An Improved Label Initialization based Label Propagation Method for Detecting Graph Clusters in Complex Networks

Jyothimon Chandran, V Madhu Viswanatham
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, Tamilnadu, India

***Abstract*—Community structure is one of the fundamental characteristics of complex networks. Detection of community structure can provide insight into the structural and functional organization that helps to understand various dynamical processes such as epidemics and information spreading. Label propagation algorithm (LPA) is a well-known method for community structure identification due to linear time complexity. However, the communities extracted by the LPA is unstable since it produces different combinations of communities at each run on the same network. In this paper, a novel label initialization method for label propagation algorithm (ILI-LPA) is proposed to detect stable and accurate community structures. The proposed ILI-LPA focuses on more accurate label initialization rather than assigning unique labels thereby reduce the effect of randomness in LPA. The experiments on several real-world and synthetic networks show that the ILI-LPA improves the quality and stability of communities compared to existing algorithms. The results also demonstrate that appropriate label initialization can significantly improve the performance of label propagation algorithms, and the stability has been improved up to 50-78% relative to the standard LPA.**

***Keywords*—Social networks; community detection; graph clustering; edge clustering coefficient; label initialization; triangle count**

I. Introduction

Complex systems can be modeled as networks, with nodes representing entities of the system and links between nodes denoting its relationships [1]. Such networks are usually termed complex networks and can explain the emergence of complex behavior of the system. Examples of such complex networks [2] are biological networks, citation networks, scientific collaboration networks, and social networks. A common and significant characteristic of complex networks is community structure or communities or clusters [3], such as bacterial communities in the microbial ecosystem and community mobility in urban transport systems. Community structure can provide an overview of the system in consideration, explain the underlying dynamics, and reveal the hidden relations among the entities. It is defined as groups of nodes in a network with dense internal connections inside the groups and fewer connections between the groups [4]. An interesting fact about communities is that the nodes that belong to a community exhibit similar characteristics or common properties that define the overall behavior of the network [5]. Detecting community structure has become an integral part of network analysis.

Several community detection algorithms exist in the literature, and they fall into optimization methods or heuristic methods. The optimization methods such as modularity maximization algorithms [6], [7], [8], spectral methods [9], [10], and evolutionary algorithms [11], [12] formulate an objective function and then estimate an optimal value to find community partitions. Modularity maximization-based methods [6], [7], [8] focus on locating the maximum modularity to extract communities. The spectral methods [9], [10] construct the Laplacian matrix of the network from its characteristic vectors by formulating a quadratic objective function to obtain communities. The methods such as whale optimization [11] and genetic algorithm [12] are evolutionary algorithms. They utilize evolutionary computations to evaluate the optimal value of the optimization function to find the communities. However, most of these algorithms are inappropriate for networks of very large in size because of high time complexity. Heuristic methods apply heuristic techniques to identify communities, which are more time efficient than optimization methods. The methods such as Infomap [13], Edge betweenness [3], [14], and Label propagation algorithm [15] are examples of heuristic community detection algorithms.

The label propagation algorithm (LPA) [15] is one of the computationally efficient community identification methods having time complexity linear ($O(n)$). The LPA consists of mainly two steps: label initialization, and label propagation. At the label initialization, unique labels are assigned to every node in the network. Then at the label propagation, node labels of every node are iteratively updated to the label of the maximum of its adjacent nodes. If more than one label satisfies the maximum criteria, then a random selection on the maximal label is considered. This iterative label update process continues until every node label is the same as its adjacent nodes maximum label. According to a random node order, nodes are processed in each iteration. Finally, the communities are extracted with respect to node labels. Due to the low time complexity, the LPA is a better choice for very large networks. However, the communities detected by the LPA on a network differ in each execution, which makes the algorithm unstable. This drawback prevents LPA from being widely used in practice.

A sample network with two communities with few connections between them is shown in Fig. 1. Nodes {1, 2, 3, 4, 5, 6} and the nodes {7, 8, 9, 10, 11, 12} constitute the first and second communities. Two communities are also

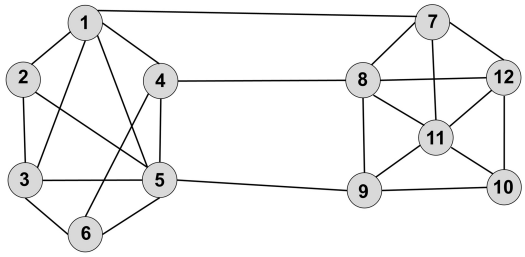

Fig. 1. A Sample Network with Two Communities, C_1 = {1, 2, 3, 4, 5, 6}, and C_2 = {7, 8, 9, 10, 11, 12}.

connected through edges (1,7), (4,8), and (5,9), and nodes $\{1, 4, 5, 7, 8, 9\}$ are called boundary nodes. Initially, node numbers are considered as node labels. While initiating the label update process, unique label initialization can cause multiple maximum labels for every node. For instance, if node 4 modifies its label with node 8, and node 11 subsequently changes with the label of node 8, and node 1 updates with node 4, then the algorithm returns a single community. In the next run, if the boundary nodes update their label with the label of nodes in their own community, then it produces two different communities. This is the instability problem of LPA.

To improve the stability of LPA, recently, many improvements have been proposed incorporating measures such as network modularity [16], [17], [18], [19], [20], node strength [21], [22], [23], [24], [25], [26], edge strength [27], [28], [29], [30], [31], [32] and other methods such as node attributes, memory constraints, and evolutionary approaches [33], [34]. Most of these LPA improvements assume that the leading cause of instability is the randomness involved in the label update process and the order of node. Hence eliminating the randomness by incorporating an order to the node selection and label update through various measures were received much attention. The modularity-based label propagation algorithms [16], [17], [18], [19], [20] update node labels according to the label of the neighbor node that produces largest modularity when multiple maximal labels exist. Similarly, node strength-based methods [21], [22], [23], [24], [25], [26] perform label selection according to the node strength measures such as node centrality, influence, or importance during the label update. Similarly, edge strength-based algorithms [27], [28], [29], [30], [31], [32] calculate the strength of connections using measures such as edge clustering coefficient, link strength to update the node label. Overall, these methods focus primarily on providing order to label update rule using various measures, thereby improving the stability of LPA. Though these researches have improved the performance of LPA, there still exists improvements in accuracy and stability.

This paper proposes an improved label propagation algorithm called Identical Label Initialization based LPA (ILI-LPA) based on a novel label initialization method for identifying community structure in networks. Instead of eliminating randomness, ILI-LPA focuses on proper label initialization to improve stability and accuracy. The label initialization of ILI-LPA is based on the measure *triangular structural influence (tsi)*, which estimates influence between nodes based on the triangles in the network. The *tsi* helps to find structurally closely connected nodes in the network to assign identical labels. Once the label initialization is over, the ILI-LPA follows the random label update strategy to extract the communities.

This paper contains the following contributions.

- Introduces *triangular structural influence (tsi)* to estimate the influence between nodes.
- Proposes a novel label initialization method to tackle stability.
- The effectiveness of ILI-LPA is tested on several real-world and synthetic networks.
- The effect of label initialization on reducing the impact of randomness is assessed.

