Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

A Proposed Algorithm to Detect the Largest Community Based On Depth Level

Mohamed H. Farrag
Canadian International College, Cairo, Egypt.
Email:Mohamed_farrag @cic-cairo.com
Mona M. Nasr
Faculty of Computers and Information, Helwan University Cairo, Egypt.
Email:m.nasr@helwan.edu.eg

ABSTRACT

The incredible rising of online networks show that these networks are complex and involving massive data.Giving a very strong interest to set of techniques developed for mining these networks. The clique problem is a well-known NP-Hard problem in graph mining. One of the fundamental applications for it is the community detection. It helps to understand and model the network structure which has been a fundamental problem in several fields. In literature, the exponentially increasing computation time of this problem make the quality of these solutions is limited and infeasible for massive graphs. Furthermore, most of the proposed approaches are able to detect only disjoint communities. In this paper, we present a new clique based approach for fast and efficient overlapping community detection. The work overcomes the short falls of clique percolation method (CPM), one of most popular and commonly used methods in this area. The shortfalls occur due to brute force algorithm for enumerating maximal cliques and also the missing out many vertices thatleads to poor node coverage. The proposed work overcome these shortfalls producing NMC method for enumerating maximal cliques then detects overlapping communities using three different community scales based on three different depth levels to assure high nodes coverage and detects the largest communities. The clustering coefficient and cluster density are used to measure the quality. The work also provide experimental results on benchmark real world network to demonstrate the efficiency and compare the new proposed algorithm with CPM method, The proposed algorithm is able to quickly discover the maximal cliques and detects overlapping community with interesting remarks and findings.

Keywords -**Graph Mining, Maximal clique problem, Overlapping community detection, Social network analysis**

Date of Submission: Sep 14, 2017 Date of Acceptance: Sep 26, 2017

I. INTRODUCTION

There are several factors that has made the study of mining and analyzing large scale online communities gain enormous importance like availableness of huge amount of those large scale on-line communities' data. This massive data makes the graphs representing these networks are getting terribly complicated, millions of nodes are present across many various scientific domains. This results in increase computation time and makes most existing algorithms become impractical once the input graph is too complex. The clique problem, and the related maximal clique find applications in a wide variety of domains as information retrieval, community detection in networks, data mining, bioinformatics, disease classification , pattern recognition and analysis of financial networks. Particularly community detection in networks is one amongst the elemental applications for clique's problems. This problem is very hard and not nonetheless satisfactorily resolved, despite the large effort of a large interdisciplinary community of scientists performing on it over the past few years. the invention of those communities in large scale online communities can will more and more being leveraged as a strong, inexpensive tool for enterprises to drive business as rating prediction, Top-N recommendation, and link recommendation, trend analysis in citation networks and additionally improving recommender systems. One among largely used and common method in community detection is clique percolation method (CPM), is a simple algorithm which has been used often as a clique based approach for overlapping community detection. However this algorithm suffers from several shortfalls that make CPM is appropriate for networks with dense connected elements only and due to large scale of real networks and the exponentially increasing computation time,makes CPM algorithm impractical within real-world scenarios. The short falls of clique percolation method happens first as a result to the approach of enumerating maximal clique, it based on brute force algorithm its cost is proportional to the number of candidate solutions. The brute force algorithm becomes impractical for massive networks and unsuitable in real-world scenarios. For instance, for an entire graph of only 100 nodes, the algorithm will generate a minimum of 2^{99} - 1 different cliques.[5] A second shortage of CPM method happens because of the problem of missing out several vertices as a result to the restriction of using threshold K value. Whereas CPM only considers the totally connected sub graphs of size k the neglect Sub graphs containing several cliques which can be a part of existing community or generate new communities within the existing graph. This would possibly provide a

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

not clear community structure therefore the poor nodes coverage problem. To overcome the shortcomings mentioned earlier, the proposed algorithm in first part of the work propose NMC method to enhance the way of enumerating maximal clique in CPM method by reducing the search vertices and pruning specific nodes and edges which will not be a part of a maximal clique for this node to make the process of enumerating maximal cliques fast and efficient. second part of the work, The proposed algorithm efficiently detects overlapping communities using three different community scales based on three different depth levels to detect the largest community in given network and assures high nodes coverage for connected network that overcomes the poor coverage problem of the CPM method. This paper is organized as follows: section two discusses background and related work, section three demonstrates the proposed algorithm and section four explains the experiment while conclusion and future work are in the last section.

