Journal of Computer Science and Cybernetics, V.38, N.1 (2022), 31–46
DOI no. 10.15625/1813-9663/38/1/16537

A FAST OVERLAPPING COMMUNITY DETECTION ALGORITHM BASED ON LABEL PROPAGATION AND SOCIAL NETWORK GRAPH CLUSTERING COEFFICIENT

NGUYEN HIEN TRINH[1], DOAN VAN BAN[2], VU VINH QUANG[1], CAP THANH TUNG[3,*]

[1] *Thai Nguyen University of Information and Communication Technology*
[2] *Institute of Information Technology - Viet Nam Academy of Science and Technology*
[3] *Thai Nguyen University of Education*

Abstract. Many networks possess a community structure, such that nodes form densely connected groups and are more sparsely linked to other groups. In many cases, these groups overlap, with some nodes shared between two or more communities. Overlapping node plays a role of interface between communities and it is really interesting to study the community establishment of these nodes because it reflects the dynamic behavior of participants. Nowadays, community identification and mining are the main directions in social network analysis.

In this paper, we present an algorithm to find overlapping communities in very large social networks. The algorithm is based on the label propagation technique, and we find the overlapping communities in the network by improving the clustering coefficient. Tests on a set of popular, standard social networks and certain real networks have shown the high speed and high efficiency in finding overlapping communities.

Keywords. Social network graph; Overlapping communities; Label propagation; Clustering coefficient; Belonging coefficient.

1. INTRODUCTION

Community structure is an important field in complex networks research. Newman et al. [5] had made clear that a community structure could be defined as a group of nodes with dense internal links and sparse connections between groups. The finding of the community structure of a social network is very important. The metabolic networks or the large-scale WWW webpage links are all community structures. Regarding community structure detecting, we must solve two complicated problems as below: Firstly, we have to deal with the running time of algorithms. Although community detection on social networks is a research direction that many scientists are interested in, many algorithms have been proposed, but one of the problems that need to be overcome is the processing speed of algorithms. In fact, the number of vertices and edges of the graph is too large, leading to a long-running time

*Corresponding author.
E-mail addresses: nhtrinh@ictu.edu.vn (N.H.Trinh), dvban@ioit.ac.vn (D.V.Ban), vvquang@ictu.edu.vn (V.V.Quang), tungct@tnue.edu.vn (C.T.Tung)

which is unsuitable for practical requirements. Since the first work on community structure by Weiss and Jacobsen (1955) dealing with separating working groups in governmental organizations, many algorithms have been studied and developed including works of Flake, Radicchi, and especially Girvan and Newman with famous GN algorithm calculating betweenness of edges and then trimmed out edges of highest betweenness. The complexity of GN is $O(m^2.n)$, (m: the number of edges, n: the number of vertexes). Although many other algorithms have been proposed later such as CONGA with $O(m^3)$; CONGO with $O(n.\log n)$; Brandes with $O(n.m)$ (weighted graph) or $O(n.m+n^2\log n)$ (unweighted graph). In general, those algorithm has high complexity. Nowadays, while social network graphs have become increasingly complex and extremely large, the common tendency is to find out the solution of acceptable accuracy in the permitted time. So the Label Propagation Algorithms (LPA) have been strongly developed. LPA is a popular method used for finding communities in an almost-linear time-consuming process. However, its performance is not satisfactory in some metrics such as accuracy and stability. The complexity of LPA is $O(m+n)$, for sparse graph graph is $O(n)$. The common feature of LPAs is to find locally optimal results, but they differ in designing optimal functions based on the interests of each author. We followed this orientation and designed the optimal function for our algorithm.

Secondly: In detecting community structure, the authors always supposed that there is a very clear division for each community: each node belongs to only one community [4, 5, 13]. In reality, networks are built in different relations, and nodes can be shared by many communities [3]. For example, in collaboration networks in scientific research, an author can participate in numerous groups with other scientists, or in bio-logical networks, a protein structure can deal with many other groups of proteins, etc... An individual can belong to many communities in simultaneous interactions with multiple groups. This feature makes overlapping to be an important characteristic of complicated networks, especially social networks. Obviously, the algorithm for the overlapping community is hence very complicated. Recently some algorithms have been proposed to detect the overlapping community structure, and two main effective means are clique and optimization theory. The methods based on clique have high accuracy, but the process is complex. While the optimization algorithms choose the appropriate object function and get a lower complexity, when and how to finish are ambiguous. As we paid attention to fast processing big data sets so we made our approach based on the second orientation. Many algorithms have used the clustering coefficient as an analysis parameter (COPRA, IVICCOPRA...). We analyzed and introduced the new parameter: the belonging coefficient, which is developed from the clustering coefficient by improving and removing irrelevant or approved nodes to reduce the number of re-updates during label propagation. In this paper, we propose a fast overlapping community detection algorithm with a belonging coefficient and implement the label propagation method (COPA-BC) to quickly detect overlapping structures.

The rest of the paper is organized as follows. We first briefly consider related work in Section 2. Section 3 presents the graph clustering coefficient and belonging coefficient. Section 4 introduces the fast overlapping community detection algorithm. Section 5 presents the results of the experiments. Conclusions appear in Section 6.

