ICIC Express Letters
Part B: Applications
Volume **12**, Number **10**, October **2021** ISSN 2185-2766
pp. **899–908**

AN EFFICIENT HYBRID ALGORITHM FOR COMMUNITY STRUCTURE DETECTION IN COMPLEX NETWORKS BASED ON NODE INFLUENCE

Phuong Nguyen Huy Pham[1], Bich-Ngan Thi Nguyen[1], Quy Thi Ngoc Co[1]
Nguyen Tu Nguyen[1], Phuoc Tran[2,*] and Václav Snášel[3]

[1]Faculty of Information Technology
Ho Chi Minh City University of Food Industry
140 Le Trong Tan Street, Tay Thanh Ward, Tan Phu District, Ho Chi Minh City 700000, Vietnam
{ phuongpnh; nganntb }@hufi.edu.vn; { quyctn.a3mk; tunguyen9a6 }@gmail.com

[2]Natural Language Processing and Knowledge Discovery Laboratory
Faculty of Information Technology
Ton Duc Thang University
19 Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City 700000, Vietnam
*Corresponding author: tranthanhphuoc@tdtu.edu.vn

[3]Faculty of Electrical Engineering and Computer Science
Technical University of Ostrava
17. Listopadu 2172/15, Ostrava-Poruba 70800, Czech Republic
vaclav.snasel@vsb.cz

Received March 2021; accepted May 2021

Abstract. *Community detection in complex networks has recently become considerable because it is possible to explore community structure, analyze behaviour and action. Detecting these communities brings enormous finance and provides informational value in the complex network. Many algorithms are proposed in previous studies for community structure detection problem. However, the majority of studies on community detection often have an element of randomization in their performance that leads to inconsistent results. In this paper, we proposed an efficient hybrid Algorithm for Community structure Detection in complex networks based on individual Node Influence (named CDANI) to detect community with stable results, and also detect overlapping nodes. Through simulations on synthetic network adapter and real-world datasets, the results show that our algorithm is innovated comparing to other community detection algorithms.*
Keywords: Community detection, Complex networks, Node influence, Overlapping nodes

1. **Introduction.** Complex networks have always been a classic topic for scientific works. Intensive researches about complex networks promote scientific development in many different fields through the different structures such as social networks, biological networks and cell networks. The most important issue as conducting a study on complex networks is to discover community structure. It helps to develop science in general and viral marketing in particular with finding communities of similar relationships such as interests, tendency, and personal habits. According to Fortunato [6], the community is a group of collected individuals on the Internet, and they were treated as a member of a group rather than as an individual. The entity has some similar properties and plays a role in a social network. Due to the complex structure and enormous scale of communities, understanding internal structure is a major challenge. A method to cope with analysing community structure problem is known as community detection. It reported internal organization and identified special existence among nodes that were not directly accessible

DOI: 10.24507/icicelb.12.10.899

or identifiable from the direct test following available experiences. The goal of the problem is to detect the community structures from given social networks and learns about the community relationships between communities and members, how those relationships affect the structure of the complex network. Following reasons mentioned, community detecting in complex networks does become a significant issue in the science of networks. Nodes in the same community must have some of the same characteristics and attributes.

To define community structure, we must analyze relationships in terms of geometrical structures among communities. Community structure was studied by Girvan and Newman in 2002 with the idea of using centrality indices to find community boundaries [7]. It is one of the earliest community detection algorithms, based on the idea that a network can be divided into communities by repeatedly removing edges between communities. Consequently, each edge in the network is valued with a metric called weight. The Newman-Girvan modularity, is a well-known measure of how well a community structure is, given in 2004 [17]. Unfortunately, Brandes et al. [14] proved that modularity optimization is NP-hard and so approximation algorithms are necessary when handling large networks. In addition, this topic is explored with modularity method such as [1, 11], which is based on modularity optimization using heuristic, is fast and scalable. However, modularity optimization methods may meet some problems like communities smaller than a predefined size may not be detected. The label propagation algorithm (LPA), which is first presented in Raghavan et al. [19], is considered to be one of the fastest algorithms in community detection with simple design and its timing complexity is almost linear. It can be generalized as follows: Each node was marked a unique label; through each iteration process, the nodes update its label to the label it most often encounters in random order from neighboring nodes. It means when there are the several of frequent labels, the node will randomly pick up to become its own label. It continues until each node's label is the most frequent of its neighbors. A node updates its label based on its neighbor's last iteration label and the labels from the other neighbors' current iteration. This process ends when the algorithm satisfies the stop condition. In the final step, community is defined as a group of nodes having same labels. Innovative studies have polished up LPA in various aspects, for example, as finding overlapping community [9]. Nevertheless, due to the randomness in choosing the order of neighboring nodes to update and randomly choosing nodes's label, the algorithm's stability was not guaranteed. This means that LPA could detect different communities in different runs. Sometimes, even small communities can be swallowed by large communities and being formatted as monster community. CDANI resolves this instability-problem by relegating random elements.