II. BACKGROUND AND RELATED WORK

Some main aspects concerning the nature and the structure of on-line networks that these network modeled by a graph that are the foremost usually used abstract data structures within the field of computer science, they enable a more complicated and wide-ranging presentation of data compared to link tables and tree structures. [1, 16, 17] A network is usually presented as a graph $G(V, E)$, where V is set of n nodes and E is set of m edges. Graph G consisting of n number of nodes denoting n individuals or the participants in the network. The connection between node i and node j is represented by the edge e_{ij} of the graph. The graph areoften represented by an adjacency matrix A in which $A_{ij} = 1$ in case there is an edge between i and j else $A_{ij} = 0$ [1, 3, 5].Another aspect is Cliques,is defined as “a set of vertices in which every pair of vertices is connected by an edge”.[5] Clique is a complete sub graph of G or in different words “is a maximum complete sub graph in which all nodes are adjacent to each other” , in a clique of size k, each node maintains degree$\geq k - 1$. Normally use cliques as a core or a seed to seek out larger communities. [5, 6, 12] another formal definition “A clique is a fully connected sub graph a set of nodes all of which are connected to each other. “ [7, 15]K-cliques are often outlined as complete graph with k vertices. [12] K-cliques are main structures in complicatednetworks, and a good way to seek out community structure. [7, 12]Maximal Clique “is a clique that's contained in no larger clique”.[7, 12, 14] Every maximal clique is a clique, by definition, however the other doesn't hold. Therefore there are always more cliques than maximal cliques. In different words “a clique is said to be maximal if it not contained in any other clique.” [7]Adjacent k-cliques, we are able to define adjacent K-cliques by two k-cliques that share $k - 1$ nodes. [7, 12]K-clique community (cluster or component),outlinedas “a union of all k-cliques that may be reached from each other through a series of adjacent k-cliques.” or “It is the union of all k-cliques that are k-Clique-connected to a particular k-clique.” [8, 12]

Community detection, real world complicated systems may be represented within the form of networks. To comprehend the in-depth structure of those systems, it’s necessary to review and analyze the networks. A trivial property of those networks is community structure obtained by splitting the network into many parts, within which connection between nodes are more dense than the remainder of the network. The sets of this sort of grouping are commonly referred as communities, however additionally called clusters, cohesive groups, or modules as there is no globally accepted unique definition. One among the restrictions of graph partitioning methods is that they typically need the user to specify the number of partitions, which cannot be identified before. One solution proposed to this problem is to use goodness metric as modularity to evaluate the partition of the graph at every step. However, this is often computationally expensive and might be infeasible for massive graphs. however just in case of community detection, it's not known that how many communities are present in the network and it is not at all obligatory for them to be of same size. The community detection approach assumes that almost all of real world networks, divide naturally into groups of nodes (community) with dense connections internally and sparser connections between groups, and therefore the experimenter’s job is only to find these already formed groups. The number of partitions and size of them are settled by the network itself and not set by the experimenter. So community detection is “the technique which aims to discover natural divisions of networks into groups based on strength of connection between vertices.” [1, 13]No formal definition of community is universally accepted, communities will have numerous properties, and community detection has been approached from many alternative views. Community detection is one among the foremost wide researched issues. Straightforward definition “A community is a densely connected group of nodes that is sparsely connected to the rest of the network” [18], Generally spoken community as “a module or cluster is typically thought of as a group of nodes with more and/or better interactions amongst its members than between its members and the remainder of the network”. [16] Primarily, community may be divided into two types; disjoint communities and overlapping communities. In disjoint communities nodes can be part of only a single community. A non-overlapping community structure or disjoint community structure may be outlined as “set of communities such that all vertices are included in exactly one community.” [2] However in overlapping communities partitions aren't essentially disjoint. There might be nodes that belong to more than one community [4, 18]. Usually in any on-line network a node may be part of more than one different group or community, thus for on-line networks, overlapping community detection technique ought to be thought of disjoint community detection technique. A number of community detection algorithms and methods have been proposed and deployed for the identification of communities in

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

literature. There have also been modifications and revisions to many methods and algorithms already proposed. Two views to divide the previous work in literature, first one depending on the nature of the relation between cluster members. This view divided into four categories. First category is node centric community detection wherever nodes satisfy different properties as complete mutuality which implies cliques; another property is reachability of members as k-clique [13]. Second category is group centric community detection (Density-Based Groups), it needs the total group to satisfy an explicit condition for instance the group density greater than or equal a given threshold and take away nodes with degree under the typical average degree. Third category is network centric community detection, must take into account connections within a network globally. Its goal to partition nodes of a network into disjoint sets, five different approaches utilized in network-centric community detection. First approach is clustering supported vertex similarity using Jaccard similarity and Cosine similarity. Second approach is Latent space models supported k-means clustering. Third approach is block model approximation based on exchangeable graph models. Fourth approach is spectral clustering are using minimum cut problem that the number of edges between the two sets is reduced. The fifth approach is modularity maximization by measures the strength of a community partition by taking into consideration the degree distribution. While the fourth category is hierarchy centric community detection, aims to build a hierarchical structure of communities supported network topology to permit the analysis of a network at different resolutions two representative approaches first divisive hierarchical clustering (top-down) and agglomerative hierarchical clustering (bottom-up). [3, 13] The strength of a tie is measured by edge betweenness that is the number of shortest paths that pass along with the edge. A summary of most algorithm used in community detection are shown in Table 1.