2. RELATED WORKS

The social network is often represented by an undirected, connected graph $G = (V, E)$, also known as a social network graph, where V is the set of nodes and E is the set of edges. The node v is adjacent to w if $(v, w) \in E$ or $(w, v) \in E$. Suppose node v has k adjacent nodes, denote $N(v) = \{v_1, v_2, \ldots, v_k\}$ representing k neighbors of v. The label of v_j:$L(v_j)$, denotes the community to which v_j belongs. Nowadays, many algorithms have been proposed for community detection in social networks. The most famous algorithm is Girvan-Newman's division algorithm [5] which proposes to detect community clusters on a social network graph using the betweenness centrality of the edge to remove the intermediate highest edge in each iteration. This process will continue until it reaches communities with high modularity. In other words, modularity features are used to evaluate the quality of detected communities. Gregory proposes the Cluster-Overlap Newman Girvan Algorithm (CONGA)[6], to detect overlapping communities using the concept of a division of community structure through local mediation. The algorithm is efficient at discovering communities of small diameter in large networks and $O(n. \log n)$ time complexity for sparse networks. The next typical and popular algorithm in the social network community discovery field is the Label Propagation Algorithm (LPA), introduced by Raghavan and Girvan [13]. LPA is an algorithm that detects communities on a social network in almost linear time. This algorithm has drawn the attention of many researchers to study, improve, develop and apply to many different cases [7, 14]. Most overlapping community detection methods cannot simultaneously satisfy the efficiency and accuracy requirements for large and dense networks. Steve Gregory proposes the Community Overlap Propagation Algorithm (COPRA) [7] to detect overlapping communities by extending the label concept and label propagation step to include information about more than one community: each existing node can belong to the maximum v community, where v is the parameter of the algorithm. In the COPRA algorithm, the label for each node x is a set of pairs (C, b), where C is the identifier (label) of the community and b is the coefficient of the community based on the degree of nodes, indicating the likelihood of membership of x in the C community, such that all the membership coefficients of having x sum equal to 1 (normalized). Each propagation step will recalculate the label of x across the labels of neighboring nodes, sum the community coefficients of all neighboring nodes, and normalize. This can cause poor performance of the algorithm, in part because the nodes with high degree values are sometimes unable to belong to more than one community, for example, the nodes of the clique graph (full graph). Recently, Saradha and Arul focused on the optimized overlapping community detection technique by applying the Improved Vertex Imitation Co-efficient based Community Overlap Propagation Algorithm (IVICCOPRA) [10], but has not yet addressed all the disadvantages of COPRA [7]. In particular, we do not have the information to specify the v parameter about the number of communities to which the nodes belong. To overcome the above limitations, we focus on studying the graph clustering coefficient [12] and its application to develop a fast overlapping community detection algorithm based on label propagation and advanced clustering coefficient (belonging coefficient).

3. GRAPH CLUSTERING COEFFICIENT AND BELONGING COEFFICIENT

3.1. Graph clustering coefficient

The clustering coefficient is used to determine to "quantify the structural properties" of the network [12]. The largest advantage of this value is the fact that it shows all the networks in which it contains clusters. The clustering coefficient of vertex v is determined by the number of unordered adjacent pairs of v that are directly connected, divided by the total number of adjacent pairs of v.

In graph theory, the clustering coefficient is a measure of the degree to which the nodes in the graph tend to gather together. Evidence shows that, in most real-world networks, especially social networks, nodes tend to form interconnected groups characterized by a relatively high density of relationships known as is the community. This probability tends to be greater than the average probability of the constraint being randomly established between two nodes.

Given an undirected, connected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes and E is a set of m edges; The two nodes $u, v \in V$ are linked (connect) together if $(u, v) \in E$. The triple of nodes v_i is a set $(v_i, \{v_j, v_k\})$ so $(v_i, v_j), (v_i, v_k) \in E$. Triangle associated with v_i is an unordered triple of nodes such that each pair of vertices is connected. The clustering coefficient of node v_i is defined as follows

$$C_i = \frac{\text{Number of triangles of node } v_i}{\text{Triple of node } v_i}. \tag{1}$$

These measurements are 1 if every neighboring node connected to v_i is also connected to every other vertex in the neighborhood and is 0 if among the adjacent nodes no pairs are connected together or degrees $\deg(v_i) = 1$. If $\deg(v_i) = d_i$, then the number of triplets associated with v_i would be $d_i * (d_i - 1)/2$. The clustering coefficient can be defined equivalent as follows

$$C_i = \begin{cases} \dfrac{\text{Number of triangles of node } v_i}{d_i(d_i - 1)/2} & \text{if } \ d_i > 1, \\ 0 & \text{if } \ d_i \leq 1. \end{cases} \tag{2}$$

The formulas (1) and (2) are equivalent. Given the adjacent matrix of the graph $G = (V, E)$, $\mathrm{A} = [a_{ij}]$, $i, j = 1, 2, \ldots, n$, $a_{ij} = 1$ if $(v_i, v_j) \in E$, otherwise $a_{ij} = 0$. Then the clustering coefficient can be defined equivalently as follows

$$C_i = \begin{cases} \frac{\sum_{j,k=1}^{n} a_{ij} * a_{jk} * a_{ki}}{d_i(d_i - 1)/2} & \text{if } \ d_i > 1, \\ 0 & \text{if } \ d_i \leq 1, \end{cases} \tag{3}$$

where $d_i = \sum_{j=1}^{n} a_{ij}$.