The rest of the paper is organised as follows: Section II outlines the most recent enhancements to the label propagation algorithm that have been made. Section II elaborates the proposed method. Section IV discusses the experimental details such as data sets, baseline algorithms, and evaluation measures. The results and discussion are presented in section V. Section VI provides the conclusion and future works.

II. Related Work

A complex network is represented in this study by an unweighted undirected network $G(V, E)$, with V denoting the node-set and E denoting the edge set. The neighbor set of node u represents $\Gamma(u)$. If an edge connects two nodes, then they are called neighbors. The degree of node u is represented as d_u. If a node u, $(u \in V)$, contains a label, then it is denoted as l_u.

Several LPA improvements were proposed to enhance stability. According to the label update strategy, those LPA improvements can be classified into four. They are modularity-based LPA methods [16], [17], [18], [19], [20], node strength-based methods [21], [22], [23], [24], [25], [26], edge strength-based methods [27], [28], [29], [30], [31], [32], and other LPA improvements [33], [34]. These methods focus mainly on eliminating the randomness to improve the stability and accuracy of the communities produced.

A. Modularity-Based LPA Methods

To improve the stability, Barber and Clark [16] developed LPAm treating the LPA as a modularity optimization problem. According to LPAm, the label to be propagated is the label that increases the modularity. Compared to the original LPA, it improves the quality of the detected communities. This approach, however, has the problem of getting trapped in the local optimum, resulting in incorrect partitions. To avoid local maxima, Liu et al. [17] combined many community pairs at once utilizing a multistep greedy agglomerative algorithm and proposed LPAm+. Both LPAm and LPAm+ have a resolution limit problem due to the modularity function, which also increases the time complexity. To address the shortcomings of LPAm+, Le et al. [19] presented an improved LPAm+ algorithm called meta-heuristic-based LPA, which was based on the Record-to-Record Travel algorithm. This algorithm improves modularity prior to community merging. Another improved LPA called Stepping LPA-S was proposed by Li et al. [18] in which labels are propagated based on similarity. The stepping

LPA-S picks the label that results in the highest modularity. A modularity gain acceleration method based on modularity was introduced in [20] by formulating an objective function. The objective function is solved using global and local sum weights. Each nodes label transition is computed using local sum and general sum is calculated for each label. However, because of the modularity function, the time complexity of the above-mentioned algorithms is significantly higher than the other LPA improvements. Therefore, these are unsuitable for very large-scale networks.

B. Node Strength-Based LPA Methods

The idea of node strength-based label propagation algorithms is that when multiple labels satisfy the maximum criteria, instead of a random selection, the label of the node with the highest importance is chosen to overcome the instability problem. More crucially, calculating each node's importance in the network is the main task of these methods. Xing et al. [21] put forward the NIBLPA method utilizing the k-shell value to determine which label to update. The influence of nodes is assessed by examining the nodes degree and k-shell value along with its neighbours k-shell values. Subsequently, Zhang et al. [22] proposed LPA_NI, which considers both node importance and label influence. LPA_NI first estimates node importance using both the node's priori influence and the degree and the influence of its neighbors. The algorithm computes the influence of each label and updates the node label with the most influential label. Tasgin and Bingol [23] presented a local approach based on label propagation for detecting communities via boundary node identification. This approach first finds and rank the boundary nodes. Subsequently, the label of the node that has the largest score among its neighbours is spread. A method (NI-LPA) based on node importance was suggested in [24]. The node importance was calculated considering each node's signal propagation capability, Jaccard distance and k-shell value. However, the time complexity is increased to $O(n^2)$. The paper [25] employed label importance and proposed a label importance-based LPA (LILPA). The label update process in LILPA depends on the importance and attraction of nodes and label importance. The LILPA follows a fixed node order in which the nodes are arranged according to node importance, calculated from using closeness and degree of nodes. Incorporating the modularity and node significance, Li et al. [26] presented an enhanced algorithm called LPA_MNI. It begins by initializing each node with a unique community. Following that, a rough community is built for every node based on modularity gain by merging each node with its neighbor community in descending order of node importance until no further improvement is possible. The node strength is quantified using normalized degree centrality.

C. Edge Strength-Based Label Propagation Methods

Some researchers considered the strength of connections between nodes (edges/links) rather than the node importance to identifying community structure in networks. Based on edge strength, Lou et al. [27] introduced LPA_CNP algorithm. To begin, this method calculates the weighted coherent neighborhood propinquity for each pair of nodes to reflect the chance that two vertices are members of the same community. A node's label is updated to the label with the highest weighted-CNP. The results indicate that LPA_CNP outperforms LPA, particularly in large-scale networks. Zhang et al. [28] suggested an edge clustering-based LPAc algorithm. It first calculates the edge clustering coefficients of every edge in the network. During label propagation, this strategy selects the label of largest edge clustering coefficient edge that connects the neighbor. According to the link influence and node strength, Berahmand and Bouyer [29] presented LP_LPA. This approach initially determines the similarity of links between nodes assuming that nodes within a community share more common neighbors than nodes in other communities. Therefore, node strength is also estimated according to degree centrality, and initial node selection is performed by calculating node strength. Jokar and Mosleh [30] proposed BLDLP, which determines the weight for each edge according to the link density. If nodes' maximum labels are not unique, the largest weight edge label is chosen. Jiang et al. [31] introduced a link similarity measure and proposed LLPA in which node labels were computed from the link weights to identify functional modules. Li et al. [32] studied the network's higher-order properties by determining the most representative triangle motif that encoded the strength of connections and presented a community recognition approach based on Motif-Aware Weighted Label propagation. As a result, a unique voting approach termed NaS is presented to reduce the randomness provided by tie-breaking.

D. Other Label Propagation Methods

Hosseini and Rezvanian [33] introduced the AntLP based on ant colony optimization (ACO). The algorithm begins by weighting all the edges employing a combination of similarity indexes. Then, it attempts to spread labels by grouping comparable vertices to optimize modularity of each community according to their similarity of vertices. Berahmand et al. [34] proposed SAS-LP algorithm, an improved LPA algorithm for attributed graphs that addresses issues of instability and low quality while maintaining structural cohesiveness and attribute homogeneity in the detected communities.

Many of the LPA improvements, discussed in Sections 2.3 and 2.4, recommend strategies or techniques on label update procedure and node order, and follow a standard unique label initialization to improve the stability. The significance of label initialization on improving the stability and accuracy still remains an open problem. The aim of the paper is also on evaluating the impact of identical label initialization instead of unique label initialization on the stability and accuracy of LPA.

III. Proposed Method: ILI-LPA

This section proposes a novel label initialization method for LPA to extract community structure in networks. The proposed algorithm is named Identical Label Initialization based Label Propagation Algorithm (ILI-LPA). Different from the standard LPA and its improvements described in section 2, the ILI-LPA focuses on appropriate label initialization to identify stabile and accurate communities.

The effect of an appropriate label initialization is illustrated in Fig. 2. Assume that some of the nodes in each community is assigned with same labels. The label of nodes in each set

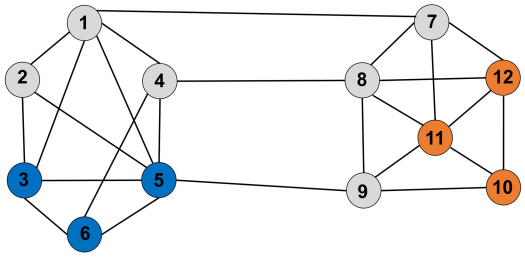

Fig. 2. A Simple Network with Two Communities in which the Nodes {3, 5, 6} Share a Common Label, Nodes {10, 11, 12} also Share another Label and the Remaining Nodes have different Labels.