Algorithm	O	D	Dir	W	P	year
Speaker–listener label propagation (SLPA)	yes	Yes	yes	yes	yes	2012
Top graph clusters (TopGC)	yes	yes	yes	yes		2010
SVINET	yes		yes		yes	2013
Multithreaded (MCD)				yes	yes	2012
Core Groups Graph clusters (CGGC-RG)				yes		2012
Complex Network Cluster Detection			yes		yes	2011
Dense sub graph extraction (DSE)			yes		yes	2012
Speed and Performance in clustering (SPICI)				yes	yes	2010
CFinder	yes					2005
Fast Greedy				yes		2004
Label Propagation			yes	yes		2007
Leading eigenvector (LE)						2006

Table1. Summary of most algorithms used in community detection, indicates that the algorithm identifies$O$: overlapping communities or D: disjoint communities, can be run on Dir: directed or W: weighted graphs and if it requires input P: parameter. [18]

From the second view we can divide the methods for detecting overlapping communities in two categories. Clique based methods and non-clique based methods. [1]

Author (Algorithm)	Approach	Parameters
Palla et al. (CPM) ,	Clique percolation method Nodes	threshold weight
Lancichinetti et al.	Fitness function	Fitness function
Du et al. (ComTector)	Kernels-based clustering	Set of all kernels
Shen at al (EAGLE)	Agglomerative hierarchical clustering	Similarity between two communities
Evans et al.	Line graph, clique graph	Links, partition
Lee et al. (GCE)	Cliques-based expansion	Fitness function
Gregory et al. (CONGA, CONGO, Peacock algorithm)	Split betweenness vertex	Short paths ratio of max. Edge betweenness and max. split betweenness

Table2. Summary for Clique based methods for overlapping community detection [1]

In clique based methods for overlapping community detection a community may be consider as "a union of smaller, complete (fully connected) sub graphs that share nodes".[1] A k-clique is a fully connected sub graph consisting of k nodes[12]. Also we can consider "A k-clique community can be defined as a union of all k-cliques that can be reached from each other through a series of adjacent k-cliques". [1] One of the most widely used techniques to discover overlapping communities is the clique percolation method (CPM). [9, 18] CPMis an efficient algorithm in discovering overlapping communities; it has a wide range of application in social networks and biological networks. The basic idea of this method is that the idea of a k-clique community that was outlined as the union of all k-cliques (complete sub graphs of size k) that can be reached from each other through a series of adjacent k-cliques (where adjacency means sharing $k-1$ vertices). The k-clique community may be thought of a usual module (community, cluster or complex) because of its dense internal links and sparse external linkage with other part of the whole network. Then build the overlap matrix of thosek-cliques. Finally, a number of k-clique communities are detected by analysis the overlap matrix. [9, 10] CPM algorithm first detects all complete sub graphs of the network that are not parts of greater complete sub graphs. These maximal complete sub graphs are known as cliques, after the cliques are settled, the clique-clique overlap matrix is ready.In this symmetric matrix every row (and column) represents a clique and the matrix elements are equal to the number of shared vertices between the corresponding two cliques, and the diagonal entries are equal to the size of the clique. The k-clique-communities for a given value of k are equal to such connected clique components in which the neighboring cliques are linked to each other by at least $k-1$shared nodes. These components can be found by removing each off-diagonal entry. [1, 8, 12]a straightforward illustration for the extraction of the k-clique communities at $k=4$ using the clique clique overlap matrix.We tend to use a

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

sample network consists of eighteen nodes and thirty six edges as shown in Table3.

Table3. Sample network edge list

Network Edges					
N1,N2	N2,N10	N3,N9	N6,N8	N8,N9	N12,N13
N1,N3	N2,N18	N3,N10	N6,N9	N8,N10	N12,N14
N1,N4	N3,N4	N4,N5	N6,N10	N9,N10	N12,N15
N2,N3	N3,N6	N4,N6	N7,N8	N11,N12	N13,N14
N2,N4	N3,N7	N5,N6	N17,N11	N11,N13	N14,N15
N2,N9	N3,N8	N6,N7	N17,N12	N11,N14	N15,N16

– **Uncover maximal cliques**

CPM finds all cliques using brute-force algorithm starting by set A initially contains vertex v, Set B contains neighbors of v. then transfer one vertex w from B to A and remove vertices that are not neighbors of w from B. Then repeat until a reaches desired size if fail, step back and try other possibilities. [8, 12] The seven maximal cliques detected by CPM for the sample network are *{N3, N6, N8, N9, N10}, {N3, N6, N7, N8}, {N2, N3, N9, N10}, {N3, N4, N6}, {N11, N12, N17}, {N11, N12, N13, N14}, {N12, N14, N15}.*

– **Maximal Cliques to k-Clique Communities**

A straightforward illustration Table4 shows the steps of extraction of K-cliques communities at $K = 4$ using the clique overlap matrix. [12]

Table4. CPM steps to extract communities at $K = 4$ [8, 12, 28]