Example 1. Consider the graph shown in Figure 1

Calculated results of every formula (1), (2), (3) all give the clustering coefficients as follows: $C_1 = 0,\ C_2 = 1/6, C_3 = 0,\ C_4 = 2/3,\ C_5 = 5/14,\ C_6 = 1,\ C_7 = 1,\ C_8 = 1, C_9 = 1,\ C_{10} = 1,\ C_{11} = 1,\ C_{12} = 0,\ C_{13} = 0,\ C_{14} = 0.$

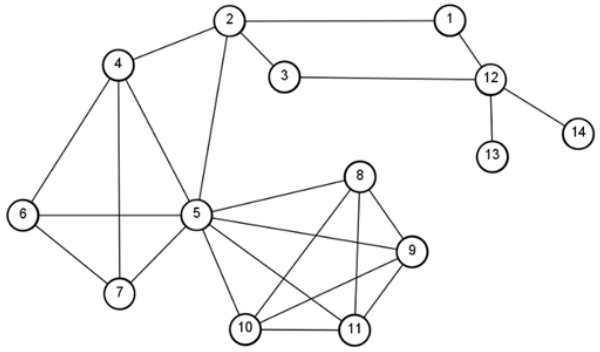

Figure 1: Network G (graph with 14 member (nodes))

For very large-scale networks, calculating the exact clustering coefficient may not be feasible because it takes a lot of computation time. It is acceptable to reduce accuracy to get results more quickly. There are many algorithms to calculate the clustering coefficient, typically the ACCA algorithm (Approximating Clustering Coefficient Algorithms) [11] to determine the clustering coefficient with complexity $O(1)$. The ACCA approximation algorithm that calculates the clustering coefficient is as follows:

Input : Adjacent matrix A of network $G = (V, E); x \in V; k$
Output : C_x

```
l = 0;
for i ∈ (1, ..., k) do {
        j = GetRandom(k);
        u = GetRandomVertex(N(j));
        v = GetRandomVertex(N(j));
        while (u==v)
            {
              v = GetRandomVertex(N(j));
              if (checkEdge(u,v))
              l = l+1;
             }
      }
 return Cx= l/k;
```

The run-in-time approximation algorithm is constant, given the fact that all methods inside the loop can be evaluated in constant time. While the loop is limited by a number k, the algorithm has $O(k)$ complexity, but since k can be chosen by the user (k is constant), so the algorithm has an exact complexity $O(k)= O(1)$.

3.2. Belonging coefficient

On the social network, a community can be defined as a set of nodes that have a high density between them and a lower density with the rest. Thus, whether a node on the

graph belongs to a community depends on the number of nodes associated (its degree) and the number of connections between it and neighbors. For example, on the graph in Figure 1, nodes: 4, 5, 6, 7 have enough associations to see if they are in a community. Thus, it is reasonable to apply the clustering coefficient to more effectively cluster overlapping communities on the social network. However, by the definition (1), nodes with degree 1 ($\deg(v) = 1$) have a clustering coefficient = 0, so they can only reside in one community. In addition, nodes whose neighborhood set is a clique that has a clustering coefficient of 1 also reside in a community. Therefore, it makes no sense: nodes located on only one community may have different clustering coefficients (is 0 or 1). From those practical analyses, we improve the clustering coefficient to the belonging coefficient of node v_i as follows

$$B_i = \begin{cases} C_i & \text{if } C_i > 0, \\ 1 & \text{if } C_i = 0. \end{cases} \tag{4}$$

Example 2. Considering the graph of Figure 1, according to (4), we get the belonging coefficients for the nodes: $B_1 = 1,\ B_2 = 1/6,\ B_3 = 1,\ B_4 = 2/3,\ B_5 = 5/14,\ B_6 = 1,\ B_7 = 1,\ B_8 = 1,\ B_9 = 1,\ B_{10} = 1,\ B_{11} = 1,\ B_{12} = 1,\ B_{13} = 1,\ B_{14} = 1.$

The belonging coefficient of node i determines the probability that node i belongs to one or more communities. From Example 2 and through the experimental results of the algorithm to calculate the belonging coefficient of the nodes, along with the statistical results on typical social networks such as Karate Club, Dolphin, Protein, Net-Science network [1, 8], we find that when $B_i = 1$, node i clearly belongs only to one community, such as hanging nodes (whose vertex degree is 1), or the central node of a star graph with zero membership triangles, or the nodes of a clique are all located in a community. Nodes with $B_i < 1$ are nodes that are likely to overlap in different communities. Continuing with Gregory's idea [7], we add the constraint by specifying the parameter v, which represents the maximum number of communities that each node can belong to. It is easy to infer that the nodes with low belonging coefficients are the ones most likely to be in overlapping regions of v communities. That is, the overlapping nodes i can only have belonging coefficients less than $1/v$. From there we deduce:

i. Nodes with the belonging coefficient $> 1/v$ belong to a community. (5)

ii. If nodes i have the belonging coefficient $0 < B_i \leq 1/v$, then i can belong to at most $[1/B_i]$ communities. Where $[r]$ is the integer part of the real number r. (6)

From (5) we deduce that the node i has $B_i > 1/v$, it is possible to assign the coefficient of the community to which it belongs to 1. From (6) we can determine the input parameter v for the proposed algorithm

$$1 \leq v \leq \left[\arg\max_{i \in V} (1/B_i) \right]. \tag{7}$$

Considering the graph of Figure 1, if you choose $v = 2$, then nodes 2 and 5 are two nodes belonging to two overlapping communities.