{3, 5, 6} and {10, 11, 12} is same and the remaining nodes ({1, 2, 4, 7, 8, 9}) carries unique labels. Since some of the nodes in each community is assigned with the same labels, the stability of the label propagation improves significantly. This is because the boundary nodes ({1, 4, 5} and {7, 8, 9} can never update their label to the node labels that lie in other community based on the label propagation rule. One can see that nodes {1, 4, 5} can never update their labels to the label of other community nodes because their maximum neighbors' labels lie within the community. This applies to the nodes {7, 8, 9} also. when the random label update is applied. The main idea of this paper is to find such nodes that possess a high probability of joining a single community and assigning the same labels to them, thereby improving stability and accuracy.

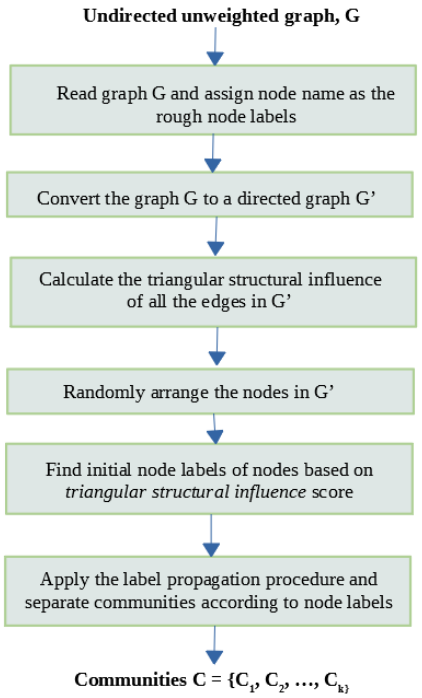

Fig. 3. The Block Diagram the of the Proposed ILI-LPA Method.

The proposed algorithm consists of mainly two phases: identical label initialization and label propagation. At the label initialization phase, ILI-LPA finds structurally closely connected nodes and assigns identical labels to them. When a node is connected with most of the neighbors of a neighbor node, then the node is said to be structurally closely connected to the neighbor node. To find the nodes that are structurally closely connected, a local measure called *triangular structural influence (tsi)* is introduced from the idea of edge clustering coefficient [36]. Then, according to the *tsi*, the label of the most influential node is given to its structurally closely connected neighbors to initialize node labels. Once the label initialization is over, the ILI-LPA performs the label propagation in which node labels are updated to the neighbors' maximal label. If there exist multiple maximum labels, then a random label selection strategy has opted. Each steps of the proposed ILI-LPA is provided in Fig. 3.

A. Identical Label Initialization

At the label initialization phase, the ILI-LPA aims to assign the same labels to structurally closely connected nodes. It can be measured by estimating the strength of connections between nodes. Several similarity measures exist in the literature that quantifies the connection strength between nodes. These measures quantify similarity considering the network structure or topology to reveal the strength of connections. It includes:

Definition 1: Cosine similarity [35] defined for node i and j is:

$$CS(i,j) = \frac{|\Gamma(i) \cap \Gamma(j)|}{\sqrt{d_i.d_j}} \quad (1)$$

Definition 2: Jaccard Similarity [35] defined for node i and j is:

$$JS(i,j) = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|} \quad (2)$$

Definition 3: Sørensen Index [35] defined for node i and j is:

$$SS(i,j) = \frac{2.\Gamma(i) \cap \Gamma(j)}{d_i + d_j} \quad (3)$$

Definition 4: Hub Depressed Index [35] defined for node i and j is:

$$HDI(i,j) = \frac{|\Gamma(i) \cap \Gamma(j)|}{max\{d_i, d_j\}} \quad (4)$$

Definition 5: Hub Promoted Index (HPI) [35] defined for node i and j is:

$$HPI(i,j) = \frac{|\Gamma(i) \cup \Gamma(j)|}{min\{d_i, d_j\}} \quad (5)$$

Definition 6: Edge clustering coefficient (ECC) [36] is:

$$ECC(i,j) = \frac{|\Gamma(i) \cup \Gamma(j)| + 1}{min\{d_i - 1, d_j - 1\}} \quad (6)$$

where d_i and d_j indicate degrees of node i and j, $\Gamma(i)$ signifies the neighbors of node i, $|\Gamma(i) \cap \Gamma(j)|$ estimates the number of common nodes. However, these measures are unidirectional, which means that they assume the influence between nodes are equal. In reality, the strength between nodes is bidirectional. Therefore, a new measure is introduced to measure the influence between nodes.

These similarity measures estimate the (structural) strength of connections between nodes accurately. More importantly, it is also known that high similarity value between nodes indicates same community participation of nodes. However, when the (structural) strength of relationship between nodes are considered, one can see that node strength from node i to j and vice versa may not be always same. If an edge that connects a pair of nodes is densely connected by their neighbors, then the edge clustering coefficient of that edge will be comparatively larger. This shows that the nodes exhibit high probability to lie in the same community. From the idea of the edge clustering coefficient, we introduce *triangular structural influence (tsi)* to quantify the node strengths. The tsi estimates the strength of the relationship (influence) between nodes based on the triangles associated with each node.

Definition 8: *triangular structural influence tsi(i, j)* denotes the influence the node i exerts on node j. If (i, j) is an edge in the network, $tsi(i, j)$ is the ratio of the actual number of connections from node i to the neighbors of j to the maximum possible connections. Therefore $tsi(i, j)$ is defined as:

$$tsi(i,j) = \frac{1 + |\Gamma(i) \cap \Gamma(j)|}{|\Gamma(j)|} \tag{7}$$

where $\Gamma(i)$ represents the neighbor set of i, $|\Gamma(j)|$ denotes the number of neighbors of node i (degree of i), $|\Gamma(i) \cap \Gamma(j)|$ indicates the number of common neighbors of node i and j, i.e., it represents the number of triangles that connects node i and j. If a node i is connected to all the neighbors of node j, then there exists high influence from i to j.

Similarly, the tsi from node i to node j cannot be same as the tsi from node j to node i. Therefore $tsi(j, i)$ is:

$$tsi(j,i) = \frac{1 + |\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i)|} \tag{8}$$

The densely connected nodes express high tsi than loosely connected nodes. Thus, it is clear that the nodes in a community exhibit higher tsi between nodes and less influence between communities.

Once the tsi between nodes are estimated, unique rough labels are assigned to every node in the network. Subsequently, in order to assign initial labels (9) is used. During the identical label initialization, each node's label is updated to the label of its neighbors based on the tsi using (9).

$$l_j^{init} = \underset{i \in \Gamma(j)}{\arg\max}\, L(l_i, l_j).f(i,j) \tag{9}$$

where l_j^{init} denotes the initial label of node j, $f(i, j)$ is a function that returns 1 if $tsi(i, j) \geq tsi(j, i)$ and $tsi(i, j) \geq \Theta$. L denotes the label. The Equation (9) can also be interpreted that every node in the network try to update the label of each of its neighbor node with its own label, if $f(i, j)$ is satisfied.