A summary of the previous studies is that they are not able to guarantee the stability among the final results. And in the most of the previous studies, the essence of those algorithms is greedy, clustering each node one by one and re-evaluating the graph after just only node merging such as [2, 4, 7, 12, 18]. To solve these problems, we introduce our algorithm in subject community detection based on node influence called CDANI for complex networks, which can not only detect communities in graphs, but also determine overlapping nodes. Firstly, network's topology information is used to measure the node intimacy. The node importance is evaluated based on the intimacy matrix to improve the stability and accuracy of the community detection by preventing as many as possible random selection processes. Sets of selected nodes will have descending by order of priority, according to the individual importance based on node influence. Secondly, we iteratively select the affected nodes again by the representative node sets to merge them into same communities. Finally, this process finishes when the influential value is greater than the allowed threshold to be merged. This threshold is predefined and scalable according to the size of networks.

The rest of this paper is organized as follows. We propose our algorithm along with its background, theoretical analysis and proofs in Section 2. In Section 3, we present the experimental evaluation and compared results with other previous detecting algorithms. Finally, Section 4 concludes the paper.

2. **Community Detection Algorithm.**

2.1. **Background and notion.**

Let $G(V, E)$ be a network that consists of set of nodes V and E which is set of edges that show connections between nodes. We define graphs $\mathcal{G}'(V', E')$ with extended set node V' inherited by $G(V)$ and set edges E' inherited by $G(E)$. Each v in V' includes extended attributes as child node set or the number of internal links. Edge $e \in E'$ is defined as a directed and weighted edge with weight computed by (5).

2.1.1. *Node intimacy.* Based on the community concept, two nodes in the network are intimate if they have a number of common neighbor nodes. Several node similarity measurements based on local topology information illustrated different efficiency for community structure detection in complex networks described in Zhou et al. [21]. Similarity measurements are used to evaluate the intimacy between each pair of nodes in the graph.

Jaccard Index (JI) [10] is a widely applied method basing on local topology information to detect community structures in complex networks. JI was defined by the following formula:

$$S_{(u,v)} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}, \quad \in [0; 1], \tag{1}$$

where $N(u)$ represents neighbors of nodes u and $N(v)$ represents neighbors of node v. $|N(u) \cap N(v)|$ shows the number of common neighbor nodes and $N(u) \cup N(v)$ is the quantity of both u and v. However, while evaluating node intimacy, JI has some limitations. JI underestimates the level of intimacy between connected nodes; it even returns zero if two nodes do not have mutual friends, which is nonsense.

To surmount the shortcomings of JI's local bridge problem, Eustace et al. [5] have introduced a more efficient function as follows:

$$I_{u \to v} = \frac{|(N(u) \cap N(v)) \cup \{v\} \cup \{u\}|}{|N(u) \cup \{u\}|} \tag{2}$$

When nodes u and v are connected by direct edge, the value of $I_{u \to v}$ is always greater than 0. The larger this value, the closer u is to v. Eustace has improved JI's zero-intimacy problem when estimating by the mutual neighbor nodes including u and v themselves. However, this algorithm only evaluates the intimacy value when there is a direct edge between two nodes, not to mention the influence of indirect edges.

Inheriting previous principles, we have overcome the disadvantages when evaluating the intimacy between two nodes or two supernodes in general. We improved the intimacy node formula above to the intimacy for groups of nodes (communities), in order to increase the accuracy when evaluating the intimacy on the relationship between the group nodes.