4. DETECT OVERLAPPING COMMUNITY BASED ON LABEL PROPAGATION AND BELONGING COEFFICIENTS

4.1. The method of label propagation is based on belonging coefficients to detect overlapping communities

Label Propagation Algorithm (LPA) is an algorithm that detects communities on a social network in almost linear time. The LPA algorithm follows these steps: The first step is to assign each node a unique label. The label of a node represents the community to which this node belongs. Next, the node is selected in random order, and its label is updated again according to the parameters of its neighbors. The new label of node x in the t iteration is updated according to the labels of the neighbors of node x in the $(t-1)$ iteration. The algorithm terminates when every node has a label that is one of the labels used by the nodes with the maximum number of neighbors. Based on the belonging coefficient and combined with the method of the label propagation of Gregory [7], we can detect overlapping communities. First, we choose the value v (maximum number of communities that a node can belong to), as a parameter of the algorithm, determined according to the condition (7). At the same time, we start to label each vertex x as a set of pairs $L_1(x) = \{(x, B_x)\}$, if $B_x \leq 1/v$, or vice versa, $L_1(x) = (x, 1)$, where B_x is the belonging coefficient of node x, calculated according to (4) and based on the conclusions (5), (6). During label propagation, the node's label is either added to a new label pair or changed to another label pair. The label $L_t(x)$ is the label used to assign node x at time t. Each propagation step will update the label of node x according to the label of the neighboring node whose community membership coefficient is maximum, often selecting adjacent nodes labeled with a belonging coefficient of 1. During the different stages of label propagation, the role of the key nodes towards community establishment will greatly influence the formation of communities in many different ways. For example, in the early stage, the key nodes of the community (nodes with a high belonging coefficient $\geq 1/v$) are nodes that make a significant contribution to determining community structure by propagating instantaneous transmission of the neighbor nodes in the same community. While potentially overlapping nodes (nodes with low belonging coefficients $\leq 1/v$) are located between many different communities, making it difficult to determine the community structure because they are located in the overlap between communities. The belonging coefficient of nodes can help to randomly adjust the label propagation steps and better determine the update order in the sequence of neighboring nodes. Due to the topology characteristics of the nodes with different roles, the propagation by the node with the first high belonging coefficient (key node) is usually quite fast, which is also the reason for the case that the overlap will be propagated very quickly according to the key node. In contrast, propagation along nodes with low belonging coefficient often leads to communities that are too large, which is unusual in reality. When the key nodes are updated, potentially overlapping nodes will soon be propagated[14]. Nodes x with a belonging coefficient greater than $1/v$ are all assigned 1, using only one label to identify the community to which it belongs, i.e. $[L_t(x)] = 1$, for nodes x with $B_x \leq 1/v$, the number of labels used for assignment must be less than or equal to v, ie $|L_t(x)| \leq v$, where t is the time t of label propagation. In each step of label propagation,

the label of node x will be updated with priority according to the nodes whose labels have the highest belonging coefficient of 1. After each iteration of label propagation, the number of labels used to assign to nodes is likely to decrease. Thus, the label sequence $L_t(x)$ of node $x \in V$ in iteration t is updated according to the labels of neighboring nodes in $N(x)$ in the previous time as follows:

a) The label of x with belonging coefficient of 1 must be updated according to the label of a neighbor node with a belonging coefficient of 1, that is, $L_t(x) = L_{t-1}(s)$ with $s \in N(x)$ and $L_{t-1}(s) = \{(w, 1)\}$, so the number of used labels will be reduced by 1, label x will be deleted.

b) It is necessary to update the label of x with community coefficient $\leq 1/v$ according to the label of $s \in N(x)$ if any $(w, 1) \in L_{t-1}(s)$, $(w, 1) \notin L_{t-1}(x)$ then $L_t(x) = L_{t-1}(x) \cup \{(w, 1)\}$, that is to add labels to overlapping nodes.

The algorithm stops when all nodes have the same used label sets between two consecutive iterations, i.e. no nodes are re-labeled. It is easy to see that this will happen after several steps and thus the algorithm is guaranteed to terminate. After the algorithm is finished, the nodes with the same label will be in the same community, and the nodes with more than one label are the overlapping nodes.

Given the graph in Figure 1, at steps $t = 1, 2, 3, 4$ with parameter $v = 2$, there are labels assigned as shown in the following table.

