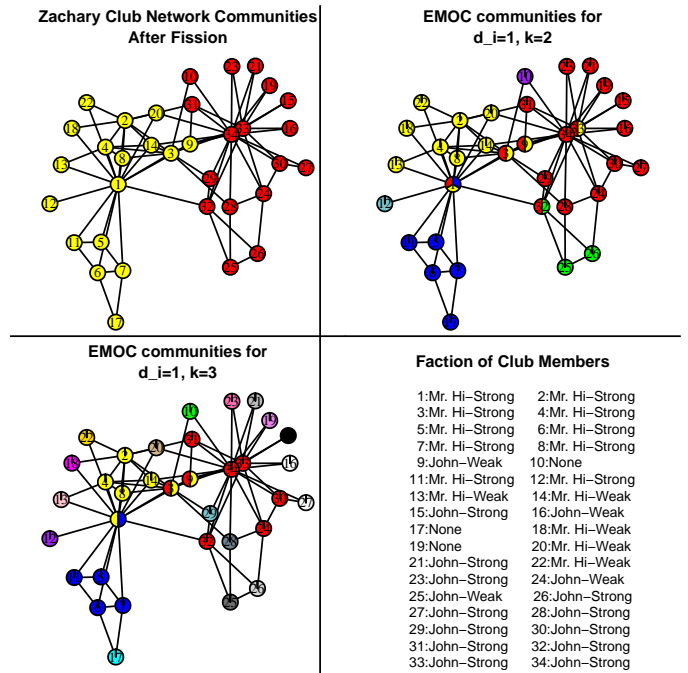


Figure 12. Zachary Karate Club Network. The different colours of nodes represent belonging to different communities. Nodes with multiple colours belong to multiple communities. Left and right plots represent ground-truth and EMOC communities for  $d_i = 1$ ,  $k = 2$  and  $threshold = 0.8$  respectively.

community. However, it is a fast algorithm. Hence, it can be combined with another algorithm to refine the result community structure. GCE is successful to find small and medium sized communities. It cannot catch highly overlapping parts. GCE is not affected by community separation level too much. Thus, it is convenient to find communities even if they are not visible at first sight. However, this algorithm does not put many nodes into any community. Thus, its result structure should be misleading. OSLOM, like GCE, is not affected by community separation level too much and it cannot catch highly overlapping parts. Differently from other algorithms, it is successful to find small, medium and also big sized communities. OSLOM can result outliers or noise nodes. Likewise, MOSES is not affected by community separation level too much and it cannot catch highly overlapping parts. It is good for detecting small and medium sized communities. As GCE, many nodes might remain community-less. So, its result structure might be misleading. EGO-BASED finds both small and medium sized communities. It is affected by community separation level too much. Thus, it might not be good for detecting non-visible communities. Differently from all the mentioned algorithms, it assigns more than one community for every node. Hence, it detects over overlaps which might be misleading for the user.

E. Results on Real-World Networks

We apply EMOC on two real-world networks. The first one is well-known Zachary karate club network [39]. This network is created by observing Zachary club members for 2 years. It shows the relations of 34 club members. Club members split into two groups because of political conflict between karate trainer John (node #34) and club president Mr. Hi (node #1). There are two natural communities whose leaders are those two members. These communities are

represented by different colours in top left plot of figure 12. The faction of each club member is also declared as strong or weak connection with one of the leaders or none. We observed the effect of EMOC parameter values on Zachary karate club network.

We find 4 communities (shown in top right plot of figure 12) when the parameters are at their default values. Natural communities of Zachary network seem to be split by EMOC. Union of red and green communities of EMOC substantially correspond John's group. Likewise, yellow and blue communities of EMOC constitute Mr. Hi's group. There are 5 overlapping nodes (#1, 3, 9, 32, 33). Mr. Hi (node #1) belongs to three communities that 2 of them correspond to his group after split. Evaluating centrality scores, overlapping nodes have the highest betweenness centrality. Overlapping nodes are either lying in-between two groups or they are embedded in the core centre of the groups. The most interesting result is about node #9. Its betweenness score is not as high as others. This node was a weak supporter of John before the split but he joined Mr. Hi's group afterwards [39]. EMOC puts him into two communities that each of them correspond to the groups of different leaders in reality. Two nodes (#10 and 12) are not placed into any community. Regarding their topological position, node #12 is connected only with node #1 (Mr. Hi). Node #10 has only two connections. The faction of this node is marked as *none*. Node #10's two friends are Mr. Hi's strong supporter and John himself. His idea about leaders is neutral and his friends are homogeneous. As a result, although he is placed in one of the groups in reality, we cannot claim that he is embedded there.

