# Centroid-Based Multiple Local Community Detection

Boyu Li, Dany Kamuhanda, and Kun He, *Senior Member, IEEE*

***Abstract*— In recent years, the research of local community detection has attracted much attention. Most existing local community detection methods aim to find a single community of closely related nodes for a given query node, but in general, nodes are possible to belong to several communities, and detecting all the potential communities for a given query node is much more challenging. In this work, we propose a novel approach called the centroid-based multiple local community detection (C-MLC) to find all the communities for a query node. Differing from the existing local community detection methods that directly find a community from the query node, we assume that every community contains a "centroid" node, which locates in the core of the community and can be used to identify the community. Then, a query node corresponds to several centroid nodes if the query node belongs to multiple communities. The key ideas of C-MLC are that C-MLC automatically determines the number of communities containing the query node by finding the related centroid nodes and uses each query node together with the centroid node to uncover the corresponding community based on a set of high-quality seeds. Through extensive evaluations on real-world networks and synthetic networks, C-MLC outperforms the state-of-the-art methods significantly, demonstrating that finding the centroid nodes is a better approach to uncover the multiple local communities.**

***Index Terms*— Centroid node, clustering, multiple local community detection (MLC), network analysis, seed set expansion.**

## I. Introduction

MANY complex systems can be modeled as networks [1], [2] in which nodes and edges represent the entities and relations, respectively. A group of nodes is considered as a community if the nodes have denser connections or exhibit a higher probability of being connected internally than externally. The research of community detection greatly contributes to the network analysis that uncovers latent structures and has been widely applied in various fields successfully, such as in biological networks [3], social networks [4], recommender systems [5], topic detection [6], and collaboration networks [7]. Thus, in the past decades, community detection has become one of the most significant topics in the area of network analysis [8], [9], [10].

The traditional community detection techniques aim to find all the communities in the network, termed as global community detection [11], [12], [13], [14], [15]. Recently, in many scenarios, instead of searching all the communities in a large network, one just wants to quickly uncover the communities for a small set of query nodes, termed as local community detection. The research of local community detection can be divided into two main categories, namely, single local community detection [16], [17], [18] and multiple local community detection (MLC) [19], [20], [21]. Single local community detection aims to find only one community containing the given set of query nodes, whereas multiple local community detection aims to find all the related communities for the set of query nodes. The set of query nodes can be just one single node, which is more challenging and studied by researchers in most cases.

Generally, nodes may belong to multiple communities in the network, and detecting all the communities for the query node is more important as it could uncover more potential information in the network. On the other hand, comparing to single local community detection, multiple local community detection is more difficult because the number of communities to be found is not given and it is hard to separately uncover the related communities. For example, in Fig. 1, given the red query node 38, there are two different communities containing 38 as shown in Fig. 1(a) and (b), respectively, where the green nodes are the members of the corresponding community. If a method for multiple local community detection determines and finds only a single community for the query node, the detection result is probably a mixer of the two ground-truth communities, and the potential information of the query node is uncovered insufficiently.

Addressing the above issues, M-local spectral (M-LOSP) [19] removes the query node from its ego network to obtain some independent connected components and then uses the local spectral (LOSP) algorithm to detect a community for each connected component. Multicom [20] is performed under the condition that the number of communities must be given. MLC [22] and sparseness-based MLC (SMLC) [23] calculate the number of communities and cluster the nodes using nonnegative matrix factorization (NMF). Despite effectiveness, these methods use global topological information or rely too much on the sampling strategy to count the number of communities and generate communities.

Recent studies [24], [25], [26] demonstrate that the core nodes of community maintain stability and are more distinct than other members of community. For instance, Xiang et al. [27] proposed an effective method that can detect multiple pairs of core–periphery structure and community structure in the networks by finding the core nodes. Therefore,

Manuscript received 12 July 2022; revised 28 September 2022; accepted 29 November 2022. Date of publication 14 December 2022; date of current version 31 January 2024. This work was supported by the National Natural Science Foundation of China under Grant 61772219. *(Boyu Li and Dany Kamuhanda contributed equally to this work.) (Corresponding author: Kun He.)*

Boyu Li and Kun He are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: afterslby@hust.edu.cn; brooklet60@hust.edu.cn).

Dany Kamuhanda is with the Department of Computer Science, University of Rwanda, Kigali 4285, Rwanda (e-mail: d.kamuhanda@ur.ac.rw).

Digital Object Identifier 10.1109/TCSS.2022.3226178

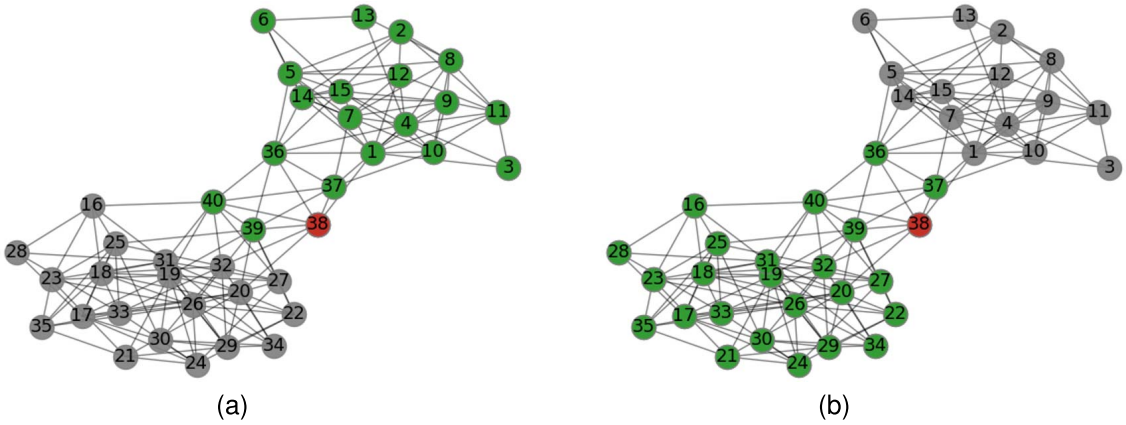


Fig. 1. Query node 38 is contained in two communities $C_1$ and $C_2$ which are represented as green nodes. Multiple local community detection aims to find the two communities based on the topological structure around the query node. (a) Community $C_1$. (b) Community $C_2$.

to effectively determine the number of communities, a feasible approach is to separately find the core nodes of the communities related to the query node. Then, we not only can automatically determine the number of potential communities according to the number of core nodes being found but also improve the detection result by uncovering the communities based on the core nodes. For instance, if we aim to uncover the two communities in Fig. 1 for the query node, finding some core nodes and picking a set of new seeds based on the core nodes for each community will be very helpful.