B. Label Propagation

During the label propagation, each node's label is updated asynchronously at random to the label shared by the majority of its neighbors. The proposed method (ILI-LPA) updates the labels in the same way as the standard LPA does. The every node label is updated using (10).

$$L_i = \arg\max_l |\Gamma^l(j)| \tag{10}$$

where $\Gamma^l(i)$ denotes the neighbors of node i with label l. Communities are formed through an iterative process in which densely connected groups of nodes reach consensus on a single label. Finally, the method converges when there are no more changes to the nodes' labels. If there exists more than one maximal label, the ILI-LPA follows the same LPA strategy, in which ties are broken randomly. Finally, nodes are categorized according to the node labels. That is, nodes with same labels join the same community. The Algorithm 1 describes the procedure of ILI-LPA in detail.

Algorithm 1 The Proposed ILI-LPA

Input: Undirected network $G = (V, E)$, parameter θ
Output: Communities C = $\{C_1, C_2, C_3, \ldots C_k\}$

1: **procedure** ILI-LPA(G,Θ)
2: Read network G ▷ Phase 1: Label Initialization
3: Assign rough unique labels to every node in V
4: Convert the network G to directed network G' by adding directions
5: **for** each edge in G' **do**
6: Calculate *tsi* of the edge
7: Attach it as edge weights
8: **end for**
9: Arrange the nodes in V in random order
10: **for** each node u in G' **do**
11: **for** each node $v \in \Gamma(u)$ **do**
12: **if** $tsi(u, v) \geq tsi(v, u)$ and $tsi(u, v) \geq \Theta$ **then**
13: update the label of v with the label of u
14: **end if**
15: **end for**
16: **end for**
17: Removes the directions and weights of G'
18: Set t=1 ▷ Phase 2: Label propagation
19: Arrange the nodes V in random order and set it to V'
20: **for** each node u in V' **do**
21: update its label according to equation (10)
22: **if** there exists more than one maximum label **then**
23: randomly update to the label of maximum of its neighbors.
24: **end if**
25: **end for**
26: goto step 27 if none of the node label changes, else set t= t+1, go to step 19
27: According to the node label, separate the communities.
28: Return communities C.
29: **end procedure**

The algorithm first assigns unique rough labels all the nodes. In step 8, it converts the input (undirected) graph to a directed graph by adding directions to all edges. Then, at step 5-8, the tsi of each edge is computed using (7) and (8) and attach it as corresponding edge weights. Subsequently, each node tries to spread its label to each of its neighbors and update the neighbor's label with its label if the tsi from the node to its neighbor is greater than the neighbor back

to the node. The label spread and label update is performed sequentially in an asynchronous fashion to the entire nodes in the network at once. It is performed in step 18-24. Before that, the algorithm converts the network to an undirected network and remove the edge weights and retain only the node labels. The network contains nodes with identical labels in densely connected regions. This process is followed by the label propagation to find final communities. At step 23, node labels are updated according to the maximum label of their neighbors.

The main steps of ILI-LPA is illustrated in Fig. 4 with the support of a toy network. Fig. 4 (a) shows a network that contains three communities {1, 2, 3, 4, 5, 6}, {7, 8, 9, 10, 11}, {12, 13, 14, 15, 16, 17, 18}. Figure 3 (b) shows the estimated tsi to between nodes and marked at the ends of each edge which represents the tsi to that node. The initial community labels identified by the proposed method by the label initialization is represented in Fig. 4 (c), where the value associated with each node indicates that the node label. Fig. 4 (d) indicates the communities identified after the label propagation. The nodes and their corresponding community labels are given to express the node and its updated nodes label.

C. Complexity Analysis

The ILI-LPA contains mainly two phases. The major steps in phase 1 are unique label initialization and *triangular structural influence* estimation. It takes $O(n)$ time to initialize rough unique labels to every node in the network. The time complexity of estimating the tsi of every edge in both directions is $O(m.d_{avg})$ where d_{avg} and m denote the average degree and the number of edges. The label propagation takes $O(m)$ time. Thus the overall time complexity of ILI-LPA is $O(m.d_{avg})$ + $O(n)$ + $O(m)$, which is approximately equal to $O(m.d_{avg}) \approx O(n.d)$ ($d \ll n$).

IV. Experimental Details

The performance of the ILI-LPA was tested on synthetic networks and real networks. Experiments were carried out on a 3.4 GHz Intel Core i7 CPU with 16.0 GB of RAM. Python-Networkx was used to develop the algorithm. The algorithm's input parameter θ is set at 0.35 on all the networks.

A. Datasets

1) Real-World Networks: The networks considered in this study are: Karate Club [37], Dolphin network [38], Football [3], Polbooks [39], Netscience [40], Email Enron [41], Cond-mat-2003 [41], Cond-mat-2005 [41], DBLP [42], Amazon [42]. The details of these networks are provided in Table I [N_C: actual communities in the network, d_{avg}: average degree, CC: Clustering coefficient].

B. Synthetic Networks

Lancichinetti-Fortunato-Radicchi (LFR) [43] is a popular synthetic network generator to test the performance of community detection algorithms. The community size and degree distributions of generated networks follow power-law distributions. The LFR generator contains the number of nodes

TABLE I. Details of the Real-World Networks

Dataset	Nodes	Edges	NC	d_{avg}	CC
Karate Club	34	78	2	4.58	0.58
Dolphin	62	159	2	5.13	0.30
Football	115	613	12	10.66	0.40
Polbooks	105	441	3	8.40	0.49
Netscience	1589	2742	-	3.451	0.878
Email-enron	36692	183831	-	10.02	0.716
Cond-mat-2003	31163	120029	-	7.703	0.723
Cond-mat-2005	40421	175692	-	8.693	0.719
Amazon	334863	925872	-	5.530	0.396
DBLP	317080	1049866	-	6.662	0.632

TABLE II. The Parameter Values of the LFR Network

Network name	N	k	$kmax$	$cmax$	t_1	t_2	μ
LFR_net1	1000	20	100	100	2	1	0.05 – 0.75
LFR_net2	5000	20	500	500	2	1	0.05 – 0.75
LFR_net3	10000	20	1000	1000	2	1	0.05 – 0.75
LFR_net4	20000	20	2000	2000	2	1	0.05 – 0.75

(N), maximum degree ($kmax$), maximum community size ($cmax$), average degree (k), degree distribution exponent (t_1), community size distribution exponent (t_2).The most important parameter that sets the character of communities is the mixing parameter μ, which indicates the percentage of linkages between communities and within communities. Table II shows the parameter values given to generate LFR network.

C. Baseline Algorithms

The proposed algorithm was compared with seven existing algorithms. They are Fastgreedy [6], Louvain [7], Infomap [13], LPA [15], NIBLPA [21], LPA-CNP-E [27] and Stepping-LPA [18].

D. Evaluation Metrics

Modularity: For assessing the quality of community partitions, the modularity [44] metric is widely used. Modularity (Q) indicates the percentage of edges within the communities minus the expected percentage of community edges of a random network with same degree distribution. The modularity value of communities C is computed using (11).

$$Q(C) = \frac{1}{2m}\sum(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j) \qquad (11)$$

where m signifies the total edges, A denotes the adjacency matrix representation of the network, A_{ij} = 1 if node i and j are connected, 0 otherwise. The k_i and k_j indicate the degrees of node i and j. The C_i and C_j denote the communities of nodes i and j. The Kronecker delta (δ) yields 1 when nodes i and j belong to a single community, otherwise, it returns 0.