2.1.2. *Mergeability.*

Definition 2.1 (Mergeability). *To evaluate whether or not a community is willing to be merged, we use mergeability. The higher the mergeability value of one community is, the more reasonable it is to merge that community into another; on the other hand, the lower the merge value is, the less community needs to expand. We proposed a formula as follows:*

$$\mathbb{M}_u = \frac{\sqrt{\left(\frac{n'}{n+n'}\right)^2 + \left(\frac{m'}{m+m'}\right)^2}}{density(u)} \tag{3}$$

n, n' are considered as the number of child nodes and neighbor nodes; m, m' are the number of internal and external links of community C_i. The concept of density of a graph or a community is defined as a ratio between the existing edges within the graph to the maximum number of possible edges between the nodes. In our article, "density" is used as the worth-merging of an community and calculated using the following formula:

$$density(u) = \begin{cases} \dfrac{m}{n*(n-1)}, & (n > 1) \\ 1, & (n \le 1);\ density(u) \in (0;1] \end{cases} \tag{4}$$

where n is the number of child nodes in community and makes $n(n-1)$ become the maximum number of possible edges in the community, and m indicates internal edges of community. When the number of child nodes ≤ 1, then $density(u) = 1$, if there is no internal link causes $density = 0$. This leads to dividing by zero error for mergeability formula; hence, the default value is 1. Range of $\mathbb{M}$ values is $\left[0;\sqrt{2}\right]$ and $\mathbb{M}_u = 0$ when u is in a state without edges to any other node.

2.1.3. *Influence weight.* When a user v is followed by user u, we consider object u to be in the community v. This process continues with the other nodes, so the nodes with the same target are regarded as a community. Community of u following v is detected when v becomes the source of maximum influence on u. Influence weight is defined as follows:

$$F_{u\to v} = I_{u\to v} * \frac{\#edge}{\#outlink_u} * \mathbb{M}_v \tag{5}$$

2.1.4. *Individual importance.* We measure individual importance corresponding to node weight. The purpose of selecting these important nodes is to determine the priority to merge that node in the graph, in order to regenerate a graph with a less complicated structure, thence improving the accuracy in community detecting. Our value is computed as follows: $\text{II}_u = \sum_{v\in S, v\neq u} F_{u\to v}$.

2.1.5. *Overlapping.* In addition to the main detecting algorithm based on community structure, we proposed searching community overlapping theory in graph. In the classic detecting algorithm, each node is labeled to identify its own community. In reality, there are many circumstances where a node intrinsically belongs to many different communities.

Definition 2.2 (Overlapping)**.** *It is a detection method where a node could belong to more than one community. In community detecting problem, grouping overlapping entirely changes the structure of whole graph, which affects results in following steps.*

Currently, problems of finding and solving overlapping nodes are solved by skipping detecting overlapping; marking overlapping node and randomly assigning it to a group which it possibly belongs to or separating them for all the communities which meet conditions to contain overlapping nodes [8]. Following the overlapping detecting theory, we suggest the algorithm works as follows: when finding a pair of nodes with the best clustering priority value, source node will be grouped with target group node. By searching the out link of the source node having the highest influence weight value and reaching the threshold, we determine whether the source node can still be in any other group target node. When finding any satisfied groups, we immediately allocate source node to it.

We assume when a node u is merged into group v, then all edges between u and v will be synchronized as internal link and all edges between u to other nodes will be pointed to v as a representation. When allocating overlapping u to k node sets, we suppose u is split into k same subnodes u and put into k communities. The edge set of graph will increase k times the number of external links of node u, due to the number of cross edges between the k sets assuming node u is in this community. We solve the problem when communities with the same overlapping are grouped together and eliminate the virtual edges stemming from node overlapping caused by splitting. The purpose of this process is to return the intact data as the original graph with the number of nodes and edges unchanged.