To this end, we propose a novel approach called centroid-based multiple local community detection (C-MLC) to detect multiple local communities for a query node. Inspired by the concept of "centroid" in the classic clustering methods, we assume that every community contains a centroid node which can be considered as a core node and identify the community uniquely. Besides, a query node could relate to a number of centroid nodes if the query node belongs to multiple communities. To find the core nodes related to the query node, C-MLC explores a path from each neighbor of the query node. The paths can always reach a set of core nodes of a related community, which is ensured by applying an appropriate localized metric of node similarity (NS). In this way, C-MLC automatically determines the number of potential communities containing the query node according to the centroid nodes being found based on local topological information. More centroid nodes are found if the query node belongs to more communities, and each node pair of the query node and a centroid node are regarded to be in a potential community. Then, for each centroid node, C-MLC combines the nodes along the explored paths as a set of initial seeds and applies a local community detection method to generate a community. As a result, C-MLC precisely uncovers all the communities based on the sets of high-quality seeds.

The main contributions of this work are summarized as follows.

1) We propose a new approach called C-MLC for multiple local community detection that can automatically determine the number of potential communities related to the query node.
2) We present a novel metric to measure the similarity between nodes based on localized topological information that can efficiently find the core nodes of communities related to the query node.
3) We generate a set of high-quality seeds based on the centroid node and explored paths to separately detect each community for the query node.
4) We conduct extensive experiments illustrating that C-MLC outperforms other competitive baselines on both the real-world networks and synthetic networks.

## II. Related Work

### A. Single Local Community Detection

The random-walk based models are the most popular approaches to uncover the local community, where personalized pagerank (PPR) [28], heat kernel (HK) [29], and LOSP [19] are three main techniques. These methods perform short random walks to diffuse the probability from the seeds, sort the nodes according to the probability vector in the decreasing order, and sweep the probability vector to find a set of nodes with the local minimum conductance. In this article, we apply PPR and HK to generate the communities expanded from seeds, and the detailed processes of these methods are introduced in Section III-D.

Moreover, some approaches [30], [31], [32], [33] are proposed that gradually expand the community starting from the query node based on the local modularity. An external node is added into the community if the node increases the local modularity to the most extent. The expansion process terminates when the local modularity could not increase any more. Yin et al. [34] proposed an algorithm called motif-based approximate PPR (MAPPR) that uncovers a local community with minimum motif conductance based on the PPR method. The MAPPR constructs a weighted graph according to the motif, uses the PPR approach to compute the probability vector, and sweeps the probability vector to find a set of nodes with minimal motif conductance. Zhang et al. [35] presented a semi-supervised approach called seed expansion with generative adversarial learning (SEAL) based on the generative adversarial network (GAN) [36], where the generator tries to uncover a community containing

the seed, and the discriminator uses the graph neural network (GNN) [37] to determine whether a community is real or produced by the generator. At the equilibrium between the generator and the discriminator, the generator produces a high-quality community according to the characteristics of the ground-truth communities.

In addition, in recent years, there are increasing studies focusing on finding the core nodes of community for improving the quality of community detection. Moradi et al. [38] quantify the importance of each node based on the total local influence. Jiang et al. [39] calculate the similarity scores between nodes according to some similarity metrics and pick the core node based on a graph coloring approach. A common weakness of these methods is that picking core nodes must relies on global topological information resulting in heavy computation in large networks. Furthermore, there are some approaches Ding et al. [17] and Bian et al. [40] proposed for single local community detection based on localized topological information. However, these methods are difficult to extend for the case of multiple local community detection. Compared with the existing methods, C-MLC uses only localized topological information to efficiently find the core nodes of different communities related to the query node. Therefore, C-MLC can be applied in more complex environments and efficiently uncovers more potential information.

### *B. Multiple Local Community Detection*

The random-walk based models can be flexibly extended to uncover multiple local communities. He et al. [19] proposed a method called M-LOSP that extends their single local community detection method of LOSP. Specifically, M-LOSP first removes the query node from its ego network and obtains some independent connected components. Then, for each connected component, M-LOSP adds the query node into the connected component and applies LOSP to find a community. Hollocou et al. [20] proposed Multicom to uncover multiple local communities given a set of query nodes. Multicom iteratively uses a scoring function (e.g., PPR, HK, or LEMON [41]) to detect a community for each seed. Then, density-based spatial clustering of applications with noise (DBSCAN) [42] is applied on the embeddings generated based on the scoring function, and the node with the highest degree is selected as a new seed in each cluster. The process finishes until the new generated clusters are not much different from the last generated communities.

NMF [43], [44], which is one of the most popular methods for global community detection, can also be extended for multiple local community detection. Kamuhanda and He [22] proposed an approach called MLC to address multiple local community detection. MLC samples a relevant subgraph from the original large graph and estimates the number of communities by decomposing the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ of the subgraph to obtain two matrices $\mathbf{W} \in \mathbb{R}^{n\times k}$ and $\mathbf{H} \in \mathbb{R}^{k\times n}$, where $\mathbf{A} \approx \mathbf{WH}$. The iteration stops when the normalized $\mathbf{H}$ (in which the values of each column sum up to 1) contains a row without any centroid node. To detect the communities, MLC applies a threshold on the community membership vectors in $\mathbf{H}$, which corresponds to the estimated number of communities, to assign nodes to communities. However, the efficiency of MLC heavily depends on the quality of the sampling strategy on large networks due to expensive computation of NMF decomposition.

Ni et al. [21] proposed a method based on local modularity with three main steps to find the local overlapping communities for a query node. First, a set of nodes are picked w.r.t. the query node by two heuristics (i.e., $C_0N$ and neighK) as a candidate set. Then, the algorithm selects the representative nodes from the candidate set as the seeds according to nearest node with greater centrality (NGC) and fuzzy relation [45]. Finally, the algorithm applies the methods of DMF_F [46] and M [31] to uncover the communities.