Normalized Mutual Information (NMI): The NMI [45] measures how similar the communities are to one other, by comparing the communities extracted by the community detection algorithm to the actual communities. Let A and B be the actual and detected communities of a given network. The NMI (A, B) is computed using (12).

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij} log\frac{N_{ij}N}{N_i N_j}}{\sum_{i=1}^{C_A} N_i log\frac{N_i}{N}} \qquad (12)$$

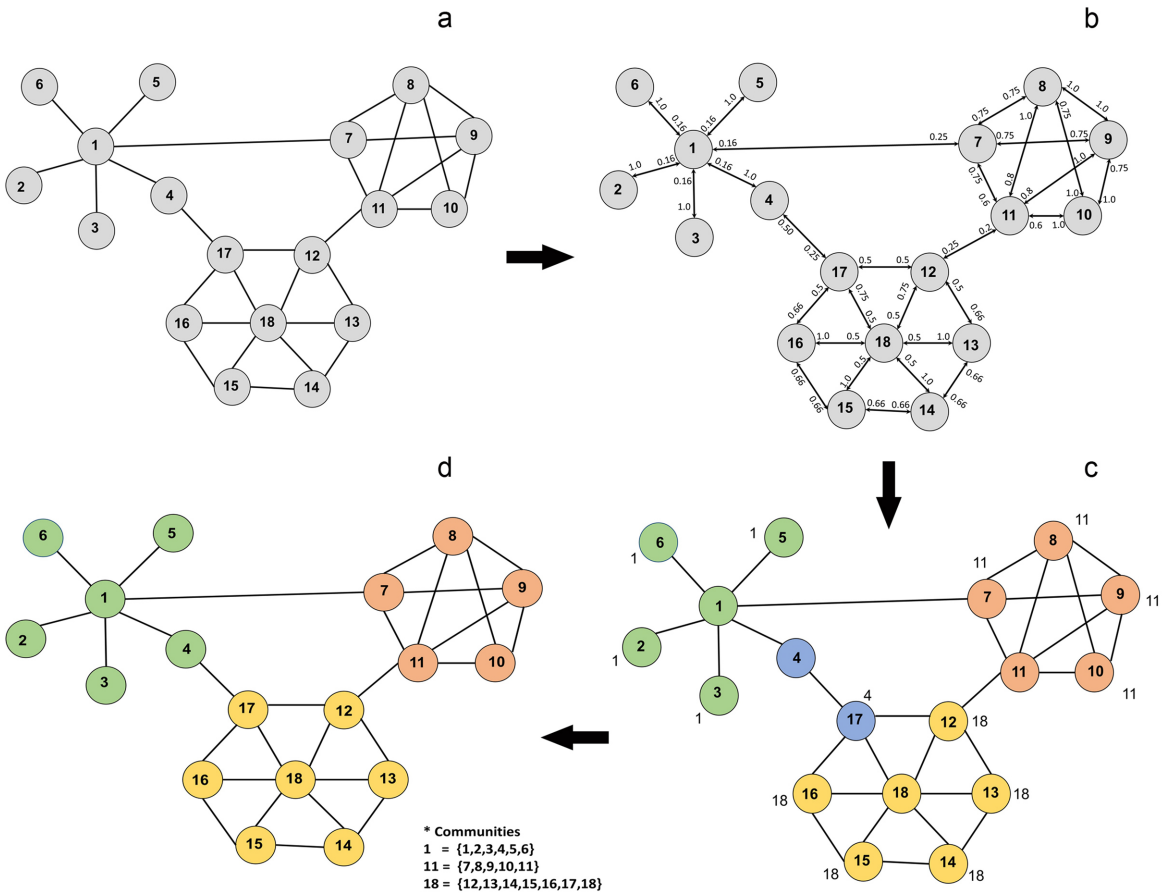

Fig. 4. Details of each Major Steps of ILI-LPA with an Example. a) Input Network, b) Triangular Structural Influence as Weights to each Edge, c) Initial Labels of the Nodes, d) the Final Extracted Communities.

where N_{ij} is the number of common nodes of A's community i and B's community j. The actual and discovered number of communities are denoted as C_A and C_B, respectively. NMI produces its maximum value of 1 if the discovered partition is identical to the actual partitions. If the two partitions are not related, the NMI returns 0.

V. Experimental Result and Discussion

A. Comparing the Modularity of Algorithms on Small Networks

The modularity of the ILI-LPA and the compared algorithms on small networks is presented in Table III. All algorithms were run 100 times, and calculated the average and standard deviation. From Table 3, we can see that ILI-LPA produces significantly better modularity in comparison with baseline algorithms, including the Louvain method on Dolphin and Football networks. On Polbooks network, the ILI-LPA algorithm produces modularity value of 0.526, which is closer to the modularity of Louvain. Though the obtained average modularity of ILI-LPA is only 0.371 on Karate dataset, but still, it is better than standard LPA.

TABLE III. Modularity of Algorithms on Small Real-World Networks

Algorithms	Karate	Dolphin	Football	Polbooks
Fastgreedy	0.381	0.495	0.568	0.502
Louvain	**0.415**	0.519	**0.604**	**0.527**
Infomap	**0.415**	0.520	0.563	0.512
LPA	0.357	0.487	0.589	0.511
LPA-CNP-E	0.303	0.463	0.601	0.451
Step-LPA-S	0.371	0.378	0.575	0.496
NIBLPA	0.40	0.43	0.50	0.55
ILI-LPA	0.371±0.00	**0.523±0.05**	**0.604±0.02**	**0.526±0.002**

B. Comparing the NMI of Algorithms on Small Networks

To evaluate the accuracy of algorithms, the NMI is calculated and reported in Table IV. It provides that on Football and Polbooks networks, the NMI score of ILI-LPA is comparatively higher compared to baseline algorithms. Though the modularity value of Dolphin network shown in Table 3 is high, the accuracy is low, with NMI value 0.566. On Karate network, ILI-LPA algorithm yields an NMI score of 0.837, which is just below the NMI of Step-LPA-S and better than other methods, including Louvain. From Tables III and IV, we can say that the ILI-LPA is better than the others on stability and accuracy on small networks.