2.1.6. *Community merging threshold.*

Definition 2.3 (Merging Threshold). *It is a given threshold to stop merging two clusters together, when the influence weight of a cluster to another cluster is higher than the threshold score expected, which means that when two clusters affect each other well, but together will reduce the quality of the cluster. In fact, in addition to the clustering condition of the existing algorithms, a different measure is necessary, which avoids community merging causing lower the quality of the clusters. We predefined it as the stop-condition threshold of our algorithm.*

We experimented several times on large and small datasets until a relatively suitable value was found to make the quality of detected community at the consequent range. After testing, we have found the formula to determine the most standard threshold value, which is presented as follows:

$$\theta_{u\to v} = \frac{\mathbb{M}_u}{\mathbb{M}_v} * \frac{\log_n(|C|) * \frac{\ln(n)}{n}}{\log_n(|\{u\} \cup \{v\}|)} * \begin{cases} \dfrac{1}{\log_{n^2}\left(\frac{\ln(|\{u\}|*|\{v\}|)}{\ln(n^2)}\right)}, & |\{u\}| * |\{v\}| \geq n \\ \dfrac{1}{\log_{n^2}\left(\frac{\ln(n^2)}{\ln(|\{u\}|*|\{v\}|)}\right)}, & 1 < |\{u\}| * |\{v\}| < n \end{cases} \tag{6}$$

2.2. **Proposed algorithm.** Community detection algorithms are divided into two types: disjoint community detection and overlapping community detection. CDANI algorithm in our study is proposed to detect overlapping communities in complex networks. Given graphs $G(V, E)$ including node set V and edge set E, our algorithm has an diagram shown in Figure 1. Based on deterministic graph G, we select independently each edge $(u, v) \in E$ with its own weight $w(u, v)$ to make *sample graph* $\mathcal{G} = (V', E')$.

2.3. **Complexity analysis.** Complexity of an algorithm is to evaluate step of effectiveness based on running time, memory usage or costs. Let network graph $G(V, E)$ have total n nodes and m edges. The time to parse $G(V, E)$ into $\mathcal{G} = (V', E')$ is $O(n)$('). Assuming x is the number of structured detecting loops, and k_x is the number of merged groups in certain loop, we have that $n' = n - k_x$ is the number of current groups, with $k_x < n$ (*). Time complexity to calculate influence weight by browsing all pairs of source-target nodes having direct edges with $n' * (n' - 1)$ edges is $O(n') = O(n') * O(n' - 1) = O(n'^2)$("). Expected running time to estimate $\amalg_u$ for all vertices is $O(n')$("'). Time complexity of finding pair of source-target nodes is $O(n') + O(n') = O(n')$. The complexity cost of finding target node with the highest $\amalg_u$ value is $O(n')$ and the cost for finding the source node of that target node by browsing the entry p of the target node is $O(p)$. Additionally, complexity of searching for more target's community is by traversing out neighbor node $O(p')$ (with p, $p' < n'$). Hence, we have time complexity of finding pair of source-target nodes is $O(n') * O(p) * O(p') = O(n' * p * p') = O(\max(n', p, p')^3) = O(n'^3)$.

Nevertheless, $O(n'^3)$ is extremely high so we applied a binary heap searching solution running in time $O(\log_2 x)$, then time to build the heap structure is $O(x * \log_2 x)$.

Proof: We have constraint $n' > 1$, then $\log_2 n' >= n$.

Time complexity to build heap for every node is $n' * O(n' \log_2 n') = O(n'^2 * \log_2 n')$. Therefore, finding the target node having highest $\amalg_u$ runs in time $O(\log_2 n')$. Similarly, the cost of finding source node is $O(\log_2 p)$ and finding target clusters of source node is $O(\log_2 p')$, and thus, expected time to find pair of node merging: $O(n'^2 * \log_2 n') < O(n'^3)$. □