M	1	2	3	4	5	6	7
1	5						
2		4					
3			4				
4				3			
5					3		
6						4	
7							3

Step1: count nodes for each maximal cliques

M	1	2	3	4	5	6	7
1	5	3	3	2	0	0	0
2	3	4	1	2	0	0	0
3	3	1	4	1	0	0	0
4	2	2	1	0	0	0	0
5	0	0	0	0	0	2	1
6	0	0	0	0	2	4	1
7	0	0	0	0	1	1	0

Step2: Calculate the intersection nodes between each two maximal cliques

M	1	2	3	4	5	6	7
1	5	3	3	0	0	0	0
2	3	4	0	0	0	0	0
3	3	0	4	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	4	0
7	0	0	0	0	0	0	0

Step3: K=4 Change to zero any maximal clique less than 4

M	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0
2	1	1	0	0	0	0	0
3	1	0	1	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0

Step4: K=4 Change to zero any intersection between maximal clique less than K-1 =3

Step5: K=4 Change all non-zero to 1

Results:
(M1,M1)-(M1,M2)-(M1,M3)-(M6,M6)
Community1:(M1,M2,M3)
Community2:(M6)

The final K-cliques communities at $K = 4$ using CPM.Consists of two communities.$C_1 = \{N2, N3, N6, N7, N8, N9, N10\}$ and $C_2 = \{N11, N12, N13, N14\}$.

Another techniques are employed in literature as a clique based overlapping communities like the algorithm projected by Lancichinetti et al.[19] It performs a local exploration in order to find the community for each of the nodes. During this method, the nodes could also be revisited any number of times. The target was to detect local maximal based on a fitness function. AlsoCFindersoftware system was developed supporting CPM for overlapping community detection. Then Du et al. [20] proposed Comtectorto detect the overlapping communities using maximal cliques. At first, all maximal cliques within the network that form the kernels of a possible community are detect. Then, the agglomerative procedure is iteratively used to add the vertices left to their nearest kernels. The obtained clusters are adjusted by merging a combine of fractional communities to optimize the modularity of the network.EAGLE is another work using agglomerative hierarchical clustering based on maximal clique algorithm has been projected by Shen et al. [21] Firstly, maximal cliques are detected, and those smaller than a threshold are neglected. Then Subordinate maximal cliques are neglected, and the remaining give the initial communities. The similarity is found between these communities, and communities are repeatedly integrated along on the premise of this similarity. This is often used until one community remains at the end. Evans et al. [22] proposed that by partitioning the links of a network, the overlapping communities is also discovered. In another work, Evans et al. [23] used weighted line graphs. In another work, Evans [24] used clique graphs to discover the overlapping communities in real-world networks. Also Greedy clique expansion [25] at the first detect cliques in a network. These cliques act as seeds for expansion along with the greedy optimization of a fitness function. A community is discovered by expanding the selected seed and performing its greedy optimization via the fitness function proposed by Lancichinetti et al. [19] another work is Cluster-overlap Newman Girvan algorithm (CONGA) was proposed by Gregory. This algorithm was based on the split- betweenness algorithm of Girvan–Newman. CONGO optimized the proposed algorithm [26], which used a local betweenness measure, giving an improved complexity. A two-phase Peacock algorithm for overlapping community detection is proposed in Gregory [27] using disjoint community-detection methods. In the first phase, the network transformation was performed using the split betweenness concept proposed earlier by the author. Within the second phase, the remodeled network is processed by a disjoint community detection algorithm, and the discovered communities were transformed back to overlapping communities of the original network. [1]

III. LC-BDL (LARGEST COMMUNITY BASED ON DEPTH LEVEL) ALGORITHM

In this work, we propose LC-BDL (largest community based on depth level) algorithm which a new clique based algorithm for overlapping community detection, LC-BDL based on the assumption of one of the first and most popular and commonly used algorithm for overlapping community detection clique percolation method (CPM).[12] the assumption that a community is 'union of all k-cliques (complete sub graphs of size k) that can be reached from each other through a series of adjacent k-cliques", (where adjacency means sharing k-1 vertices). The k-clique community can be considered as a usual module (community, cluster) because of its dense internal