TABLE IV. NORMALIZED MUTUAL INFORMATION OF ALGORITHMS ON REAL-WORLD NETWORKS

Algorithms	Karate	Dolphin	Football	Polbooks
Fastgreedy	0.693	0.573	0.744	0.439
Louvain	0.707	0.474	0.885	0.418
Infomap	0.707	0.563	0.921	0.467
LPA	0.649	0.540	0.893	0.524
LPA-CNP-E	0.837	0.731	0.909	0.571
Step-LPA-S	**0.924**	**0.888**	0.925	0.571
NIBLPA	0.58	0.50	0.72	0.53
ILI-LPA	0.837	0.566	**0.927**	**0.593**

TABLE V. MODULARITY OF ALGORITHMS ON LARGE REAL-WORLD NETWORKS

Algorithms	Net science	Email-enron	Cond-mat -2003	Cond-mat -2005	Amazon	DBLP
Fastgreedy	0.955	0.510	0.678	0.631	0.879	0.728
Louvain	0.959	0.605	0.761	0.722	0.910	0.810
Infomap	0.931	0.527	0.661	0.631	0.232	0.714
LPA	0.912	0.337	0.592	0.620	0.784	0.634
LPA-CNP-E	0.932	0.512	0.736	0.631	-	-
Step-LPA-S	0.921	0.531	0.694	0.625	-	-
NIBLPA	0.68	0.12	0.50	0.23	0.67	0.61
ILI-LPA	0.921	0.562	0.632	0.645	0.813	0.653

C. Comparing the Modularity of Algorithms on Large Networks

Table V provides the modularity produced by the algorithms on large networks. As seen on the Table, the modularity value of the proposed algorithm on the network science dataset is 0.921, significantly better than LPA, Step-LPA-S, NIBLPA. On Email-enron network, the ILI-LPA gives 0.562 modularity, which is higher than all the algorithms except the Louvain method. On Cond-mat-2003 dataset, our algorithm is not performing well because the modularity of the detected communities is just 0.632. At the same time, On Cond-mat-2005 network, the modularity of the proposed method is superior to other algorithms except for Louvain. On Amazon, only Fastgreedy and Louvain produces superior modularity than the proposed method. On DBLP network, though our algorithm is inferior to non-LPA-based algorithms, still better than both LPA and NIBLPA. The experiments on large-scale networks compared to LPA-based (LPA, LPA-CNP-E, StepLPA-S, NIBLPA) and non-LPA-based (Fastgreedy, Louvain, Infomap) algorithms on modularity metric show that the ILI-LPA has better performance on LPA-based algorithms and is closer to non-LPA based algorithms except Louvain. Since Louvain is a modularity optimization method, Louvain can return higher modularity on most of the network. Table 5 demonstrates that the ILI-LPA performs well on large-scale real-world networks.

D. Evaluating the Performance ILI-LPA on LFR Networks

Extensive tests have been carried out on the LFR network in order to validate the performance of the ILI-LPA. It is analyzed on the LFR network in three different aspects: modularity, NMI, and the number of communities. Since the actual community information is available in the LFR network, the actual modularity and number of communities of the corresponding network are considered as GroundTruth value. The algorithms employed for the comparison are Fastgreedy, Louvain, Infomap, and LPA. Since non-LPA algorithms are better than LPA variants, only these four algorithms are considered for synthetic network evaluation. The results with respect to Modularity (Q) (including the GroundTruth modularity of LFR communities) is illustrated in Fig. 5. With an increase in the mixing parameter (μ), the accuracy of algorithms steadily diminishes. Fig. 5 shows that the modularity of the communities identified by the proposed algorithm is the same as that of the GroundTruth modularity of LFR until reaches 0.70. Though the modularity of Fastgreedy is stable, it is significantly lower than the GroundTruth.

Fig. 5 shows that on all the four networks, the ILI-LPA produces modularity closer to the GroundTruth and better than the other LPA methods. When the μ is less than 0.45, the modularity of detected communities of different algorithms, except the Fastgreedy, is closer to GroundTruth modularity on all four networks. Both LPA and Infomap modularity drastically reduce when at 0.50 and 0.55, respectively. However, ILI-LPA is the same as GroundTruth and Louvain communities until the reaches 0.70. when the modularity of standard LPA drops between 0.40 and 0.50, the proposed improved LPA (ILI-LPA) algorithm maintains the quality of communities till reaches 0.70. In all four networks in Fig. 5, the ILI-LPA shows a similar pattern of modularity and better performance than other algorithms.

The experimental result with respect to NMI is reported in Fig. 6. The results indicate that except the FastGreedy, all algorithms produce good performance on all the four networks until is 0.45. With respect to the increase in the mixing parameter (μ), the difficulty in community identification also increases. Fig. 6 shows that the proposed method produces stable and accurate communities on all the four LFR networks until is less than 0.7. The performance of Fastgreedy and Infomap significantly decreases when is above 0.4, and the performance of LPA also drastically drops when crosses 0.5. The experiments on four LFR networks demonstrate that the ILI-LPA is improved in terms of algorithms than compared algorithms.

Additionally, while evaluating the performance of community detection methods, the number of communities detected is a significant performance metric to consider. In LFR networks, since the actual number of communities is known, the comparison can provide more insights into the performance. The number of communities produced by the algorithms corresponding to four different mu values is illustrated in Fig. 7. In LFR net1, the number of communities of the ILI-LPA is very close to the GroundTruth communities of the LFR network in all the four different μ values. On all four networks, the number of communities of Fastgreedy is significantly low than the GroundTruth values. While μ is greater than 0.4, as expected the both Infomap and LPA yield poor performance. There is only one algorithm that produces an exact number of communities to the ground truth in all the test cases is our proposed ILI-LPA algorithm. Though the Louvain shows high modularity value, the identified community size of Louvain is significantly smaller than that of the actual number of communities. Overall, the experimental results illustrate that the ILI-LPA is stable and accurate in finding communities without consuming much computational time.

Analysis and Discussion The results illustrate that the ILI-LPA improves the stability and accuracy without significantly

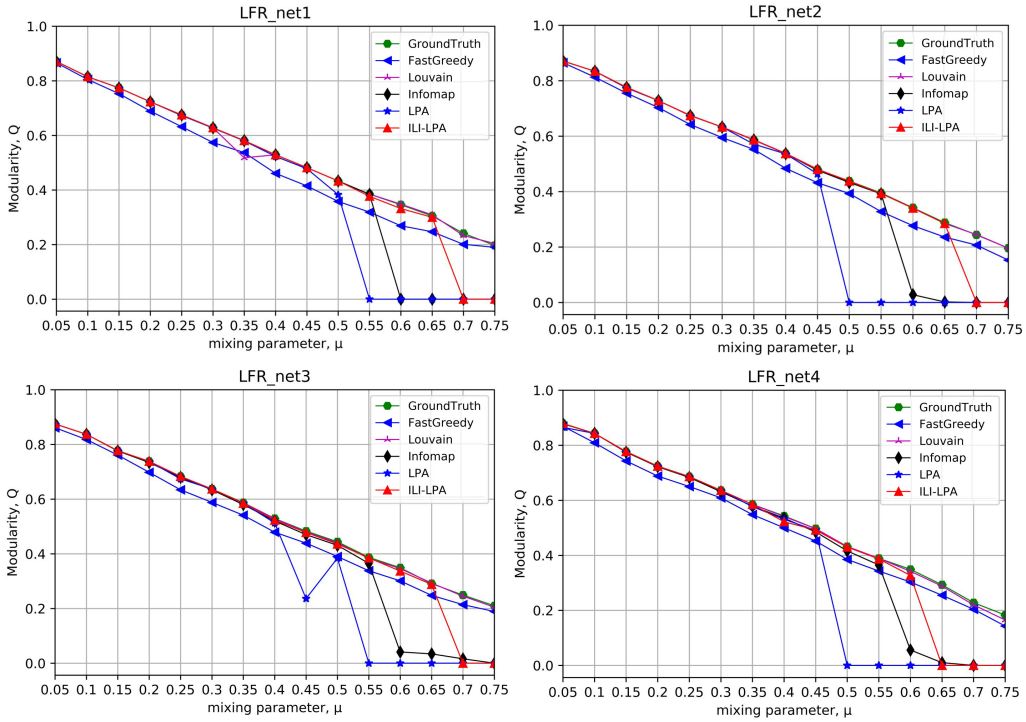

Fig. 5. Modularity of the Communities Detected by the Five Community Detection Algorithms along with the Actual Modularity (GroundTruth) on Four LFR Networks.