Although the existing multiple local community detection methods can find and regard the core nodes of communities as the initial seeds, the detection results may not be the expected communities. For example, given a core node in the overlapping region of two communities and the query node only belongs to one community, the detection result must involve many members of the other community if the expansion starts from only the core node. Our approach can alleviate this issue significantly because C-MLC not only finds the core nodes of related local communities by exploring paths but also separately uses the nodes along the paths as the initial seeds to uncover the corresponding communities. As a result, C-MLC can accurately uncover the target community even when the core node is in the overlapping region of some communities, which happens frequently in real-world social networks.

## III. Preliminaries

### *A. Problem Statement*

Given a network modeled as an undirected and unweighted graph $G = (V, E)$, where $V$ and $E$ denote the set of nodes and edges, respectively. $n = |V|$ is the number of nodes, and $m = |E|$ is the number of edges in the graph. Let $\mathbf{A} \in [0, 1]^{n\times n}$ be the associated adjacency matrix, and $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ be the set of ground-truth communities each containing the query node $s \in V$. The goal is to detect a set of communities $\mathcal{C}' = \{C'_1, C'_2, \ldots, C'_{K'}\}$ each containing the query node $s$, such that $K' \approx K$, and for any community $C_i \in \mathcal{C}$, there exists a detected community $C'_j \in \mathcal{C}'$ satisfying $C'_j \approx C_i$.

### *B. Evaluation Metrics*

In this article, we use the Jaccard $F_\sigma$-score [47] to measure the quality of detected communities for a query node, such that

$$F_\sigma = \left(1 + \sigma^2\right) * \frac{\text{precision} * \text{recall}}{\left(\sigma^2 * \text{precision}\right) + \text{recall}} \tag{1}$$

where $\sigma$ is a hyperparameter to control the balance of recall and precision, and we set $\sigma$ at 1 to compute the Jaccard $F_1$-score. The *precision* and *recall* are given as follows:

$$\text{recall}(C) = \max_j \frac{|C \cap C'_j|}{|C \cup C'_j|}, \quad C'_j \in \mathcal{C}' \tag{2}$$

and

$$\text{precision}(C') = \max_i \frac{|C_i \cap C'|}{|C_i \cup C'|}, \quad C_i \in \mathcal{C}. \tag{3}$$

### C. Node Similarity

The metric of NS plays an important role in C-MLC because the metric will lead the paths explored from the neighbors of query node to the cores of communities. The core of community has two basic properties, i.e., the nodes share many common neighbors and the links are dense. Therefore, we calculate the similarity for a pair of nodes using two metrics to satisfy the properties, respectively.

The first metric is the Jaccard index (JA) [48] defined as the ratio of common neighbors in the complete set of neighbors for the two nodes, such that

$$\mathrm{JA}(u,v)=\frac{|\Gamma(u,v)|}{|\Gamma(u)\cup\Gamma(v)|}$$

where $\Gamma(u)$ denotes the neighbors of $u$, and $\Gamma(u,v)$ represents the common neighbors of $u$ and $v$. JA ensures that for the last added node along the path, the expanded node always has the biggest ratio of common neighbors.

The second metric is the node influence area density (NI) [17] defined as the number of links that actually exists in $\Gamma(u,v)$ divided by the number of all possible links in $\Gamma(u,v)$, such that

$$\mathrm{NI}(u,v)=\frac{\left|\{(u',v')\in E|u',v'\in\Gamma(u,v)\}\right|}{\binom{|\Gamma(u,v)|}{2}}.$$

NI ensures that the path reaches the region of dense connections as far as possible.

Overall, we define the metric of NS as follows:

$$\mathrm{NS}(u,v)=\beta\cdot\ \mathrm{JA}(u,v)+(1-\beta)\cdot\mathrm{NI}(u,v) \tag{4}$$

where $\beta$ is a hyperparameter to control the balance of JA and NI. In the experiments, we set $\beta=0.5$, indicating that JA and NI are equally important. Note that both JA and NI can be calculated based on localized topological information. Consequently, (4) can be efficiently calculated during the process of exploring paths.

### D. Graph Diffusion Methods

Graph diffusion is one of the most popular approaches for local community detection [28], [29]. For a seed $s$, an initial one-hot vector $p^{(0)}\in\mathbb{R}^n$ consisting of 1 for the seed and 0 for the other nodes indicates the distribution of a random walker at the initial stage. Then, the random walker spreads the probability starting from the neighbors of $s$ over the graph. At the first iteration, the neighbors of $s$ have $1/d(s)$ probability of being visited, where $d(s)$ is the degree of node $s$. For the entire network, the diffusion of probabilities can be summarized into a transition matrix $\mathbf{N}_{\mathrm{rw}}$, where $\mathbf{N}_{\mathrm{rw}}=\mathbf{D}^{-1}\mathbf{A}$, and $\mathbf{D}$ is the diagonal degree matrix. Thus, the probability of reaching each node can be iteratively approximated by

$$p^{(l)}=\mathbf{N}_{\mathrm{rw}}^{\mathrm{T}}p^{(l-1)}.$$

PPR [28] diffuses the probabilities with $\alpha$ probability of following adjacent edges and $1-\alpha$ probability to restart, such that

$$p^{(l)}=\alpha\cdot\mathbf{N}_{rw}^{\mathrm{T}}p^{(l-1)}+(1-\alpha)\cdot p^{(0)} \tag{5}$$

where $\alpha\in(0,1]$.

**Algorithm 1** Framework of C-MLC

**Input:** Graph $G$, query node $s$
**Output:** The detected local communities $\mathcal{C}'$
1: $all\_paths=explore\_paths(G,s)$
2: $cen\_paths=choose\_centroids(G,all\_paths)$
3: $cen\_seeds=combine\_paths(cen\_paths)$
4: $\mathcal{C}'=\emptyset$
5: **for** $cen\in cen\_seeds$ **do**
6: $\quad C'=detect\_community(G,cen\_seeds[cen],s,cen)$
7: $\quad$ add $C'$ to $\mathcal{C}'$
8: **end for**
9: **return** $\mathcal{C}'$

HK [29] is a function of temperature $t$ and initial heat distribution $h^{(0)}$, such that $h=\mathbf{H}^{(t)}h^{(0)}$, where $\mathbf{H}^{(t)}$ is the heat operator given as $\mathbf{H}^{(t)}=e^{-t\mathbf{N}_{rw}^{\mathrm{T}}}$. The exponent of any matrix $\mathbf{A}$ is denoted as $e^{\mathbf{A}}=\sum_{k=0}^{\infty}(\mathbf{A}^k/k!)$. Then, the HK diffusion becomes

$$h=e^{-t}\left(\sum_{k=0}^{\infty}\frac{t^k}{k!}\left(\mathbf{N}_{rw}^{\mathrm{T}}\right)^k\right)h^{(0)}. \tag{6}$$

After obtaining the probability vector returned by PPR or HK, a sweep operation is performed on the probability vector to generate a community. Specifically, we denote the set of nodes $V_q=\{v_1,v_2,\ldots,v_n\}$ according to the probability-per-degree in descending order. Let $\mathbf{z}_i=\{v_1,\ldots,v_i\}(1\le i\le n)$ be an ordered set of nodes in $V_q$. The local community is generated by picking the set of nodes with the minimum conductance from the sets $\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_n\}$.