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

links and sparse external linkage with other part of the whole network. Clique percolation method (CPM) is a simple algorithm which has been used often in this area. However this algorithm suffers from several consideration that make CPM is appropriate for networks with dense connected parts only and due to large scale of real networks and the exponentially increasing computation time it makes CPM algorithms impractical in the real-world scenarios. *First consideration concern to the way of detecting the maximal cliques* which based on brute force algorithm as a general problem-solving technique consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement. While this algorithm will always find a solution if it exists, its cost is proportional to the number of candidate solutions which in many practical problems tends to grow very quickly as the size of the problem increases. CPM starts to determine the largest possible clique size in the studied graph from the degree-sequence. Starting with this clique size, CPM algorithm repeatedly chooses a node, extracts every clique of this size containing that node, For instance, for a complete graph of only 100 nodes, the algorithm will generate at least 2^{99} - 1 different cliques starting from any node in the graph.Deletes the node and its edges. When no nodes are left, the clique size is decreased by one and the clique finding procedure is restarted on the original graph. [5] The already found cliques' influence the further search since the yet unrevealed (smaller) cliques cannot be subsets of them. In the first part of the work the proposed algorithm produce the method NMC to overcome this point of consideration by enhancing the way of enumerating maximal cliques by reducing the search vertices and pruning the nodes and edges that will not be a part of a maximal clique for each node to make the process of enumerating maximal cliques fast and efficient. *Second consideration concern to the way of creating the communities*, occurs due to the problem of missing out many vertices as a result for restriction of using threshold K value which might give a not clear community structure and the poor nodes coverage problem**.** While CPM only considers the fully connected sub graphs of size k the neglect Sub graph containing many cliques which may be part of existing community or generate new communities in the existing graph. It may happen that a clique of size smaller than K but it adjacent to a series of adjacent sub cliques to another maximal clique of size K. To overcome this consideration in the second part of the work LC-BDL produce three different community scales depending on the target depth level for generating these communities. *First "Restricted community scale"* in which the depth level value equal zero it means that it detects the communities among only maximal cliques of threshold size K. *Second "Flexible community scale"* in which the depth level value is variant and flexible according to business target for detecting the communities, it means that it detects the communities among maximal cliques of threshold size K and its adjacent sub cliques of size equal maximal clique size till given depth L. This may lead to enlarge the detected communities in restricted scale by integrating these communities into larger communities and

---

Method: NMC (Nodes maximal cliques) for finding the maximal clique in a graph
Input: Input: Graph $G = (V, E)$
Output: Nodes maximal cliques

For $i = 1{:}n$
  $A_i = \{N(v_i) \cup v_i\}$
  for each node N_j in Set A_i
    $X_j = \{d(A_i) | d(N_{j+1}) \leq d(N_j) \ \forall N_j \in A_i\}$
    $X = \bigcup_j X_j$
  Loop
  Let x_j be an elementary element of X
  For $j = 1{:}|X|$
    $CARD_j = |\{N_j | d(N_j) \geq x_j \ \forall N_j \in A_i\}$
    If $CARD_j \geq x_j$ Then
      $B_i = \{A_i | d(N_j) \geq x_j\}$
      For each node N_k in Set B_i
        $X'_k = \{d(B_i) | d(N_{k+1}) \leq d(N_k) \ \forall N_k \in B_i\}$
        $X' = \bigcup_k X'_k$
      Loop
      Let x_k be an elementary element of X'
      $MINS = Min(X')$
      $CARDB_i = |B_i|$
      $CARDMINB_i = |\{x_k | x_k = MINS \ \forall x_k \in X'\}|$
      If $CARDB_i = CARDMINB_i$ then
        $MC_i = B_i$
      Else
        $MC_i = \{B_i | x_k \neq MINS \ \forall \ x_k \in X'\}$
    Else
  Loop
Loop

---

detect the hidden pattern of relation among these communities was discovered in restricted community scale. *Third "Power community scale"* in which the depth level value is the maximum to detect the largest communities could be reached by test all the maximal cliques adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique of size is larger than three to assure that no adjacent sub cliques of a maximal clique belongs to series of adjacent sub cliques for another maximal clique that helps to detect the largest communities in a given network without restriction to threshold size K. Also help to avoid the problem of poor node and cliques' coverage that may be part of existing community or generate new communities in the existing graph. In the following section, we present our LC-BDLalgorithm as a clique based algorithm overlapping community detection. LC-BDLalgorithm consists of two phases, in the first phase the algorithm enumerate nodes maximal cliques using NMCmethod, while phase two aims to discover the communities among the discovered maximal cliques in phase one according to three different community scales.

Let n be the number of vertices of the input graph$G = (V, E)$where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{(v_1, v_2), (v_1, v_3), \dots (v_i, v_z), \dots, (v_{n-1}, v_n)\}$ denote the set of vertices and edges, respectively. The set of vertices adjacent to a vertex v_i, the set of its neighbors is defined as $N(v_i) = \{v_z | v_i \in E\}$and the cardinality of

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

$N(v_i)$is denoted by $d(v_i)$, the maximal cliques denotes by $MC \subseteq V$. The community detection problem is typically formulated as finding a partition $C = \{v_1, v_2, \dots, v_k\}$of G,

For the set A_{12}						
N_i	N_{12}	N_{11}	N_{14}	N_{13}	N_{15}	N_{17}
$d(A_i) = X$	5	4	4	3	2	2

where $\forall vK \in G$.$C$is also known as a clustering of G. We use N to denote the number of resulting communities, that is $|C| = N$.