Table 1: Labeling process at steps $t = 1, 2, 3, 4$

L_t\x	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L_1(x)	(1,1)	(2,1/6)	(3,1)	(4,1)	(5,5/14)	(6,1)	(7,1)	(8,1)	(9,1)	(10,1)	(11,1)	(12,1)	(13,1)	(14,1)
L_2(x)	(12,1)	{(2,1/6); (12,1)}	(12,1)	(6,1)	{(5,5/14); (6,1)}	(6,1)	(6,1)	(9,1)	(9,1)	(9,1)	(9,1)	(12,1)	(12,1)	(12,1)
L_3(x)	(12,1)	{(2,1/6); (12,1); (6,1)}	(12,1)	(6,1)	{(5,5/14); (6,1); (9,1)}	(6,1)	(6,1)	(9,1)	(9,1)	(9,1)	(9,1)	(12,1)	(12,1)	(12,1)
L_4(x)	(12,1)	{(2,1/6); (12,1); (6,1)}	(12,1)	(6,1)	{(5, 5/14); (6,1); (9,1)}	(6,1)	(6,1)	(9,1)	(9,1)	(9,1)	(9,1)	(12,1)	(12,1)	(12,1)

The number of labels assigned in the initial step is the same as the number of nodes, that is, 14 labels. The number of labels used to assign labels in step $t = 2$ is 3, in step $t = 3$ is 3. The algorithm stops after iteration $t = 4$ because the number of labels used for labeling has not changed and there are no more nodes to update the labels again.

The number of labels used to assign labels in the last step is 3. The graph is divided into 3 overlapping communities: community 1, 2, 3, 12, 13, 14 labeled 12, community 2, 4, 5, 6, 7 labeled 6 and community 5 , 8, 9, 10, 11 are labeled 9. Node 2 is in the first two communities and node 5 is in the latter two.

4.2. Algorithm COPA-BC

In the algorithm COPA-BC (Community Overlap Propagation Algorithm Based on New Belonging Coefficient), a sequence of pairs of labels L_t is assigned to the nodes at the t iteration, and $N(x)$ is the set of neighbors of the node $x \in V$. The value of B_x determines the assignment of a new label during propagation.

Input: Graph $G = (V, E)$ and parameter v satisfy condition (7)
Output: Overlapping Communities.

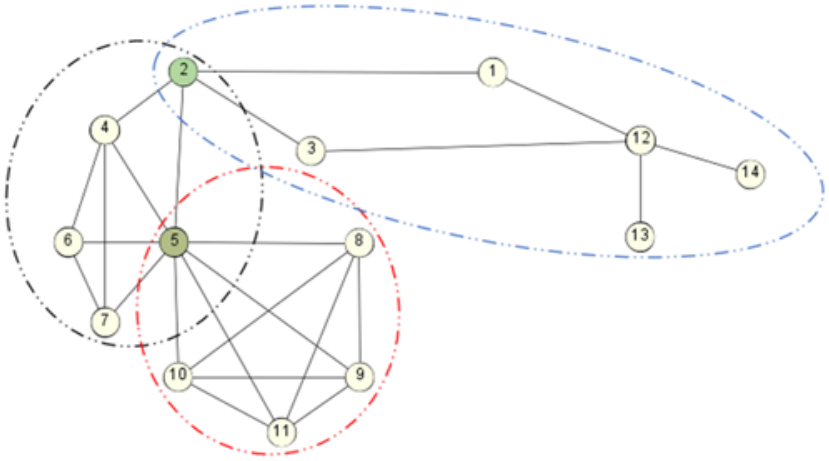

Figure 2: Network graph G has 3 overlapping communities with two overlapping nodes 2, 5

COPA-BC(G, v) {

CO $= \{\}$;

$t = 1$;

for each $x \in V$ do{

$B_x = ACCA(x, k)$; //Calculate the coefficient according to the ACCA

if $\left(B_x > \frac{1}{v}\right)$ then $L_t(x) = \{(x, 1)\}$;

else $L_t(x) = \{(x, B_x)\}$;

};

$w = \arg\max_{x \in V} \left([\frac{1}{B_x}]\right)$; // [h] take the integer value of h

if$(v > w)$ then $v = w$; // Check condition (7);

$t = t + 1$;

do {

for each $x \in V$ do {

$N(x) = \{y \in V | (x, y) \in E\}$;

Propagate$(N(x), L_{t-1}, L_t)$;

} while$(L_{t-1} \neq L_t)$;

SplitCommunities(L_t) ; // To split into overlapping communities.

return CO;

}

Propagate$(N(x), L_{t-1}, L_t)$ {

for each $y \in N(x)$ do {

if ($L_{t-1}(y) == \{(s,1)\}\&\&B_x == 1$) then {

$L_t(x) = L_{t-1}(s)$;

return; // stop : Ending Propagate().