## IV. Methodology

In this section, we present the C-MLC algorithm that contains two main stages, as shown in Algorithm 1. In the first stage, C-MLC explores the paths from the neighbors of query node to find the centroid nodes of different potential communities (lines 1 and 2). In the second stage, for each centroid node, C-MLC uncovers a community based on a set of high-quality seeds (lines 3–8). We elaborate each stage as follows.

### A. Finding Centroid Nodes

The process of finding centroid nodes includes two main steps, namely, exploring paths and choosing centroid nodes. The goal of exploring paths is to expand the paths from the query node to the cores of different potential communities, and the detailed procedures are presented in Algorithm 2. For each neighbor node of the query node, C-MLC first initializes a path as a list including the query node $s$ and the neighbor node $u$, and then iteratively expands the path by adding the most similar node sim_node of the last added node until sim_node already exists in the path (lines 5–14). The most similar node sim_node is picked by function choose_sim_node($G$, last_node) as implemented in Algorithm 3. Specifically, C-MLC calculates the similarities between last_node and its neighbor nodes according to (4) and

**Algorithm 2** *Explore_Paths*$(G, s)$

**Input:** Graph $G$, query node $s$
**Output:** The explored paths *all_paths*

```
1: all_paths = ∅
2: for u ∈ Γ(s) do
3:   exploring_path = [s, u]
4:   last_node = u
5:   while True do
6:     sim_node = choose_sim_node(G, last_node)
7:     if sim_node ∉ exploring_path then
8:       add sim_node to exploring_path
9:       last_node = sim_node
10:    else
11:      add exploring_path to all_paths
12:      break
13:    end if
14:  end while
15: end for
16: return all_paths
```

**Algorithm 3** *Choose_Sim_Node*$(G, Last\_Node)$

**Input:** Graph $G$, last added node *last_node*
**Output:** The most similar node *sim_node*

```
1: max_score = −1
2: sim_node = null
3: for u ∈ Γ(last_node) do
4:   score = sim(G, u, last_node) /*calculate by (4)*/
5:   if score > max_score then
6:     max_score = score
7:     sim_node = u
8:   end if
9: end for
10: return sim_node
```

chooses the most similar node as sim_node. Note that to avoid repeated calculations of NS, we can store the similarity scores of node pairs in global variable during the whole exploration process.

In the second step, C-MLC separately chooses a centroid node from each explored path. Algorithm 4 shows the detailed process. For each explored path, C-MLC compares the degrees of the last two nodes along the path and chooses the node with higher degree as the centroid node (lines 3–6). The node with higher degree is more likely to be the core node as illustrated in [49] and [50]. If the degrees of the last two nodes are equal, C-MLC chooses the node with the bigger ID as the centroid node to avoid that a community is identified by different centroid nodes (line 8). For example, as shown in Fig. 2(a), for the query node 38, C-MLC explores two paths starting from its neighbors 1 and 36, respectively, and finds a common centroid node 12 for the two paths.[1] Moreover, as shown in Fig. 2(b), C-MLC explores another two paths starting from the neighbors 20 and 39, respectively, to find the same centroid node 33.

[1]For simplicity, we only show two explored paths for each centroid node.

**Algorithm 4** *Choose_Centroids*$(G, All\_Paths)$

**Input:** Graph $G$, the explored paths *all_paths*
**Output:** The centroid nodes and paths *cens_paths*

```
1: cens_paths = ∅
2: for path ∈ all_paths do
3:   if degree(G, path[−2]) > degree(G, path[−1]) then
4:     centroid_node = path[−2]
5:   else if degree(G, path[−2]) < degree(G, path[−1])
     then
6:     centroid_node = path[−1]
7:   else
8:     centroid_node = max(path[−2], path[−1])
9:   end if
10:  add path to cens_paths[centroid_node]
11: end for
12: return cens_paths
```

Note that in large networks, some nodes may have too many neighbors that need a big amount of calculation. To reduce such calculation, instead of exploring paths from all the neighbors of the query node, a sample of $\omega$ neighbors are randomly selected to speed-up the algorithm.[2] We have tested for various sample sizes, and $\omega = 10$ yields good results. This is because for real-world networks, almost every node belongs to less than five communities. Therefore, most of the explored paths lead to the same centroid node. Similarly, we sample $\gamma$ neighbors for last_node in Algorithm 3 to expand the most similar node. Based on the experiments, we set $\gamma$ to 100 that yields excellent results.

### B. Uncovering Local Communities

After exploring the paths and selecting the centroid nodes, C-MLC starts to classify the paths according to the centroid nodes as shown in Algorithm 5. Specifically, for each centroid node cen in cens_paths, C-MLC stores the nodes along all the paths in cens_paths[cen] to cens_seeds[cen] as a set of high-quality seeds. For instance, in Fig. 2(a), C-MLC combines all the nodes {38, 1, 12, 36, 14, 15} along the two paths as the set of seeds to uncover the community identified by centroid node 12. Similarly, in Fig. 2(b), C-MLC generates {38, 39, 19, 18, 33, 20, 26} as another set of seeds to expand the second community identified by centroid node 33.

Then, C-MLC generates a community for each centroid node as shown in Algorithm 6. To make full use of the centroid node and the query node, C-MLC allocates more probabilities to the centroid node and the query node in the initial vector. Thus, the centroid node and the query node play crucial roles in the process of uncovering the community. For example, in Fig. 2, assigning higher probabilities to centroid nodes 12 and 33 contributes to distinguishing different potential communities, and the generated communities are more prone to cover the corresponding members. In this work, we assign the probabilities to the centroid node, query node,

[2]Note that this strategy is an optional choice depending on the scale of network and the available computational resources.