3.1. Phase1: Enumerate Maximal Cliques

NMC method aims to optimize the process of enumerating the maximal cliques to reduce time cost and enhance performance of the algorithm. To discover nodes maximal cliques NMC method reducing the search vertices and pruning the nodes and edges that will not be a part of a maximal clique for each node in two steps, first step generate set$A_i = \{N(v_i) \cup v_i\}$ then compute $d(A_i)$ equal the cardinality of $N(v_i)$ for each member in set A_i . Then set A_i sorted descending according to $d(A_i)$. The method iterate the next procedure till $CARD_j \geq x_j$by let $CARD_j$ equal cardinality of nodes in set A_i that its $d(N_j) \geq x_j$then compare $CARD_j$ against x_j, if $CARD_j$ is greater than or equal x_j the method returns set $B_i = \{A_i | d(N_j) \geq x_j\}$ by remove all nodes that will not be a part of a maximal clique for this vertex in case else the method uses the next x_j.

In second step for each node $N_k$in Set B_i the method assign X'_kequal $d(B_i)$for each member in set B_i. Then let $X' = \bigcup_k X'_k$Then set B_i sorted descending according to $d(B_i)$. Then three variables are used first $MINS$ equal the minimum value in X' , $CARDB_i$ equal the cardinality of set B_i and$CARDMINB_i$equal the cardinality of set B_i where $X'_k = MINS$. Then compare $CARDB_i$ against $CARDMINB_i$, if $CARDB_i$ is equal $CARDMINB_i$ the method returns maximal clique equal set B_i case else the method exclude all vertices where itsx_k equal to the $MINS$ and returns maximal clique equal set B_i where $x_k \neq MINS$. Finally define the set $MC = \{MC_1, MC_2, \dots, MC_w\}$.

A straightforward illustration, LC-BDL algorithm uses a tiny network consists of eighteen nodes and thirty six edges as shown in Table3.

For the node N_{12} the method begins togenerate set A_{12}, A_{12}= $\{N_{11}, N_{13}, N_{14}, N_{15}, N_{17}, N_{12}\}$ then compute $d(A_i)$equal the cardinality of $N(v_i)$. Then set A_i sorted descending according to $d(A_i)$ as shown in Table5.

Table5. For the set A_{12} sorted descending according to $\boldsymbol{d(A_i)}$

The method iterate the next procedure till $CARD_j \geq x_j$by let $CARD_j$ equal cardinality of nodes in set A_{12} that its $d(N_j) \geq x_j$then compare $CARD_j$ against x_j, if $CARD_j$ is greater than or equal x_j the method returns set $B_{12} = \{A_{12} | d(N_j) \geq x_j\}$ by removing all the nodes that will not be a part of a maximal clique for node N_{12}, if the condition is not satisfied the method uses the next x_j. The result is set $B_{12} = \{N_{11}, N_{13}, N_{14}, N_{12}\}$ excluding nodes $\{N_{15}, N_{17}\}$ from set A_{12} as shown in Table6.

Table6. Iteration for generating set B_{12}

x_j	$CARD_j$	Nodes that satisfy $CARD_j \geq$	set B_{12}
5	1	N_{12}	
4	3	N_{12},N_{11},N_{14}	{ N_{12},N_{11},N_{14},N_{13}}
3	4	N_{12},N_{11},N_{14},N_{13}	

In second step for each node $N_k$in Set B_{12} the method assign X'_kequal $d(B_{12})$for each member in set B_{12}. Then let $X' = \bigcup_k X'_k$then set B_{12} sorted descending according to $d(B_{12})$ as shown in Table7.

Table7. For the set B_{12} sorted descending according to $d(B_i)$

For the set B_{12}				
N_i	N_{12}	N_{14}	N_{11}	N_{13}
$d(B_i) = X'$	3	3	3	3

Then three variables are used first $MINS$ equal the minimum value in X' , $CARDB_i$ equal the cardinality of set B_i and$CARDMINB_i$equal the cardinality of set B_{12} where $X'_k = MINS$. Then compare $CARDB_{12}$ against $CARDMINB_{12}$, if $CARDB_{12}$ is equal $CARDMINB_{12}$ the method returns maximal clique equal set B_{12} case else the method exclude all vertices where itsx_k equal to the $MINS$ and returns maximal clique equal set B_{12} where$x_k \neq MINS$.As a result $MINS = 3$,$CARDB_i = 4$ and$CARDMINB_i = 4$while$CARDB_i$= $CARDMINB_i$the method returns setB_{12}as maximal clique MC_i ={N_{11}, N_{13}, N_{14}, N_{12}}.As a final result for phase one NMC method returns seven maximal cliques as *MC={(N11,N12,N13,N14),(N12,N14,N15),(N11,N12, N17),(N10, N2, N3 ,N9),(N10,N3, N6,N8,N9),(N3,N4,N6),(N3,N6,N7,N8)}* for the given sample network.

3.2. Phase2: Discover the communities

LC-BDL algorithm aims to discover the communities among the discovered maximal cliques in phase one according to three different community scales in two steps. Step1 aims to create test cliques list and generate adjacent list among them. This may be generated according to one of three different community scales depending on the target depth level for generating these communities. First "*Restricted community scale*" in which the depth level value equal zero it means that it detects the communities among only maximal cliques of threshold size K. Second "*Flexible community scale*" in which the depth level value is variant and flexible according to business target for detecting the communities, it means that it detects the communities among maximal cliques of threshold size K and its adjacent sub cliques of size equal maximal clique size till given depth L. Third "*Power community scale*"in which the depth level value is the maximum value to detect the largest communities could be reached by testing all the maximal cliques adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique whose size is greater than three to assure that no sub cliques of a maximal clique belongs to series of adjacent sub cliques for another maximal clique. This helps to detect the largest communities in a

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

given network without restriction to threshold value K. While step two aims to generate the communities among adjacent sub cliques of step one by detecting the communities' seeds then generate communities among the discovered seeds.