}

if ($B_x \leq 1/v\&\&(w,1) \in L_{t-1}(y)\&\&(w,1) \notin L_{t-1}(x)\&\&|Lt| < v$) then {

$L_t(x) = L_{t-1}(x) \cup \{(w,1)\}$;

return; // stop : Ending Propagate()

}

}

SplitCommunities(L_t) {

for each $x \in V$ do if ($L_t(x) == (x, B_x)\&\&B_x \leq \frac{1}{v}$) then $delete(L_t(x))$;

//After label propagation, remove labels with belonging coefficient $\leq \frac{1}{v}$

for each $x \in V$ do {

$CO_x = \{\}$; // *Empty set*

if ($L_t(x) == \{(x, Bx)\}\&\&Bx == 1$) then

$\{CO_x = CO_x \cup \{x\}$;

$CO = CO \cup CO_x; \}$

//To determine communities with nodes of the same label (having a belonging coefficient of 1)

for each $y \in V - \{x\}$ do

if ($L(y) == L(x)\&\&(y, B_x) == (x, B_x)$ then

$\{CO_x = CO_x \cup \{y\};$

$V = V - \{x\};$

$CO = CO \cup CO_x; \}$

else

$\{CO_v = CO_v \cup \{y\};$

$V = V - \{y\};$

$CO = CO \cup CO_v;$

}

4.3. Evaluate the complexity of the algorithm

The time complexity of the algorithm is estimated below. n is the number of nodes and v is the parameter (maximum number of communities per node).

- Statement for each $x \in V$ determines the belonging coefficient and assigns it to the nodes of V. The time complexity of this statement is $O(n.O(1))$, where $O(1)$ is the complexity of the ACCA algorithm [11].
- Next statement calculates w as the maximum value of $1/Bv$, with $v \in V$, so the number of executions is $O(n)$.
- In the Propagate procedure, for each node x, the label is updated according to its neighbors. Nodes with a belonging coefficient equal to 1 have only one pair of labels assigned, while nodes with community dependency coefficient less than or equal to $1/v$ have at most $v + 1$ pairs of labels, used to assign (update) labels. For each adjacent node y, it iterates over all (at most v) the number of community labels in the used label sequence, so that the update time for x's labels takes at most v number of checks. Then, the time complexity of Propagate algorithm is $O(v)$.
- Statement do... while propagating the label Propagate(), repeats until two consecutive iterations with no node updating the label. Therefore, the time complexity of this statement is $O(n.v)$, where n is the number of nodes of the graph and v is the parameter (maximum number of communities per node).
- The procedure SplitCommunities() divides nodes with the same label into overlapping communities, also with a complexity of $O(n)$.
- The remaining statements are single statements, only executed once.

Therefore, the time complexity of the algorithm COPA-BC is $O(n.O(1)) + O(n) + O(v) + O(n.v) + O(n) = O(n.v)$. Since, v is constant number, then the time complexity of the algorithm COPA-BC is nearly linear.

5. EXPERIMENTAL EVALUATION

We have experimental results on published standard data sets [1, 8] to evaluate the effectiveness of the proposed algorithm COPA-BC for fast detection of communities, compared with recent popular algorithms, such as COPRA [7], IVICCOPRA [10]. We also compare the accuracy of the algorithm through the quality of the community (modularity measure) and the normalized mutual information (NMI) of the community detected by the proposed algorithm.

5.1. Evaluate the effectiveness of the algorithm

Experimental results have confirmed that the proposed algorithm COPA-BC to detect overlapping communities runs faster than the recently popular algorithm COPRA [7] on av-

Table 2: Experimental results on execution time (n-number of vertices, m-number of edges, cc-number of announced communities, cn-number of result communities, dcl number of overlapping vertices). The execution time is in s seconds.

N0	**Social Network**	Nodes (n)	Edges (m)	cn	dcl	Runtime(s)		
						COPA-BC	COPRA	IVICCOPRA
1	Karate Club	34	78	3	3	4.75	5.15	5.94
2	Dolphin Group	62	159	2	3	6.02	6.96	6.56
3	Email-Eu-core	1005	25571	29	13	65.02	69.72	68.12
4	DBLP	317080	1049866	12556	127	554.23	649.18	633.21
5	Amazon	334863	925872	35123	253	579.09	659.22	655.11
6	Youtube	1134890	2987624	8135	189	1109.15	1254.35	1158.23

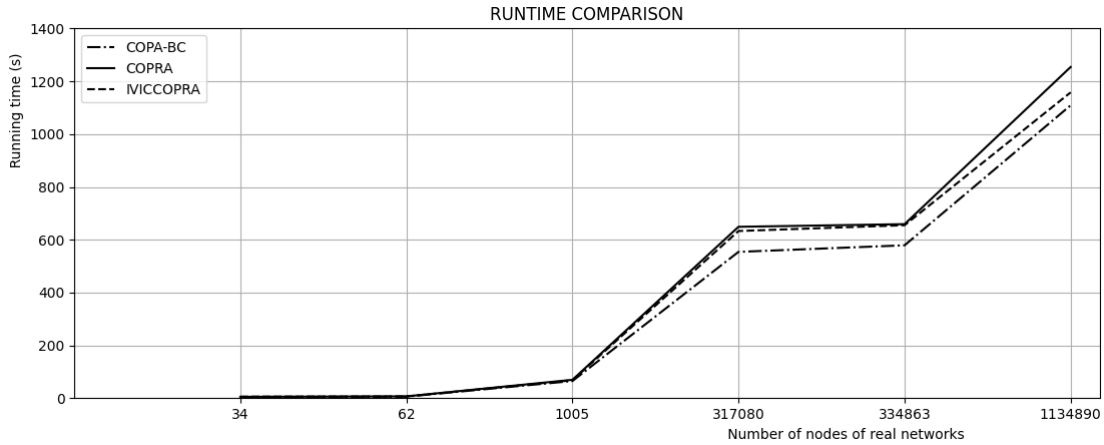

Figure 3: Runtime comparison of COPA-BC algorithm on 6 real networks [1, 8]

erage by 15%, IVICCOPRA [10] on average by 10%, for all experimental data sets, especially for networks with a large number of vertices.