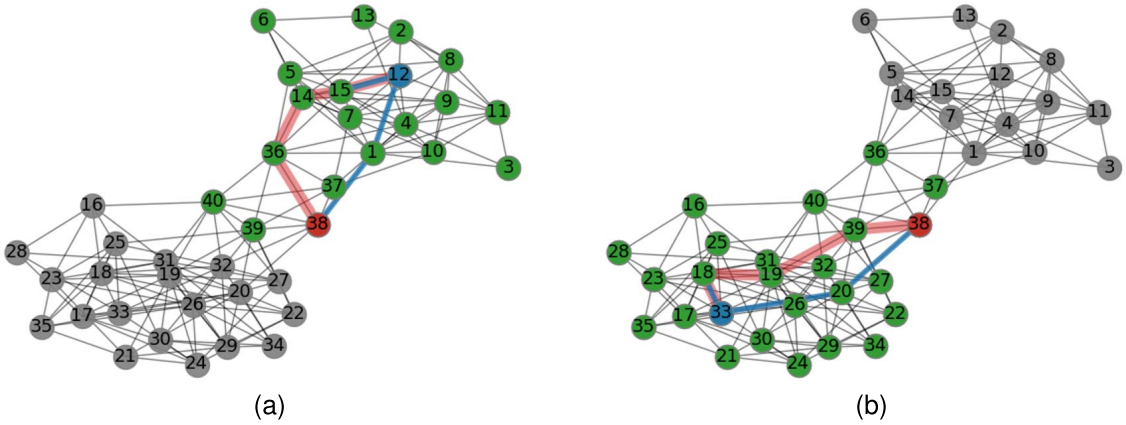


Fig. 2. C-MLC finds two centroid nodes 12 and 33 by exploring paths starting from the neighbors of the query node 38. (a) Community $C_1$. (b) Community $C_2$.

**Algorithm 5** *Combine_Paths*(*Cen_Paths*)

**Input:** The centroid nodes and paths *cens_paths*
**Output:** The centroid nodes and seeds *cens_seeds*

```
cens_seeds = ∅
for cen ∈ cens_paths do
  for path ∈ cens_paths[cen] do
    cens_seeds[cen] = cens_seeds[cen] ∪ cens_paths[cen][path]
  end for
end for
return cens_seeds
```

**Algorithm 6** *Detect_Community*($G$, *Cen_Seeds*[*Cen*], $s$, *Cen*)

**Input:** Graph $G$, seeds *cens_seeds*[*cen*], query node $s$, centroid node *cen*
**Output:** Generated community $C'$

```
initialize a probability vector p^(0) based on cens_seeds[cen], s and cen
apply PPR or HK on p^(0) to obtain a probability vector p
perform a sweep operation on p to obtain the community C' with the minimum conductance
return C'
```

and every other node as $(0.5/|\text{seeds}|) + 0.2$, $(0.5/|\text{seeds}|) + 0.3$, and $(0.5/|\text{seeds}|)$, respectively, where $|\text{seeds}|$ denotes the size of seeds. After initializing the probability vector, we apply a graph diffusion method, i.e., PPR [28] or HK [29], to uncover a community based on the probability vector.

### C. Complexity Analysis

The complexity of C-MLC mainly depends on the number of edges in the graph and the number of neighbors of the query node. In Section IV-A, we introduce a method of sampling neighbors that limits the complexity to $O(\omega)$. The complexity of (4) depends on the number of neighbors of the input nodes $u$ and $v$. Similarly, the worst case would be $O(\gamma + m)$ when either $u$ or $v$ has a large number of neighbors, where $m$ is the number of edges in the graph, and $\gamma$ is the number of sampled neighbors. Moreover, if the optional sampling strategies are not used, the complexity becomes $O(n + m)$, where $n$ is the number of nodes, and $m$ the number of edges in the graph.

For the stage of uncovering communities, the complexity depends on the algorithm applied for local community detection. As approximate PageRank is used by default, the complexity is $O((\log n/\epsilon\alpha))$ [28], where $n$ is the number of nodes in the graph, and parameters $\epsilon$ and $\alpha$ are used to control the walk length.

## V. Experiments

We conduct extensive experiments on the real-world networks and synthetic networks to measure the effectiveness of C-MLC. Let C-MLC$_P$ and C-MLC$_H$ represent that C-MLC grows the seeds into a community using PPR and HK, respectively. We fix $\alpha = 0.99$ and $\epsilon = 0.001$ for PPR, and $\epsilon = 0.01$ and $t = 80$ for HK. All the experiments are performed on a processor: i5 at 3.3 GHZ, RAM: 16 GB, and a 64-bit Windows operating system.

### A. Baselines

In this article, we use four state-of-the-art methods, namely, M-LOSP, Multicom, MLC, and SMLC, to compare with C-MLC. All the baselines perform with their default parameters, and their introductions are summarized as follows.

1) *M-LOSP [19]:* M-LOSP removes the query node from its ego network and obtains some independent connected components. Then, for each connected component, M-LOSP adds the query node into the connected component and applies the LOSP [19] method, which is a variant of the spectral method, to find a community.
2) *Multicom [20]:* Multicom iteratively uses a scoring function to embed the network and applies DBSCAN [42] on the embeddings to cluster the network. Then, for each cluster, Multicom obtains a new seed by picking the node with the highest degree and generates a community for the new seed. The process finishes until the new generated communities are not much different from the last generated communities.

TABLE I
PARAMETERS OF THE LFR NETWORKS

| Parameter | Description |
|---|---|
| $n \in \{5,000; 10,000\}$ | Number of nodes |
| $\mu \in \{0.1, 0.3\}$ | Mixing parameter |
| $\bar{d} = 10$ | Average degree |
| $d_{max} = 50$ | Maximum degree |
| $C_{min} = 20$ | Minimize size of community |
| $C_{max} = 100$ | Maximize size of community |
| $\tau_1 = 2$ | Node degree distribution exponent |
| $\tau_2 = 1$ | Community size distribution exponent |
| $om \in \{2, 3, 4, 5\}$ | Overlapping membership |
| $on \in \{100, 200\}$ | Number of overlapping nodes |

3) *MLC [22]:* MLC uses the breadth-first search (BFS) method to sample a subgraph from the original network and estimates the number of potential communities by iteratively decomposing the adjacency matrix **A** into two matrices **W** and **H**, where $\mathbf{A} \approx \mathbf{WH}$. The iteration stops when the normalized **H** contains a row without any centroid node. Finally, MLC applies a threshold on the community membership vectors in **H** to assign nodes to communities.
4) *SMLC [23]:* SMLC improves MLC on two aspects. On one hand, S-MLC uses the PPR method to sample a subgraph for the query node. On the other hand, SMLC applies sparse NMF (SNMF) [51] method to learn the topological information of networks.