3.2.1. "Restricted community Scale" - Zero Depth level

Depth level $L = 0$ in the *restricted community scale,* it means that it detects the communities among only maximal cliques of threshold size K.

Step 1: Generate adjacent list

The algorithm in the *restricted community scale* aims first to generate the test maximal cliques among only the maximal cliques was detected in first phase and according to threshold of size K where $K = 3,4,5 \dots t < \infty$ and depth level L where $k - L \geq 3 \; \forall \; L = 0,1,2,..r < \infty$. We denote test maximal cliques byTMC is the set of maximal cliques of size greater than or equal K.

$$TMC = \{MC_i || MC_i| \geq K\}$$

As a simple illustration, we use $K = 4$ and depth level $L = 0$, it means that the algorithm will select all maximal cliques its size greater than or equal four among the seven maximal cliques was detected in previous phase for the given sample network, TMC = *{N11, N12, N13, N14}, {N10, N2, N3, N9}, {N10, N3, N6, N8, N9}, and {N3, N6, N7, N8}*. Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ is all possible sub cliques of size equal $TMC_i - L$, where TMC_i is the i^{th} tuple in TMC, using the power set $P(.)$ we can write

$$TMCS_i = \{D_i \in P(TMC_i) || D_i| = |TMC_i| - L\}$$

Therefore the depth level L equal zero in the *restricted community scale* thus $TMCS_i$= TMC_i = *{N11, N12, N13, N14}, {N10, N2, N3, N9}, {N10, N3, N6, N8, N9}, and {N3, N6, N7, N8}*.

Now define a set

$$TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\}$$

Consider the set

$$BT = \{(TMCS_1, TMCS_1), (TMCS_2, TMCS_2), \dots, (TMCS_h, TMCS_h)\}$$

Then LC-BDL algorithm define adjacent test list by the set ATas a list of testing each sub clique of $TMCS$ with the rest of $TMCS$ union sub clique with itself.

$$AT = \{AT \in P(TMCS) || AT_i| = 2) \bigcup BT\}$$

$$= \{(AT_{g1}, AT_{u1}), (AT_{g2}, AT_{u2}), \dots, (AT_{gq}, AT_{uq})\}$$

Finally to generate the adjacent list the algorithm perform the needed computation for each tuple in $AT$by detecting the cardinality for each sub clique andcalculate the adjacently vertices between these two sub cliques for the i^{th}tuple AT_i to be considered adjacent if one of two sub cliques share greater than or equal its cardinality -1. LC-BDL algorithm produce only six adjacent sub cliques among the ten tuples in AT as final adjacent list for the *restricted community scale* using the sample network as shown in Table8.

Table8. Step1 result for *restricted community scale* using the sample network where the cardinality of the first sub clique is$|AT_{g1}|$the cardinality of the second sub clique is$|AT_{u1}|$, $d(AT_{g1}, AT_{u1})$ is the adjacent vertices between the two sub cliques and the R column equal 0 for non-adjacent tuples and 1 show the adjacent tuples.

AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$	R
1	4	1	4	4	1
1	4	2	4	0	0
1	4	3	5	0	0
1	4	4	4	0	0
2	4	2	4	4	1
2	4	3	5	3	1
2	4	4	4	1	0
3	5	3	5	5	1
3	5	4	4	3	1
4	4	4	4	4	1

Step2: Detect the communities

Step2 aims to generate the communities among the adjacent tuples was detected in previous step. First by detect the communities' seeds and then uses these seeds to generate the final communities.

A. Detect communities' seeds

LC-BDL algorithm creates communities' seeds by selecting the sub cliques without duplicates among the adjacent list was created in the previous step. Then retrieve back the maximal cliques instead of sub cliques.LC-BDL algorithm detects four seeds from sub cliques in adjacent tuples *{1, 4, 5, 7}.*

B. Generate communities

Merge each two communities' seeds if they share one adjacent sub clique and loop until no possible merge to produce the largest community among these communities' seeds.LC-BDLresults for *restricted community scale*with depth level L equal zero and threshold Kequal four is$|C| = 2$, C_1= *{N11, N12, N13, N14}* an C_2 = *{N10, N2, N3, N6, N7, N8, N9}* .