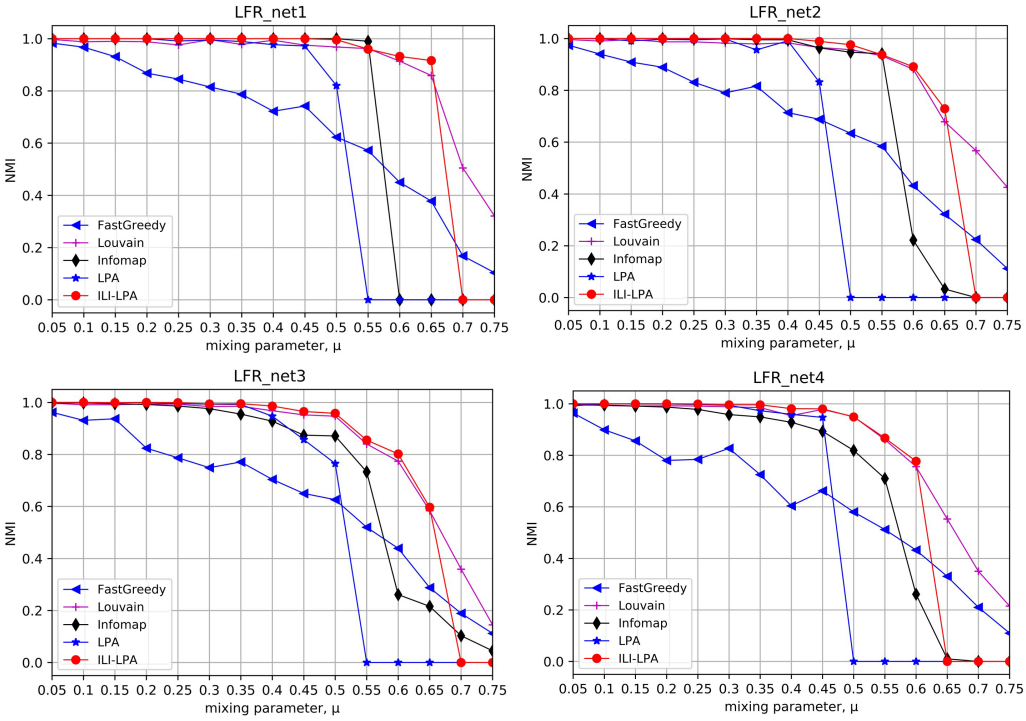

Fig. 6. NMI Produced by the Algorithms on Four LFR Networks.

increasing the execution time. Unlike the standard and other improvements of LPA that assign nodes with unique labels during the label initialization, the ILI-LPA focuses on identical label initialization where two nodes get the same label if the nodes are structurally closely connected. That is, the two nodes expresses a high probability to continue in a single community

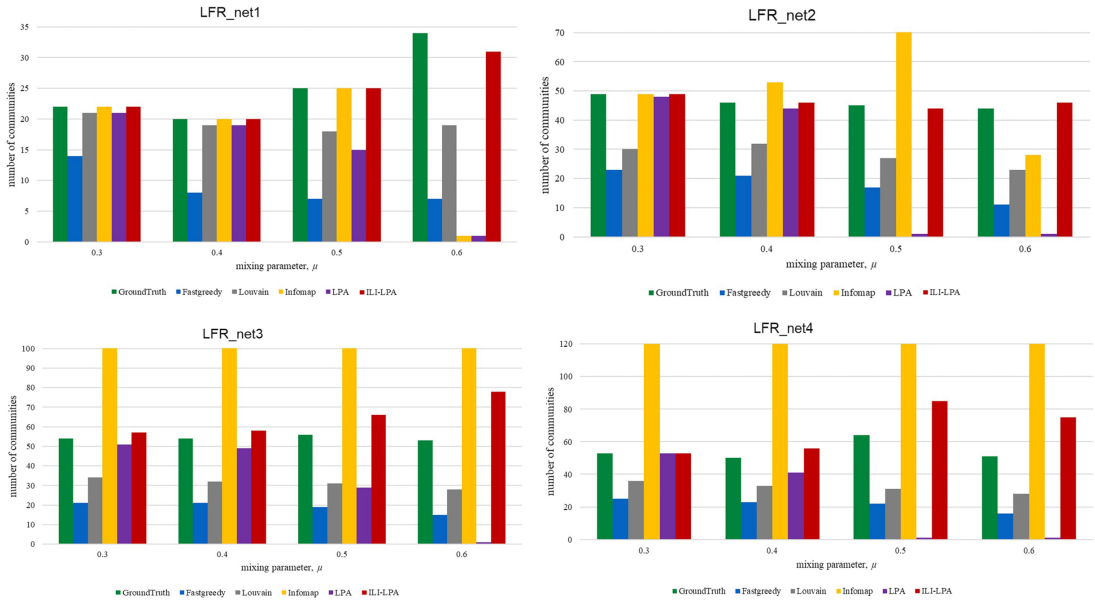

Fig. 7. The Number of Communities Produced by the Algorithms on Four LFR Networks.

from the initialization to the final communities during the label propagation. Real-world and synthetic networks are employed for conducting the experiments. The results prove the importance of more accurate label initialization than the conventional unique label initialization to improve the stability and accuracy of improved LPAs. The main advantage of ILI-LPA is that the identical initialization of labels reduces number of iterations at the label propagation phase. Also, the label initialization helps each node to differentiate its own community neighbors from other neighbors, which solves the instability due to random selection. So that, without eliminating the randomness in LPA and employing proper label initialization, the ILI-LPA achieves better performance.

VI. Conclusion and Future Work

The LPA is a popular time-efficient community detection algorithm. However, the instability in results is its main drawback. Many of the recent LPA improvements concentrate primarily on eliminating randomness by introducing various measures that provide an order to the label update process. However, these improvements either increase the computational time or reduce the accuracy. This paper presents a novel technique called identical label initialization to improve stability and accuracy and proposes the ILI-LPA. The ILI-LPA finds nodes that are structurally closely connected, then assigns identical labels to them. Our approach focuses primarily on proper label initialization rather than assigning unique labels. The ILI-LPA maintains the random label selection strategy when multiple maximal labels exist to update node labels. The results demonstrate that the proposed ILI-LPA has better performance than the existing algorithms. The results also demonstrate that proper label initialization is also an important factor for improving LPA stability and accuracy. In future, the proposed method can be extended for finding overlapping communities in networks. In addition, the label initialization can be extended to detect evolving communities in dynamic networks.

References

[1] M. E. J. Newman, "The structure and function of complex networks," SIAM Review., vol. 45, no. 2, pp. 167–256, 2003.

[2] M. E. J. Newman, "Networks: An Introduction," J. Math. Sociol., 2013.

[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks.," Proc. Natl. Acad. Sci. U. S. A., vol. 99, no. 12, pp. 7821–7826, 2002.

[4] G. Xu, J. Guo, and P. Yang, "TNS-LPA: An Improved Label Propagation Algorithm for Community Detection Based on Two-Level Neighbourhood Similarity," IEEE Access, vol. 9, pp. 23526–23536, 2021.