5.2. Evaluate the modularity quality of the algorithm

Modularity Q [2] is proposed to measure the community division status of the entire network. Networks with high modularity Q show that there are much more links between vertices in the community and fewer links between vertices in different communities. Therefore, the larger the value of the modularity Q is the higher the accuracy of the algorithm shall be, leading to the quality of community detection being assessed as good

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta\left(C_i, C_j\right), \tag{8}$$

where A_{ij} is the adjacency matrix, m is the total number of edges of the graph, d_i is the degree of vertex i, d_j is the degree of vertex j, C_i, C_j are the communities where vertex i and vertex j are, respectively in that community. $\delta\left(C_i, C_j\right) = 1$ if vertex i and vertex j belong to the same community, otherwise $\delta\left(C_i, C_j\right) = 0$.

Table 3: Experimental results on community quality Modularity

No	**Social Network**	Modularity		
		COPA-BC	COPRA	IVICCOPRA
1	Karate Club	0.69	0.58	0.57
2	Dolphin Group	0.63	0.55	0.53
3	Email-Eu-core	0.79	0.77	0.75
4	DBLP	0.81	0.75	0.77
5	Amazon	0.87	0.81	0.83
6	Youtube	0.79	0.75	0.76

Through the above experimental data with the modularity Q in these datasets, it has been shown that the proposed method COPA-BC has a better community quality (about 10% on average) than the recently popular algorithms such as COPRA [7], IVICCOPRA [10] on all experimental datasets and solved the overlapping community problem.

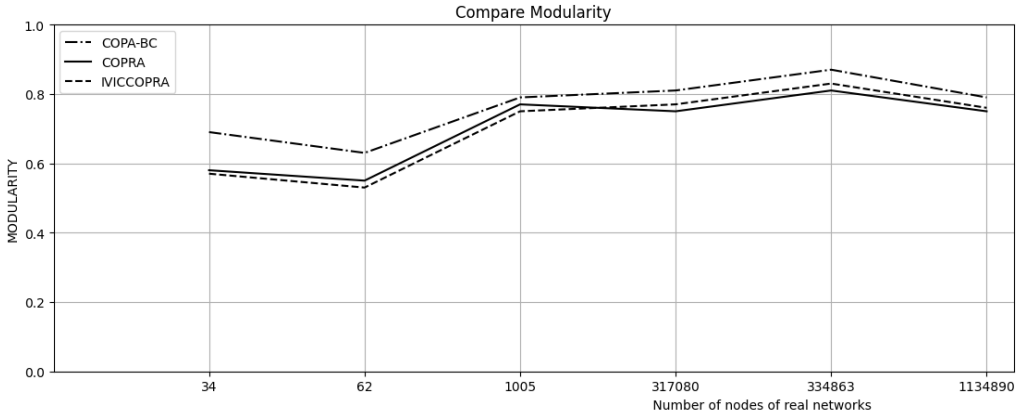

Figure 4: Comparison of detected community quality across 6 real networks

5.3. Evaluate the NMI of the algorithm

Metrics based on information theory present an alternative approach to community quality testing with a given reference partition. The most commonly used metric is the Normal Mutual Information (NMI) [9], which can be calculated and determined according to the arithmetic mean of the entropies

$$NMI(A,B) = \frac{2I(A,B)}{H(A)+H(B)},$$

where, $H(A) = -\sum_{i=1}^{|A|} p_i \log p_i$, and

$$H(B) = -\sum_{j=1}^{|B|} p_j \log p_j, \;\; p_i = \frac{|A_i|}{N}, \;\; p_j = \frac{|B_j|}{N}, \;\; p_{ij} = \frac{|A_i \cap B_j|}{N}.$$

Or

$$NMI(A,B) = \frac{-2\sum_{i,j} p_{ij} \log \frac{p_{ij}}{p_i p_j}}{\sum_i p_i \log p_i + \sum_j p_j \log p_j}. \tag{9}$$

It is common sense that N is the number of vertices of the network graph under consideration; A : set of real communities; B : the set of communities detected by the community detection algorithm in use. The value of NMI ranges from 0 to 1. The NMI is equal to 1 if the detected community matches the real community. In contrast, NMI is 0. With different community detection methods, when calculating the corresponding NMI, the closer the value of NMI is to 1, the better.