### B. Comparison of Synthetic Networks

We use the LFR benchmark [52] to evaluate the performance of C-MLC. The LFR networks simulate the characteristics of the real-world networks on heterogeneity of node degree and community size distributions. We produce 32 unweighted and undirected LFR networks by controlling the number of nodes $n$, the mixing parameter $\mu$, the number of memberships of the overlapping nodes $om$, and the number of overlapping nodes $on$. Specifically, we set $n$ at 5000 and 10000, $\mu$ at 0.1 and 0.3, $om$ at 2–5, and $on$ at 100 and 200, respectively. For each network, we randomly choose 100 nodes belonging to one community and 100 nodes belonging to $om$ communities as the query nodes. The whole parameter setting is shown in Table I.

The results on the LFR networks are given in Figs. 3–6. We can observe that C-MLC$_P$ or C-MLC$_H$ achieves all the highest $F_1$-score. The $F_1$-score declines with $om$ increases because the nearby structure is more complex when the query node belongs to more communities. Therefore, the algorithms are hard to distinguish the different communities. Moreover, most methods are influenced by the mixing parameter $\mu$ to a large extent. It is because the external degree of community increases with the growth of $\mu$, so that the community is more difficult to detect when $\mu$ is bigger. However, C-MLC$_H$ is barely affected because the method of HK is robust for $\mu$ to uncover the dense structure around the node. Besides, we can also observe that $n$ and $on$ have little influence on all the algorithms.

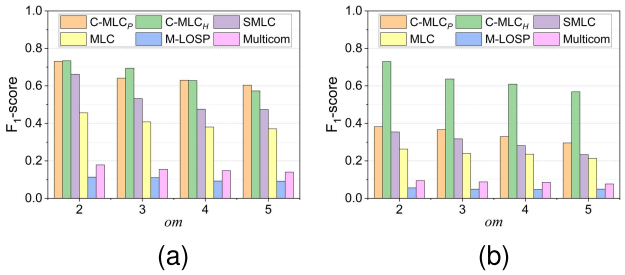


Fig. 3. Comparison of the LFR networks with $n = 5000$ and $on = 100$. (a) $\mu = 0.1$. (b) $\mu = 0.3$.

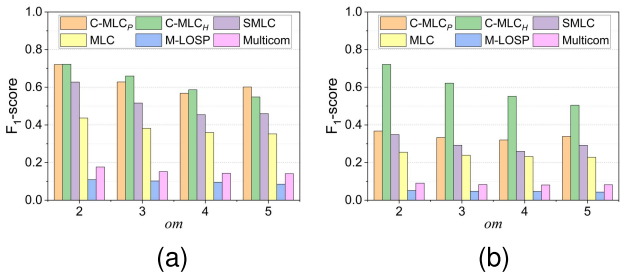


Fig. 4. Comparison of the LFR networks with $n = 5000$ and $on = 200$. (a) $\mu = 0.1$. (b) $\mu = 0.3$.

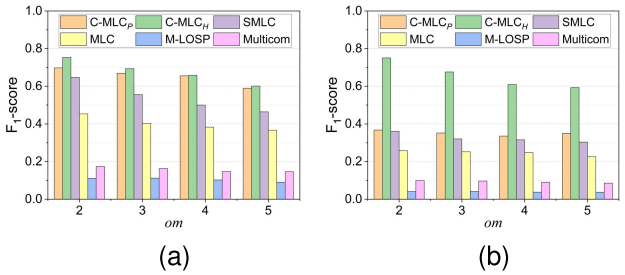


Fig. 5. Comparison of LFR networks with $n = 10,000$ and $on = 100$. (a) $\mu = 0.1$. (b) $\mu = 0.3$.

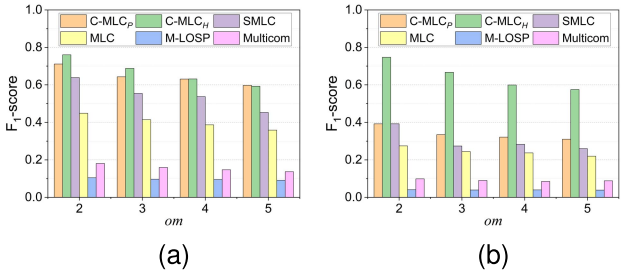


Fig. 6. Comparison of the LFR networks with $n = 10,000$ and $on = 200$. (a) $\mu = 0.1$. (b) $\mu = 0.3$.

In addition, SMLC has the best performance among the baselines. It is because SMLC uses the PPR method to sample a subgraph and the SNMF method to cluster the subgraph. MLC is worse than SMLC mainly because of using the BFS method as the sampling strategy, illustrating the importance of the sampling strategy for local community detection. M-LOSP fails to count the number of potential communities because when the query node locates in the dense structure, M-LOSP

TABLE II
STATISTICS OF THE REAL-WORLD NETWORKS

| Network | $n$ | $m$ | $\|\mathcal{C}\|$ | $\mu$ |
|---|---|---|---|---|
| Amazon | 334,863 | 925,872 | 1,517 | 0.06 |
| DBLP | 317,080 | 1,049,866 | 4,945 | 0.25 |

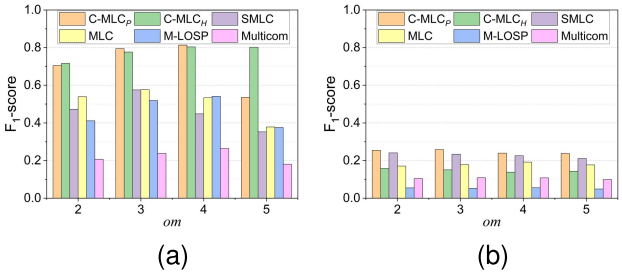


Fig. 7. Comparison of the real-world networks. (a) Amazon network. (b) DBLP network.

obtains one connected component by removing the query node. Multicom is limited since the number of community should be given as a hyperparameter.