[5] S. Kumar, L. Singhla, K. Jindal, K. Grover, and B. S. Panda, "IM-ELPR: Influence maximization in social networks using label propagation based community structure," Appl. Intell., vol. 51, pp. 7647–7665, 2021.

[6] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top., vol. 70, no. 6, p. 6, 2004.

[7] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech. Theory Exp., vol. 2008, no. 10, 2008.

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

[9] L. Huang, R. Li, H. Chen, X. Gu, K. Wen, and Y. Li, "Detecting network communities using regularized spectral clustering algorithm," Artif. Intell. Rev., vol. 41, pp. 579-594, 2014.

[10] Y. Li, K. He, K. Kloster, D. Bindel, and J. Hopcroft, "Local Spectral Clustering for Overlapping Community Detection," ACM Trans. Knowl. Discov. Data, vol. 12, no. 2, pp. 1–27, 2018.

[11] Y. Zhang et al., "WOCDA: A whale optimization based community detection algorithm," Phys. A Stat. Mech. its Appl., vol. 539, p. 122937, 2020.

[12] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," IEEE Trans. Evol. Comput., vol. 16, no. 3, pp. 418–430, 2012.

[13] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," Eur. Phys. J. Spec. Top., vol. 178, no. 1, pp. 13–23, 2009.

[14] M. Arasteh and S. Alizadeh, "A fast divisive community detection algorithm based on edge degree betweenness centrality," Appl. Intell., vol. 49, pp. 689–702, 2019.

[15] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., vol. 76, no. 3, pp. 1–11, 2007.

[16] M. J. Barber and J. W. Clark, "Detecting network communities by propagating labels under constraints," Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., vol. 80, no. 2, pp. 026129, 2009.

[17] X. Liu and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," Phys. A Stat. Mech. its Appl., vol. 389, no. 7, pp. 1493–1500, 2010.

[18] W. Li, C. Huang, M. Wang, and X. Chen, "Stepping community detection algorithm based on label propagation and similarity," Phys. A Stat. Mech. its Appl., vol. 432, pp. 145–155, 2017.

[19] B. D. Le, H. Shen, H. Nguyen, and N. Falkner, "Improved network community detection using meta-heuristic based label propagation," Appl. Intell., vol. 49, no. 4, pp. 1451–1466, 2019.

[20] S. Yazdanparast, M. Jamalabdolahi, and T. Havens, "Linear Time Community Detection by a Novel Modularity Gain Acceleration in Label Propagation," IEEE Trans. Big Data, vol. 7, no. 6, pp. 961–966, 2020.

[21] Y. Xing, F. Meng, Y. Zhou, M. Zhu, M. Shi, and G. Sun, "A node influence based label propagation algorithm for community detection in networks," Sci. World J., vol. 2014, 2014.

[22] X. K. Zhang, J. Ren, C. Song, J. Jia, and Q. Zhang, "Label propagation algorithm for community detection based on node importance and label influence," Phys. Lett. Sect. A Gen. At. Solid State Phys., vol. 381, no. 33, pp. 2691-2698, 2017.

[23] M. Tasgin and H. O. Bingol, "Community detection using boundary nodes in complex networks," Phys. A Stat. Mech. its Appl., vol. 513, pp. 315-324, 2019.

[24] T. Wang, S. Chen, X. Wang, and J. Wang, "Label propagation algorithm based on node importance," Phys. A Stat. Mech. its Appl., vol. 551, pp. 124137, 2020.

[25] Y. Zhang, Y. Liu, Q. Li, R. Jin, and C. Wen, "LILPA: A label importance based label propagation algorithm for community detection with application to core drug discovery," Neurocomputing, vol. 413, pp. 107–133, 2020.

[26] H. Li, R. Zhang, Z. Zhao, and X. Liu, "Lpa-mni: An improved label propagation algorithm based on modularity and node importance for community detection," Entropy, vol. 23, no.5, 2021.

[27] H. Lou, S. Li, and Y. Zhao, "Detecting community structure using label propagation with weighted coherent neighborhood propinquity," Phys. A Stat. Mech. its Appl., vol. 392, no. 14, pp. 3095-3105, 2013.

[28] X. K. Zhang, X. Tian, Y. N. Li, and C. Song, "Label propagation algorithm based on edge clustering coefficient for community detection in complex networks," Int. J. Mod. Phys. B, vol. 28, no.30, p. 1450216, 2014.

[29] K. Berahmand and A. Bouyer, "A Link-Based Similarity for Improving Community Detection Based on Label Propagation Algorithm," J. Syst. Sci. Complex., vol. 32, pp. 737-756, 2019.

[30] E. Jokar and M. Mosleh, "Community detection in social networks based on improved Label Propagation Algorithm and balanced link density," Physics Letters, Section A: General, Atomic and Solid State Physics. vol. 383, no. 8, pp. 718-727, 2019.

[31] H. Jiang et al., "A Robust Algorithm Based on Link Label Propagation for Identifying Functional Modules from Protein-protein Interaction Networks," IEEE/ACM Trans. Comput. Biol. Bioinforma., 2020.

[32] P. Z. Li, L. Huang, C. D. Wang, J. H. Lai, and D. Huang, "Community detection by motif-aware label propagation," ACM Trans. Knowl. Discov. Data, vol. 14, no. 2, pp. 1-19, 2020, doi: 10.1145/3378537.

[33] R. Hosseini and A. Rezvanian, "ANTLP: Ant-based label propagation algorithm for community detection in social networks," CAAI Trans. Intell. Technol., vol. 7, pp. 10, 2020.

[34] K. Berahmand, S. Haghani, M. Rostami, and Y. Li, "A new attributed graph clustering by using label propagation in complex networks," Journal of King Saud University - Computer and Information Sciences. vol. 34, no. 5, pp. 1869-1883, 2020.

[35] T. Zhou, L. Lü, and Y. C. Zhang, "Predicting missing links via local information," Eur. Phys. J. B, vol. 71, pp. 623-630, 2009.

[36] J. Wang, M. Li, H. Wang, and Y. Pan, "Identification of essential proteins based on edge clustering coefficient," IEEE/ACM Trans. Comput. Biol. Bioinforma., vol. 9, no. 4, pp. 1070–1080, 2012.

[37] W. W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," J. Anthropol. Res., vol. 33, no. 4, pp. 452–473, 1977.

[38] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," Behav. Ecol. Sociobiol., vol. 54, pp. 396–405, 2003.

[39] Krebs, V.: A network of co-purcheased books about US politics. October, vol. 20, no. 1, pp. 0-03, 2008.

[40] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., vol. 74, no. 3, pp. 36104-36123, 2006.

[41] M. E. J. Newman, "Network data", 2013, *http://www-personal. umich. edu/ mejn/netdata*.

[42] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," Knowl. Inf. Syst., vol. 42, pp. 181–213, 2015.

[43] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., vol. 78, no.4, pp. 46110-46115, 2008.

[44] M. E. J. Newman, "Modularity and community structure in networks," Proc. Natl. Acad. Sci. U. S. A., vol. 103, no. 23, pp. 8577-8582, 2006.

[45] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," J. Stat. Mech. Theory Exp., vol. 2005, no.09, pp. 9008, 2005.