When we increase the minimum number of disjoint paths from  $k = 2$  to  $k = 3$ , EMOC naturally finds more and smaller communities (see bottom-left image of Figure 12. Not only major communities are split but also many nodes becomes community-less. Among them, node #10,17 and 19 did not belong to any group after the split. EMOC could catch those nodes as community-less when we increase  $k$  value. Observing general topology of the nodes overall, only the tightly knit groups did not dissolve. Perhaps, the ideal EMOC parameter values depend on both network size and its density. In general, EMOC finds consistent communities with real groups in Zachary Karate network. Overlapping nodes have important topological situation. Community-less nodes can be either non-effective or easily affected by other people.

Second real-world network we deal with is Facebook Network [40]. This network is a combination of 10 ego-centred networks that each of them includes the social circles of ten different Facebook users. There are 4039 nodes corresponding 10 ego and their friends and 88234 links representing the friendship relation of them. This network is ego-centralised by its construction. We apply EMOC with same parameter values. We examine a possible relation between numbers of communities that a node belongs and topological properties. For this reason, we represent in figure 13 the scores of page rank, betweenness, closeness and degree centralities with the numbers of overlapping communities for each node. As it is case for Zachary karate network, in Facebook, the most overlapping nodes (#108 and 1685, points at the top-right corners of each plot) are the most central ones. Their page rank score is also high. Those nodes correspond to two egos having hub position in the network. Some nodes belonging

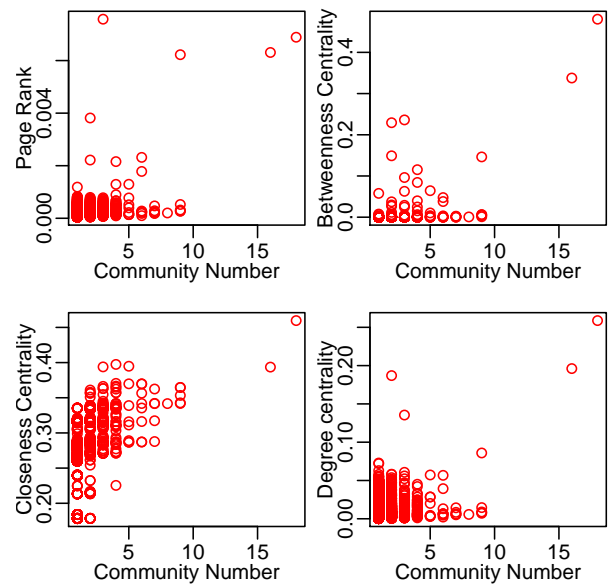


Figure 13. Relation between nodal topological measures and the numbers of communities that Facebook users belong. Each red circle in the plots corresponds to the scores of one node. Top-left, top-right, bottom-left, bottom-right plots show the relation between community numbers and page rank, betweenness, closeness and degree centralities respectively

to low numbers of community have high page rank scores. Looking at them in more detail, we notice that those nodes are not central themselves but they have direct connection with important hubs. As a result, their page rank score is high because of knowing important people.