### C. Comparison on Real-World Networks

We use two real-world networks with ground-truth communities, namely, Amazon and DBLP, to evaluate the performance of C-MLC. Amazon is a product network in which an edge between two products indicates that they have been purchased together, and DBLP is a coauthorship network where an edge between two authors indicates that they have coauthored a journal or conference paper. For both the networks, we use the top 5000 ground-truth communities. These datasets can be download from the Stanford Network Analysis Project.[3] In addition, we remove the identical communities and the communities whose sizes are bigger than 1000. For each $om \in [1, 5]$ in the networks, we randomly choose 100 nodes as query nodes. The statistics of the networks are given in Table II, where $n$ is the number of nodes, $m$ represents the number of edges, $|\mathcal{C}|$ denotes the number of ground-truth communities, and $\mu$ indicates the mixing parameter.

The results on the real-world networks are given in Fig. 7. We can observe that C-MLC$_P$ or C-MLC$_H$ achieves all the highest $F_1$-score on the Amazon network, while the results become much worse on the DBLP network. It is because there are too many super nodes (i.e., the nodes having a huge number of neighbors) and the topological structure is sparse on the DBLP network, resulting that the localized metric of C-MLC is difficult to lead the paths to the cores of communities. Besides, since SMLC and MLC use the NMF method, the results of SMLC and MLC are close to that of C-MLC$_P$ because the NMF method clusters the network using a wider range of topological information than C-MLC. Therefore, SMLC achieves higher $F_1$-score than C-MLC when $om$ is less than or equal to 2. However, with the growth of $om$, the topological structure around the query node becomes dense. Then, C-MLC can accurately find the centroid nodes of communities using the localized metric. Consequently, C-MLC$_P$ is more effective than SMLC when $om$ is bigger than 2. In addition, differing from the LFR networks, since the community structures of the real-world networks are highly irregular, the $F_1$-scores on the real-world networks do not decline consistently with the growth of $om$.

[3]http://snap.stanford.edu/

TABLE III
CONDUCTANCE COMPARISON OF THE DETECTED COMMUNITIES

| Network | $om$ | Ground-truth | C-MLC$_P$ | C-MLC$_H$ |
|---|---|---|---|---|
| Amazon | 1 | 0.0555 | 0.0380 | 0.0305 |
| | 2 | 0.0527 | 0.0246 | 0.0198 |
| | 3 | 0.0487 | 0.0173 | 0.0167 |
| | 4 | 0.0508 | 0.0163 | 0.0176 |
| | 5 | 0.0451 | 0.0322 | 0.0080 |
| DBLP | 1 | 0.4636 | 0.2407 | 0.1102 |
| | 2 | 0.4803 | 0.2457 | 0.1107 |
| | 3 | 0.5180 | 0.2795 | 0.1184 |
| | 4 | 0.5410 | 0.2874 | 0.1321 |
| | 5 | 0.5526 | 0.2784 | 0.1156 |

Furthermore, to analyze the reason why C-MLC$_H$ performs much worse than C-MLC$_P$ on DBLP, we count the average conductance of the ground-truth communities and the detected communities obtained by C-MLC$_P$ and C-MLC$_H$ on Amazon and DBLP, respectively. The results are shown in Table III. We can observe that the conductance of the ground-truth communities on DBLP is much higher than on Amazon. Since both PPR and HK uncover local communities based on conductance, C-MLC is difficult to find the community structure on DBLP. Besides, the conductance of the communities obtained by C-MLC$_H$ is far from the conductance of the ground-truth communities on DBLP. Therefore, C-MLC$_H$ is less effective than C-MLC$_P$ on DBLP.

## VI. CONCLUSION

In this work, we address the problem of multiple local community detection in social networks. Differing from typical local community detection approaches that uncover the local communities directly from the query node, we assume that every community contains a centroid node and first search the centroid node of each potential community. Thereafter, we propose a new algorithm called C-MLC to uncover multiple communities related to the query node. First, C-MLC explores the paths from the query node to find the centroid nodes using a localized metric. Then, for each centroid node, C-MLC combines the nodes along the paths as a set of initial seeds and applies PPR or HK to generate each community. The main advantages are that C-MLC can automatically determine the number of communities for the query node and uncover the communities based on the sets of high-quality seeds. Extensive experiments show that C-MLC outperforms other baselines significantly on the real-world networks and synthetic networks.

## REFERENCES


[1] L. Yu, G. Li, and L. Yuan, "Compatible influence maximization in online social networks," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 4, pp. 1008–1019, Aug. 2022.

[2] E. Cesario, C. Comito, and D. Talia, "An approach for the discovery and validation of urban mobility patterns," *Pervas. Mobile Comput.*, vol. 42, pp. 77–92, Dec. 2017.

[3] S. Rahiminejad, M. R. Maurya, and S. Subramaniam, "Topological and functional comparison of community detection algorithms in biological networks," *BMC Bioinf.*, vol. 20, no. 1, pp. 1–25, Dec. 2019.

[4] Z. Bu, C. Zhang, Z. Xia, and J. Wang, "A fast parallel modularity optimization algorithm (FPMQA) for community detection in online social network," *Knowl.-Based Syst.*, vol. 50, pp. 246–259, Sep. 2013.

[5] V. N. Ioannidis, A. S. Zamzam, G. B. Giannakis, and N. D. Sidiropoulos, "Coupled graphs and tensor factorization for recommender systems and community detection," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 909–920, Mar. 2021.

[6] C. Comito, C. Pizzuti, and N. Procopio, "Online clustering for topic detection in social data streams," in *Proc. IEEE 28th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2016, pp. 362–369.

[7] W. Zhao, J. Luo, T. Fan, Y. Ren, and Y. Xia, "Analyzing and visualizing scientific research collaboration network with core node evaluation and community detection based on network embedding," *Pattern Recognit. Lett.*, vol. 144, pp. 54–60, Apr. 2021.

[8] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3, pp. 75–174, Jan. 2010.

[9] V. Moscato and G. Sperlì, "A survey about community detection over on-line social and heterogeneous information networks," *Knowl. Based Syst.*, vol. 224, pp. 107–112, Jul. 2021.

[10] C. Comito, "How COVID-19 information spread in U.S.? The role of Twitter as early indicator of epidemics," *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1193–1205, May 2022.

[11] X. Luo, Z. Liu, M. Shang, J. Lou, and M. Zhou, "Highly-accurate community detection via pointwise mutual information-incorporated symmetric non-negative matrix factorization," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 463–476, Jan. 2021.

[12] M. Hajiabadi, H. Zare, and H. Bobarshad, "IEDC: An integrated approach for overlapping and non-overlapping community detection," *Knowl.-Based Syst.*, vol. 123, pp. 188–199, May 2017.