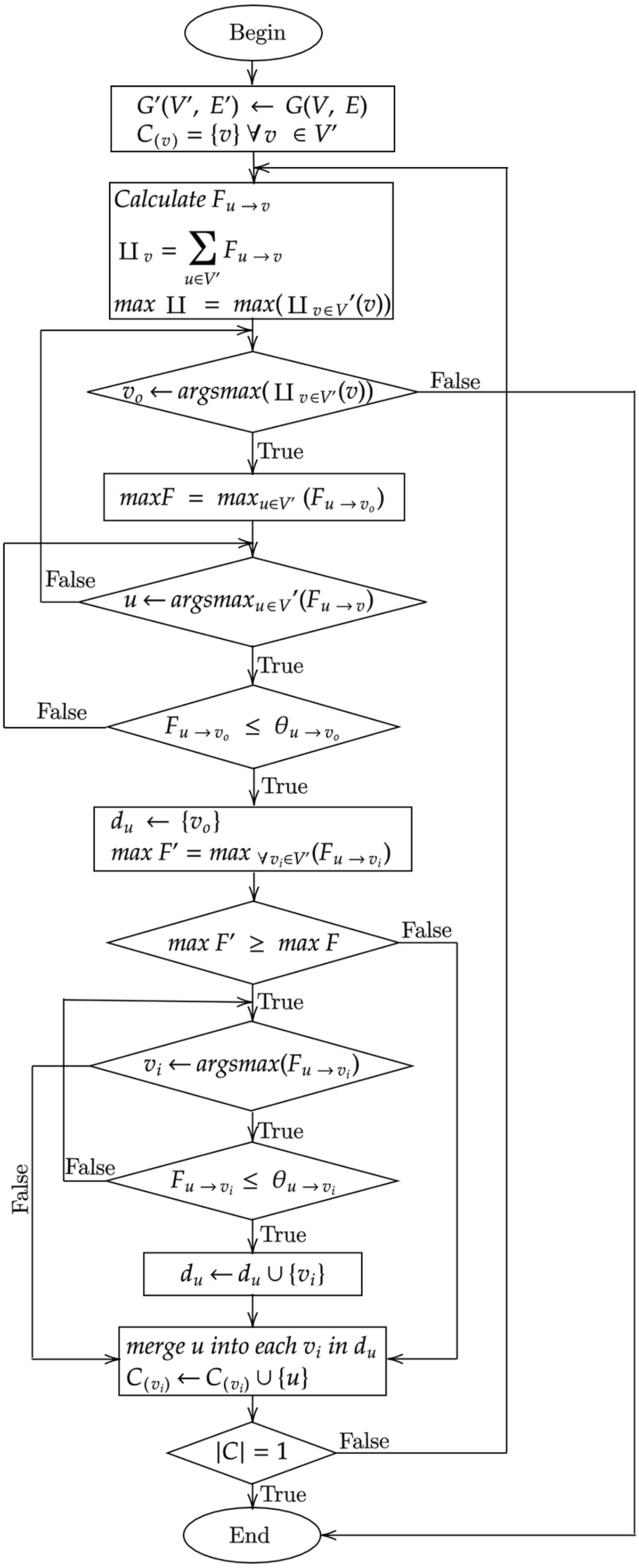

FIGURE 1. Diagram of CDANI

In the worst case, time of merging the source node into h groups while distribution overlapping is $O(h)$ with $(h < n)$ and the time to synchronize edges for p neighbors of h new communities is $h * O(p) = O(h * p)$ with $(p < n')$. Hence, total merging time is

$$O(h) + O(h * p) = O(\max(h, h * p)) = O(h * p) = O(n'^2) \tag{7}$$

After above analysis steps, we evaluate the complexity for single iteration of CDANI through proof in ("), ("'), *Proof*, (7) as follows:

$$O\left(n'^2\right) + O(n') + O\left(n'^2 * \log_2 n'\right) + O\left(n'^2\right) = O\left(n'^2 * \log_2 n'\right) \tag{8}$$

Additionally, execution time complexity of every x searching iterations in (*) is

$$\sum O\left(n'^2 * \log_2 n'\right) = \sum O\left((n - k_x)^2 * \log_2(n - k_x)\right) = O\left(n^2 \log_2 n\right)$$

From (') and (8), we have $O(n) + O\left(n^2 \log_2 n\right) = O\left(\max\left(n, n^2 \log_2 n\right)\right) = O\left(n^2 \log_2 n\right)$.

In summary, we compute the complexity of CDANI as a cost of $O\left(n^2 \log_2 n\right)$ for overlapping graph structure detecting. In terms of offering innovative solutions to improve our speed, we apply *k-heap* data structures to increasing detecting speed for large-scale data.