Third real-world network is well-known dolphin social network [41]. There are 62 nodes with 159 links. According to [3], there are two communities in this network. Later on, with sub communities, it is stated as having four communities. Previous studies are dedicated to find network partition as dolphin communities. Looking at Figure 14, one can discover overlapping community results. OSLOM tends to find big communities with few numbers of overlapping nodes. COPRA and GCE find smaller communities. However, COPRA seems to find no overlapping nodes, while GCE finds them. MOSES and EMOC finds the smallest communities. Among all algorithms, only EMOC can discover the nodes belonging to more than two communities and identifies several different type outliers. For instance node #62 is the most isolated node and put into no community by EMOC (see top-left network in the figure). Node #58 is an important bridge but is not situated in any closed cyclic connection. That's why, it is in its own community. The examples are numerous. Briefly, considering three real-world network experiments, EMOC in general finds small communities. Some of them corresponds to highly interconnected node groups which do not have relation between each other except connecting to the same hub. In general the most overlapping nodes are those hubs. Other highly overlapping nodes are bridges connecting different communities. The community-less nodes are different types of outliers such as the people having unexpected behaviour or people who have no more connections than one in whole network.

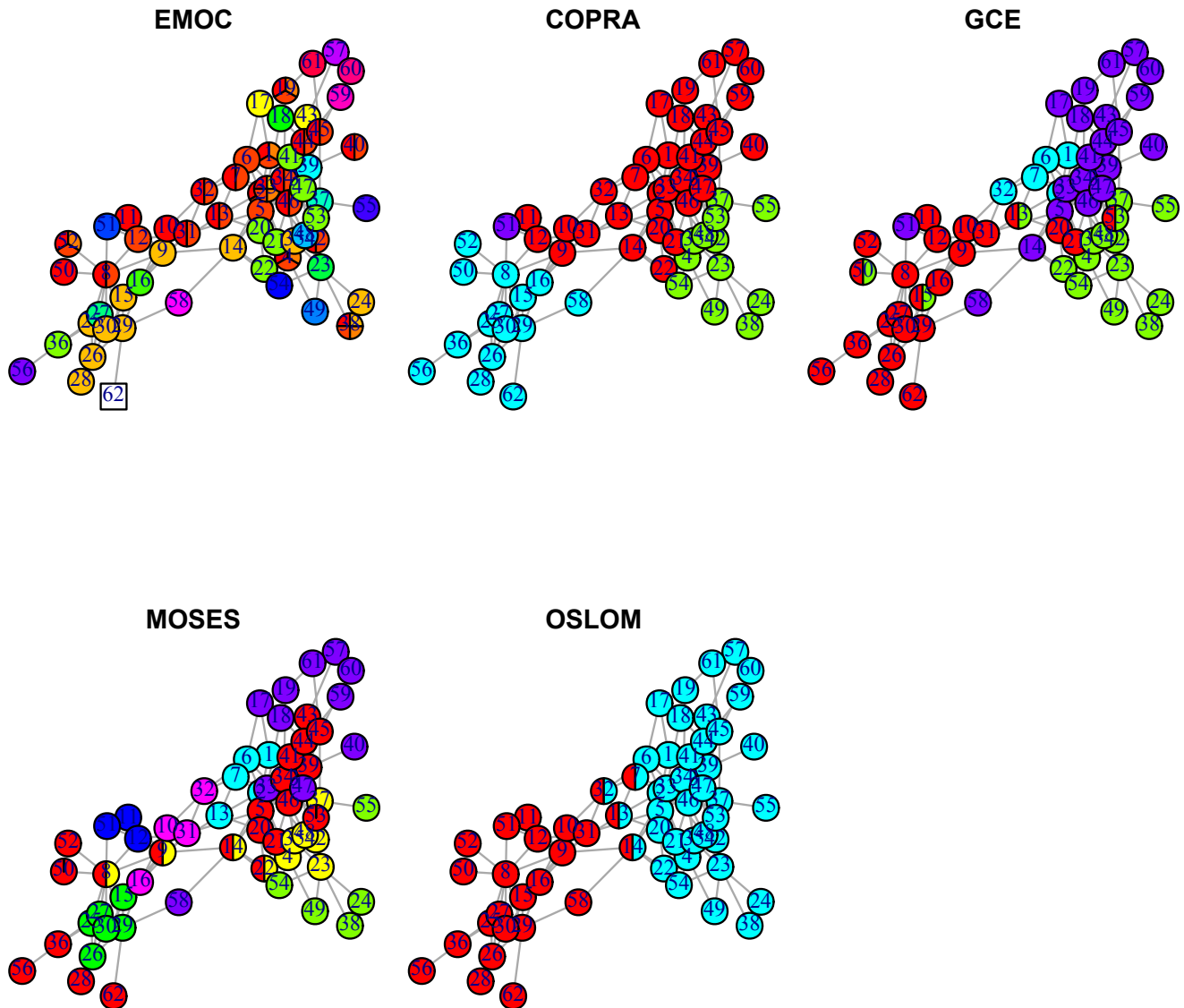


Figure 14. Overlapping community structures of Dolphin network found by different algorithms.

## V. CONCLUSION

In this work, we examine the algorithm EMOC analytically in detail and perform a qualitative and quantitative analysis to evaluate its overlapping community structure detection performance by comparing it with five foremost methods in the literature. EMOC is based on a definition of  $k$ -connected group around each ego. A  $k$ -connected group is a cohesive group in which every couple of nodes have at least  $k$  different node disjoint paths between them. As it handles each node separately, EMOC is convenient for parallel programming. To understand its detection performance, we consult to not only quantitative analysis, as it is the most common method for algorithms performance comparison, but also qualitative analysis which includes to reveal the topological quality of the result community