Table 4: Experimental results on NMI

No	**Social Network**	NMI		
		COPA-BC	COPRA	IVICCOPRA
1	Karate Club	0.89	0.79	0.77
2	Dolphin Group	0.91	0.83	0.84
3	Email-Eu-core	0.74	0.67	0.69
4	DBLP	0.68	0.65	0.67
5	Amazon	0.65	0.61	0.63
6	Youtube	0.59	0.55	0.54

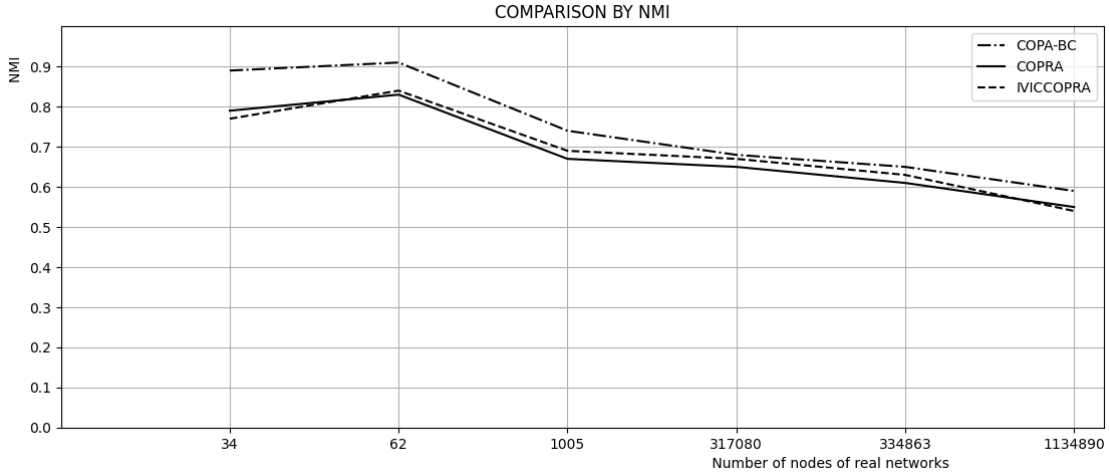

Figure 5: Comparison by NMI of communities detected on 6 real networks

Through the data of NMI in these data sets, it is shown that the proposed method COPA-BC has an NMI value close to 1 and greater than that of the popular algorithm recently (average about 10%) as COPRA and IVICCOPRA across all experimental data sets. It proves that the community detected by the proposed algorithm is close to the announced community and also solves the overlapping community problem.

We can see that the COPA-BC algorithm is better than other algorithms (COPRA, IVICCOPRA) because the repeating times have been reduced thanks to parameter v as analyzed in conclusions (5) and (6). The community quality has been improved due to the exploration and selection of communities according to Q (Modularity) in each repeated step. Experimental results have proved it.

6. CONCLUSION

The article introduced the COPA-BC algorithm to detect overlapping communities in social networks based on the label propagation method and belonging coefficient. The algorithm is developed based on the label propagation method and uses the belonging coefficient advanced from the clustering coefficient to quickly and effectively find overlapping communities. Compared to other overlapping community detection algorithms, the advantage of COPA-BC is execution speed and accuracy. The execution time is linear with the node number of the social network graph. Experimental results show that COPA-BC is very effective in quickly detecting overlapping communities in social networks. The value of v satisfies the condition (7) that makes the algorithm highly efficient in detecting the overlap. However in case v is very large (belonging coefficient is too small), then the number of labeling pairs is much over the reality and efficiency decreases. On experiments for real social networks, v should be $1 < v < 10$, then the algorithm is very efficient, yields good results suitable with reality. The algorithm has a very high parallelization capacity because each node can be independently updated in propagation steps, due to the use of the label propagation synchronous update mechanism.

REFERENCES

[1] Network repository. [Online]. Available: https://networkrepository.com/

[2] C. M. A. Clauset, M. E Newman, "Finding community structure in very large networks," *Physical Review E, 066111*, vol. 70, no. 6, 2004.

[3] J. K. A. Lancichinetti, S. Fortunato, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, Article ID 033015, 2009.

[4] M. Arab and M. Hasheminezhad, "Efficient community detection algorithm with label propagation using node importance and link weight," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, pp. 510–518, 2018.

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

[6] S. Gregory, "A fast algorithm to find overlapping communities in networks," *Lect. Notes Comput. Sci. 5211 408*, 2008.

[7] ——, "Finding overlapping communities in networks by label propagation," *New Journal of Physics, 103018.*, vol. 12, no. 10, 2010.

[8] J. Leskovec and Krevl. Datasets stanford large network dataset collection. [Online]. Available: https://snap.stanford.edu

[9] M. Needham and A. E. Hodler, *Graph Algorithms.* Oreilly, 2019.

[10] C. S. Saradha and D. P. Arul, "An optimized overlapping and disjoint community detection techniques using improved community overlap propagation algorithm in complex networks," *Advance Scientific Research JCR*, vol. 7, no. 4, pp. 782–790, 2020.

[11] T. Schank and D. Wagner, "Approximating clustering coefficient and transitivity," *Journal of Graph Algorithms and Applications*, vol. 9, no. 2, pp. 265–275, 2005.

[12] L. Tang and H. Liu, "Graph mining applications to social network analysis," *Managing and Mining Graph Data, Advances in Database Systems*, vol. 40, pp. 487–513, 2010.

[13] R. A. U. N. Raghavan and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E 036106*, vol. 76, 2007.

[14] H. L. J. P. Xuegang Hu, Wei He, "Role-based label propagation algorithm for community detection," *Social and Information Networks*, 2016.

Received on September 9, 2021
Accepted on January 23, 2022