3. Experiment.

Through our experimental results of detecting community on real-world datasets, we compared our results with many other algorithms and then came to a conclusion about the optimization of CDANI. Our experiments are executed on a Linux machine with a 2 × Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz, 64GB RAM DDR4 @ 2400MHz. Our implementation is written in Python language.

3.1. Datasets.

We choose a set of datasets including Karate, Dolphins, Football, Political Books and Net-science to propose the experiment. Details of used datasets are described in Table 1. Graphs in Experiment section interpret community structure in the following format: each community has a distinctly particular color, separates from others. Overlapping nodes are highlighted in white, positioned in the middle of its communities.

Table 1. Datasets

Dataset	#Nodes	#Edges	Type
Karate [20]	34	78	Directed and Unweighted
Dolphins [15]	62	159	Directed and Unweighted
Football [7]	115	613	Directed and Unweighted
Political Books	105	411	Directed and Unweighted
Net-science [16]	1589	2742	Directed and Unweighted

The Zachary's Karate Club is a classical network in community detection. In our algorithm, it was detected with 4 communities. As shown in Figure 2, node 10 is the overlapping of the community colored in the community "34" and "3".

The Football Network [7] simulates the schedule of games between United States college football teams. In this dataset, each edge represents regular season matches between two connected teams. Our algorithm detected 9 communities within this network.

Lusseau et al. [15] conducted a study of connection between 62 dolphins living in the Doubt Sound Strait of New Zealand. In Figure 3, the network is divided into 10 groups in a real community structure and 9 overlapping nodes are detected.

PolBooks[1] is a network of American politics books published around the time of the 2004 presidential election and sold by Amazon.com. Nodes present books and edges between books represent the frequent purchase of the same book by the same buyer.

Net-science [16] is a dataset describing the collaborative network of 1589 scientists (network scientists) working on network theory and experiment from many different fields.

[1]http://www-personal.umich.edu/ mejn/netdata/.

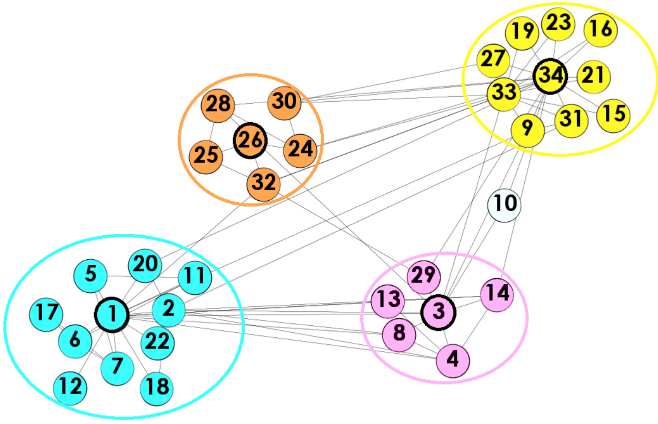

FIGURE 2. Detecting community with CDANI in Zachary's Karate Club

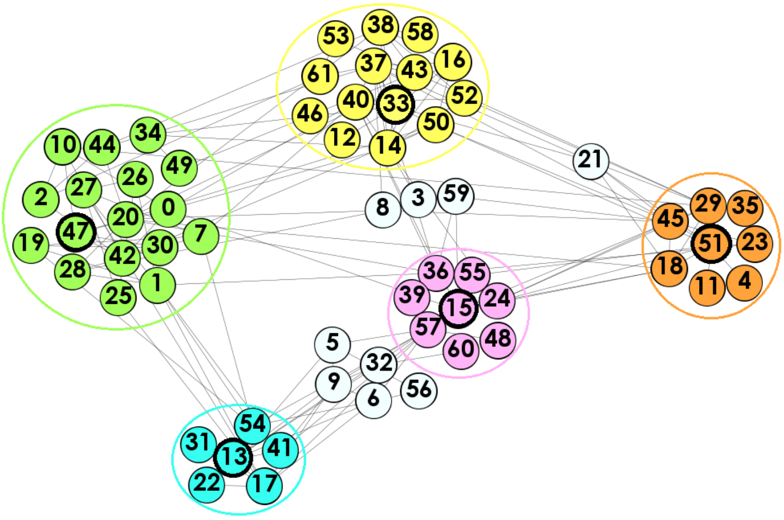

FIGURE 3. Detecting community with CDANI for Dolphins Network

3.2. **Evaluation.** We use Modularity [17], F1-score [13] and NMI [3] to compare detected communities. Modularity is a classical measure to evaluate the quality of network's partition. Dividing a network into k community, element e_{ij} shows edges which connect one node in community C_i to another in C_j. Modularity Q is defined as follows:

$$Q = \frac{1}{2m}\sum_{uv}\left[A_{uv} - \frac{k_v k_w}{2m}\right]\frac{s_u s_v + 1}{2}, \text{ while } e_{ij} = \sum_{uv}\frac{A_{uv}}{2m} \tag{9}$$

Modularity's returned value is limited by $[0;1]$. The higher value Q, the better performance and quality of the community structure detected by that algorithm is. Commonly, it is in the range of $[0.3 : 0.7]$. NMI (Normalized Mutual Information) [3] is proposed to calculate the similarity between the ground-truth community structure and the community structure found by our algorithm.

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log\left(\frac{C_{ij}n}{C_{i.}C_{.j}}\right)}{\sum_{i=1}^{C_A} C_{i.}\log\left(\frac{C_{i.}}{n}\right) + \sum_{j=1}^{C_B} C_{.j}\log\left(\frac{C_{.j}}{n}\right)} \tag{10}$$

Given a discrete community set C generated by algorithm and groundtruth community set S, $precision = \frac{|C \cap S|}{C}$ and $recall = \frac{|C \cap S|}{S}$, F1-score [13] is defined as follows:

$$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{11}$$

3.3. **Result.** After the experiment and obtained results on above networks, we give some general comments about the experimental results and the algorithm we suggest. We apply algorithms above to calculating the quality of our algorithm with the results of two previous detecting algorithms to compare GN [17] and LPA [19]. Particularly, the Net-science dataset has a runtime exceeding acceptable threshold so we set it aside. Our experiment results are described in Table 2, and best value for each test formula is bolded.

TABLE 2. The results comparison of GN, LPA and CDANI

Dataset	GN				LPA				CDANI			
	Q	NMI	F1-score	#Com	Q	NMI	F1-score	#Com	Q	NMI	F1-score	#Com
Karate [20]	0.401	**0.579**	0.042	5	0.345	0.657	**0.692**	1	**0.426**	**0.679**	0.511	4
Dolphins [15]	0.495	**0.554**	0.204	8	0.498	0.595	0.312	10	**0.504**	0.621	**0.469**	5
Football [7]	**0.595**	**0.879**	0.204	8	0.583	0.872	0.312	10	0.562	0.809	**0.389**	9
Political Books	0.517	**0.558**	0.074	5	0.481	0.423	0.087	4	**0.528**	0.394	**0.294**	10
Net-science [16]	#n/a	#n/a	#n/a	480	**0.891**	#n/a	#n/a	455	0.714	#n/a	#n/a	708

Based on Table 2, most computed Modularity values are approximate or higher than other comparising methods in spite of the different quantity and structure of detected communities. Results show that each community detected by CDANI has high quality. As we can see in Figure 4, Modularity value of CDANI is even better compared to the datasets's ground truth, only Modularity value of Football Network is lower than ground truth. The number of conmmunities between ground truth and CDANI's networks has been increasing the differences according to the network size. Our results in NMI and F1-score values are sometimes lower than other comparing methods.