structure. Qualitative analysis is usually ignored in such comparative studies although it helps to better understand the real performance. In our experiments, we use an artificial network benchmark generated by LFR model. Comparing EMOC with other algorithms, it is highly performing to find small and well-separated communities with high overlapping nodes. In real world social systems, they might be family members or close friend groups. Indeed, the experiments on Zachary Karate Club and facebook networks are validated that EMOC finds small and highly overlapping communities. Its performance is not affected by changing amount of overlapping nodes or overlapping communities. In the literature, existing approaches are limited on these issues. Thus, EMOC can be a solution for finding the communities at any overlapping level.

On one hand different perspectives related to the development of EMOC can be listed as; first, examining the roles of EMOC's input parameters on the topology of detected communities, second, applying it on different types of real-world networks and interpreting the results and third, developing a strategy to automatically and dynamically determine EMOC parameter values according to the topological positions of the nodes. Moreover, it is also possible to modify EMOC for weighted and directed networks. Modifications for directed networks can easily be done by considering link directions at the ego-centred network creation and finding the numbers of node disjoint paths. For weighted networks, it is possible to put an additional criterion for node disjoint path search or to merging strategy. On the other hand, our results reveal important facts about algorithm comparison. We see that although some algorithms have high score on performance metrics, their estimated community structure might be misleading. Hence, we want to ameliorate algorithms performance evaluation in three ways. First, we want to use other metrics such as omega index and other topological properties related to overlapping community structure. Second, we want to analyse in detail the weak points of metrics and propose a more complete ones which overcome current issues. Third, we want to develop new topological measures for assigning the roles of nodes into the overlapping state.