[13] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[14] V. Lyzinski, M. Tang, A. Athreya, Y. Park, and C. E. Priebe, "Community detection and classification in hierarchical stochastic blockmodels," *IEEE Trans. Netw. Sci. Eng.*, vol. 4, no. 1, pp. 13–26, Jan./Mar. 2017.

[15] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using Bayesian non-negative matrix factorization," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 6, Jun. 2011, Art. no. 066114.

[16] K. He, P. Shi, D. Bindel, and J. E. Hopcroft, "Krylov subspace approximation for local community detection in large networks," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 5, pp. 1–30, Oct. 2019.

[17] X. Ding, J. Zhang, and J. Yang, "A robust two-stage algorithm for local community detection," *Knowl.-Based Syst.*, vol. 152, pp. 188–199, Jul. 2018.

[18] P. Shi, K. He, D. Bindel, and J. E. Hopcroft, "Local Lanczos spectral approximation for community detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2017, pp. 651–667.

[19] K. He, Y. Sun, D. Bindel, J. Hopcroft, and Y. Li, "Detecting overlapping communities from local spectral subspaces," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2015, pp. 769–774.

[20] A. Hollocou, T. Bonald, and M. Lelarge, "Multiple local community detection," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 45, no. 3, pp. 76–83, 2018.

[21] L. Ni, W. Luo, W. Zhu, and B. Hua, "Local overlapping community detection," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 1, pp. 1–25, 2019.

[22] D. Kamuhanda and K. He, "A nonnegative matrix factorization approach for multiple local community detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2018, pp. 642–649.

[23] D. Kamuhanda, M. Wang, and K. He, "Sparse nonnegative matrix factorization for multiple-local-community detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 5, pp. 1220–1233, Oct. 2020.

[24] X. Bai, P. Yang, and X. Shi, "An overlapping community detection algorithm based on density peaks," *Neurocomputing*, vol. 226, pp. 7–15, Feb. 2017.

[25] W. Zhi-Xiao, L. Ze-chao, D. Xiao-fang, and T. Jin-hui, "Overlapping community detection based on node location analysis," *Knowl.-Based Syst.*, vol. 105, pp. 225–235, Aug. 2016.

[26] T. Magelinski, M. Bartulovic, and K. M. Carley, "Measuring node contribution to community structure with modularity vitality," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 707–723, Jan. 2021.

[27] B.-B. Xiang, Z.-K. Bao, C. Ma, X. Zhang, H.-S. Chen, and H.-F. Zhang, "A unified method of detecting core-periphery structure and community structure in networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 28, no. 1, Jan. 2018, Art. no. 013122.

[28] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using PageRank vectors," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2006, pp. 475–486.

[29] K. Kloster and D. F. Gleich, "Heat kernel based community detection," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 1386–1395.

[30] A. Clauset, "Finding local community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 2, p. 26132, Aug. 2005.

[31] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," *Web Intell. Agent Syst., Int. J.*, vol. 6, no. 4, pp. 387–400, 2008.

[32] J. Chen, O. Zaïane, and R. Goebel, "Local community identification in social networks," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, 2009, pp. 237–242.

[33] W. Luo, D. Zhang, L. Ni, and N. Lu, "Multiscale local community detection in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1102–1112, Mar. 2021.

[34] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 555–564.

[35] Y. Zhang et al., "SEAL: Learning heuristics for community detection with generative adversarial networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 1103–1113.

[36] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 27, no. 5, pp. 2672–2680.

[37] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.

[38] F. Moradi, T. Olovsson, and P. Tsigas, "A local seed selection algorithm for overlapping community detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 1–8.

[39] F. Jiang, S. Jin, Y. Wu, and J. Xu, "A uniform framework for community detection via influence maximization in social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 27–32.

[40] Y. Bian, J. Huan, D. Dou, and X. Zhang, "Rethinking local community detection: Query nodes replacement," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 930–935.

[41] Y. Li, K. He, D. Bindel, and J. E. Hopcroft, "Uncovering the small community structure in large networks: A local spectral approach," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 658–668.

[42] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[43] Y. Zhang and D.-Y. Yeung, "Overlapping community detection via bounded nonnegative matrix tri-factorization," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 606–614.

[44] D. Li, X. Zhong, Z. Dou, M. Gong, and X. Ma, "Detecting dynamic community by fusing network embedding and nonnegative matrix factorization," *Knowl.-Based Syst.*, vol. 221, Jun. 2021, Art. no. 106961.

[45] W. Luo, Z. Yan, C. Bu, and D. Zhang, "Community detection by fuzzy relations," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 2, pp. 478–492, Apr. 2020.

[46] W. Luo, D. Zhang, H. Jiang, L. Ni, and Y. Hu, "Local community detection with the dynamic membership function," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3136–3150, Oct. 2018.

[47] K. He, Y. Li, S. Soundarajan, and J. E. Hopcroft, "Hidden community detection in social networks," *Inf. Sci.*, vol. 425, pp. 92–106, Jan. 2018.

[48] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM Comput. Surv.*, vol. 49, no. 4, p. 69, Feb. 2017.

[49] M. A. Tabarzad and A. Hamzeh, "A heuristic local community detection method (HLCD)," *Appl. Intell.*, vol. 46, no. 1, pp. 62–78, Jan. 2017.

[50] D. F. Gleich and C. Seshadhri, "Vertex neighborhoods, low conductance cuts, and good seeds for local community methods," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 597–605.

[51] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.
[52] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.

**Boyu Li** received the M.S. and Ph.D. degrees in computer science and technology from Jilin University, Changchun, China, in 2014 and 2018, respectively.

He is currently a Post-Doctoral Researcher with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. His research interests include privacy-preserving data publishing, social networks, and graph embedding.

**Dany Kamuhanda** received the bachelor's degree in computer science from the Kigali Institute of Education, Kigali, Rwanda, in 2010, the master's degree in computer science and information systems from Nelson Mandela Metropolitan University, Gqeberha, South Africa, in 2015, and the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2019.

He is currently an Assistant Lecturer of computer science with the University of Rwanda, Kigali. His research interests include machine learning and community detection in social networks.

**Kun He** (Senior Member, IEEE) received the Ph.D. degree in system engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2006.

She is currently a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology. Her research interests include deep learning, machine learning, social networks, and intelligent optimization.