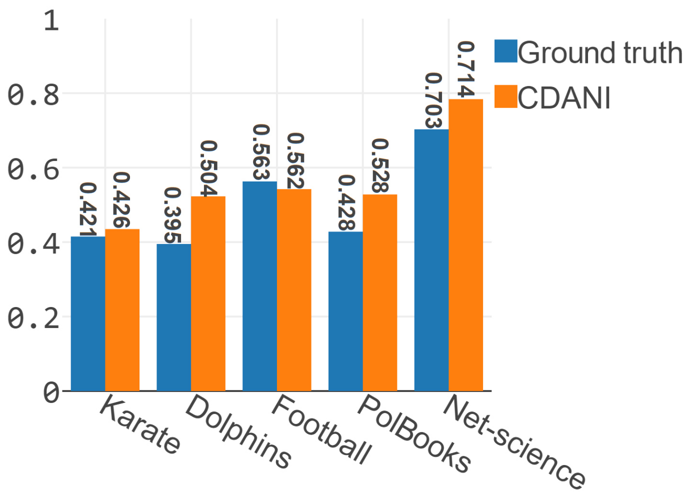

FIGURE 4. Q of CDANI and ground truth

4. **Conclusion.** In this paper, we studied the problem of community discovery in the subject of complex network analysis. We proposed an improved algorithm based on node intimacy. At first, we calculated the node's individual importance by inherited formula and arranged them in descending order of priority. Next, the nodes are conducted merging based on the most important nodes to their neighbors. Our algorithm is also capable of finding overlapping communities and improving stability. Finally, we compare the performance of the algorithm proposed in the paper with the previous represented methods with real-world networks, and give the expected test results. The quality of the discovered community structure is better than the compared algorithms.

In the future, we research on parallelizing our algorithm to amend the results when running large-scale datasets.

REFERENCES

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, *Journal of Statistical Mechanics Theory and Experiment*, 2008.

[2] A. Clauset, M. E. J. Newman and C. Moore, Finding community structure in very large networks, *Physical Review E*, vol.70, no.6, 2005.

[3] L. Danon, A. Díaz-Guilera, J. Duch and A. Arenas, Comparing community structure identification, *Journal of Statistical Mechanics Theory and Experiment*, DOI: 10.1088/1742-5468/2005/09/P09008, 2005.

[4] J. Ding, Y. Sun, P. Tan and Y. Ning, Modularity maximization for community detection in networks using competitive hopfield neural network, *International Journal of Innovative Computing, Information and Control*, vol.15, no.4, pp.1455-1467, 2019.

[5] J. Eustace, X. Wang and Y. Cui, Community detection using local neighborhood in complex networks, *Physica A*, vol.436, pp.665-677, 2015.

[6] S. Fortunato, Community detection in graphs, *Physics Reports*, vol.486, nos.3-5, pp.75-174, 2009.

[7] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, *Proc. of the National Academy of Sciences*, pp.57-66, 2002.

[8] S. Gregory, An algorithm to find overlapping community structure in networks, *Proc. of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp.91-102, 2007.

[9] S. Gregory, Finding overlapping communities in networks by label propagation, *New J. Phys*, vol.12, no.10, 2010.

[10] P. Jaccard, The distribution of the flora in the alpine zone, *New Phytologist*, vol.11, no.2, 1912.

[11] G. Agarwal and D. Kempe, Modularity-maximizing graph communities via mathematical programming, *Physica A: Statistical Mechanics and Its Applications*, vol.66, no.3, 2008.

[12] W. Li, Q. Kang, H. Kong, C. Liu and Y. Kang, A novel iterated greedy algorithm for detecting communities in complex network, *Social Network Analysis and Mining*, vol.10, no.29, 2019.

[13] X. Li, M. K. Ng and Y. Ye, Multicomm: Finding community structure in multi-dimensional networks, *IEEE Trans. Knowledge and Data Engineering*, vol.26, no.4, pp.929-941, 2013.

[14] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski and D. Wagner, Maximizing modularity is hard, *arXiv.org*, arXiv: physics/0608255, 2006.

[15] 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 – Can geographic isolation explain this unique trait?, *Behavioral Ecology and Sociobiology*, vol.54, pp.396-405, 2003.

[16] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, *Physical Review E*, vol.74, no.3, 2006.

[17] M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks, *Physical Review E*, vol.69, no.2, 2004.

[18] K. S. Phyu and M. M. Min, Graph-based community detection in social network, *The 18th International Conference on Computer and Information Science (ICIS)*, 2019.

[19] U. N. Raghavan, R. Albert and S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, *Physical Review E*, vol.76, no.3, 2007.

[20] W. W. Zachary, An information flow model for conflict and fission in small groups, *Journal of Anthropological Research*, vol.33, no.4, pp.452-473, 1976.

[21] T. Zhou, L. Lü and Y.-C. Zhang, Predicting missing links via local information, *The European Physical Journal B*, vol.71, no.4, pp.623-630, 2009.