## REFERENCES

- [1] S. Fortunato, "Community detection in graphs," *Phys. Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
- [2] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003.
- [3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [4] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *ICDM*, 2013, pp. 1151–1156.
- [5] J. Chen, Y. Tang, J. Li, C. Mao, and J. Xiao, "Community-based scholar recommendation modeling in academic social network sites," in *WISE*, 2014, vol. 8182, pp. 325–334.
- [6] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, May 2007.
- [7] P. Parau, A. Stef, C. Lemnaru, M. Dinsoreanu, and R. Potolea, "Using community detection for sentiment analysis," in *ICCP*, Sept 2013, pp. 51–54.
- [8] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *JSTAT Mech.*, p. P10008, 2008.
- [9] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *ISCI 2005*. Springer Berlin Heidelberg, 2005, vol. 3733, pp. 284–293.
- [10] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, 2004.
- [11] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *PNAS*, vol. 105, no. 4, p. 1118, 2008.
- [12] J. Yang and J. Leskovec, "Overlapping communities explain core-periphery organization of networks," *Proceedings of the IEEE*, vol. 102, no. 12, pp. 1892–1902, Dec 2014.
- [13] F. Reid, A. McDaid, and N. Hurley, "Partitioning breaks communities," in *ASONAM*, 2011, pp. 102–109.
- [14] C. Lee, F. Reid, A. McDaid, and H. Neil, "Detecting highly overlapping community structure by greedy clique expansion," in *SNA-KDD10*, 2010.
- [15] A. Lancichinetti, F. Radicchi, J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS ONE*, vol. 6, no. 4, p. e18961, 2011.
- [16] A. McDaid and N. Hurley, "Detecting highly overlapping communities with model-based overlapping seed expansion," in *ASONAM*, 2010, pp. 112–119.
- [17] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.
- [18] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 43:1–43:35, Aug. 2013.
- [19] B. Rees and K. Gallagher, "Overlapping community detection by collective friendship group inference," in *ASONAM*, 2010, pp. 375–379.
- [20] X. Chen and J. Li, "Community detection in complex networks using edge-deleting with restrictions," *Physica A: Statistical Mechanics and its Applications*, vol. 519, pp. 181 – 194, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378437118315358>
- [21] H. Jin, W. Yu, and S. Li, "Graph regularized nonnegative matrix tri-factorization for overlapping community detection," *Physica A: Statistical Mechanics and its Applications*, vol. 515, pp. 376 – 387, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378437118312251>
- [22] T. Chakraborty, S. Ghosh, and N. Park, "Ensemble-based overlapping community detection using disjoint community structures," *Knowledge-Based Systems*, vol. 163, pp. 241 – 251, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705118304258>
- [23] I. Derenyi, G. Palla, and T. Vicsek, "Clique percolation in random networks," *Physical Review Letters*, vol. 94, no. 16, 2005.
- [24] J. Xie, B. Szymanski, and X. Liu, "Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *ICDMW*, Dec 2011, pp. 344–349.
- [25] S. Gregory, "A fast algorithm to find overlapping communities in networks," in *ECML PKDD '08*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 408–423.
- [26] G. K. Orman, O. Karadeli, and E. Çalıřır, "Overlapping communities via k-connected ego centered groups," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ser. ASONAM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1598–1599. [Online]. Available: <https://doi.org/10.1145/2808797.2809351>
- [27] G. K. Orman, "Multiple communities of ego in social networks," in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2021*, London, U.K., 7-9 July, 2021, pp. 128–133.
- [28] G. Orman and V. Labatut, "A comparison of community detection algorithms on artificial networks," in *DS*, 2009, vol. 5808, pp. 242–256.
- [29] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E*, vol. 80, p. 016118, Jul 2009.
- [30] A. Lancichinetti, S. Fortunato, and J. Kert'esz, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.
- [31] G. K. Orman, V. Labatut, and H. Cherifi, "Comparative evaluation of community detection algorithms: a topological approach," *JSTAT Mech*, vol. 2012, no. 08, p. P08001, 2012.
- [32] G. Orman, V. Labatut, and H. Cherifi, "Qualitative comparison of community detection algorithms," in *DICTAP*, vol. 167, 2011, pp. 265–279.
- [33] J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," in *PAKDD*, 2012, pp. 25–36.
- [34] D. R. White and F. Harary, "The cohesiveness of blocks in social networks: Node connectivity and conditional density," *Sociological Methodology*, vol. 31, no. 1, pp. 305–359, 2001.
- [35] M. L. Balinski, "On the graph structure of convex polyhedra in  $n$ -space," *Pacific J. Math.*, vol. 11, no. 2, pp. 431–434, 1961. [Online]. Available: <https://projecteuclid.org:443/euclid.pjm/1103037323>
- [36] R. Diestel, *Graph Theory*, ser. Electronic library of mathematics. Springer, 2006. [Online]. Available: <https://books.google.com.tr/books?id=aR2TMYQr2CMC>
- [37] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.
- [38] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random Structures and Algorithms*, vol. 6, no. 2/3, pp. 161–179, 1995.
- [39] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.
- [40] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *NIPS*, 2012, pp. 548–556.
- [41] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. [Online]. Available: <http://networkrepository.com>