3.2.2. "Flexible community Scale" – variant depth level

The depth level value is variant and flexible according to business target in *"Flexible community scale"*. Detecting communities in this scale based on detects the communities among only maximal cliques of threshold K and its adjacent sub cliques of size equal maximal clique size till given depth L. If the maximal clique size is seven and the depth level is 2 it means that the list will contain all adjacent sub cliques for this maximal clique of size equal five. This give the algorithm the ability to check if there is any adjacent sub clique of size $K - N$ which already are adjacent cliques because it belongs to a sequences of adjacent sub cliques from one maximal clique. To check if these sub cliques are adjacent with the rest of all other maximal cliques or its adjacent sub cliques. This gives LC-BDL algorithm the ability to detect this type of communities that consists of a series of adjacent sub cliques till given depth level.

Step 1: Generate adjacent list

The algorithm in the *flexible community scale* aims first to generate the test maximal cliques among only the maximal cliques was detected in first phase and according to threshold of size K where $K = 3,4,5 \dots t < \infty$ and depth level L where $k - L \geq 3 \; \forall \; L = 0,1,2,..r < \infty$. We

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

denote test maximal cliques byTMC is the set of maximal cliques of size greater than or equal K.

$$TMC = \{MC_i || MC_i| \geq K\}$$

As a simple illustration, we use K equal four and since that the triangle structure or 3-clique is a basic sub-structure of any clique whose size is larger than three the depth level L equal one. it means that the algorithm will select all maximal cliques its size greater than or equal four among the seven maximal cliques was detected in previous phase for the given sample network TMC = *{N11, N12, N13, N14}, {N10, N2, N3, N9}, {N10, N3, N6, N8, N9}, and {N3, N6, N7, N8}*. Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ is all possible sub cliques of size equal $TMC_i - L$, where TMC_i is the i^{th} tuple in TMC, using the power set $P(.)$ we can write$TMCS_i = \{D_i \in P(TMC_i) || D_i| = |TMC_i| - L\}$.

As an illustration for TMC_3 contains vertices *{ N10, N3, N6, N8, N9}* the algorithm uses only all the five adjacent sub cliques of size equal four $TMCS_3$ =*{(N10, N3, N6 , N8), (N10, N3 , N6 , N9), (N10, N3 , N8, N9), (N10, N6, N8, N9), (N3, N6 , N8 , N9)}*. Now define a set

$$TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\}$$

$TMCS$ = *{N11, N12, N13}, {N11, N12, N14}, {N11,N13,N14}, {N12, N13, N14}, {N10, N2, N3}, {N10, N2, N9}, {N10, N3, N9}, {N2, N3, N9}, {N10, N3, N6, N8}, {N10, N3, N6, N9}, {N10, N3, N9, N8}, {N9, N3, N6, N8}, { N3, N6, N7},{ N3, N6, N8}, { N3, N8, N7}, { N8, N6, N7}*. Consists of seventeen sub cliques for the four maximal cliques when $K = 4$ and $L = 1$. Then LC-BDL algorithm define adjacent test list by the set ATas a list of testing each sub clique of $TMCS$ with the rest of $TMCS$.

$$AT = \{AT \in P(TMCS) || AT_i| = 2\}$$
$$= \{(AT_{g1}, AT_{u1}), (AT_{g2}, AT_{u2}), \dots, (AT_{gq}, AT_{uq})\}$$

But to enhance the performance and computation cost the LC-BDL algorithm avoids generating a test case between two sub cliques if they belong to the same $TMCS_i$, in some cases the depth level is greater than one the algorithm uses only the minimum size for the adjacent sub cliques because the goal is to check if any adjacent sub clique of a maximal clique is adjacent to another maximal clique or even belongs to a series of adjacent sub cliques to another maximal clique. So the small adjacent sub cliques is better to use instead of using all possible sub cliques for a maximal clique to enhance the performance and reduce computation cost to be available to use in real world networks.

Finally to generate the adjacent list the algorithm perform the needed computation for each tuple in $AT$by detecting the cardinality for each sub clique andcalculate the adjacently vertices between these two sub cliques for the i^{th}tuple AT_i to be considered adjacent if one of two sub cliques share greater than or equal its cardinality -1. LC-BDL algorithm produce only twenty eight adjacent sub cliques among the one hundred and eight tuples in AT as final adjacent list for the *flexible community scale* using the sample network as shown in Table9.

Table9. Step1 result for adjacent tuples in *flexible community scale* using the sample network where the cardinality of the first sub clique is$|AT_{g1}|$the cardinality of the second sub clique is$|AT_{u1}|$, $d(AT_{g1}, AT_{u1})$ is the adjacent vertices between the two sub cliques.

AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$	AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$
2	3	3	4	2	7	4	10	3	3
2	3	7	4	2	7	4	14	3	2
2	3	11	4	2	8	3	11	4	2
3	4	4	3	2	8	3	15	4	2
3	4	8	3	3	8	3	17	4	3
3	4	10	3	2	10	3	11	4	3
3	4	12	3	2	10	3	15	4	2
3	4	16	3	2	10	3	17	4	2
4	3	7	4	2	11	4	12	3	2
4	3	17	4	2	11	4	14	3	2
6	3	7	4	2	12	3	17	4	2
6	3	11	4	2	14	3	17	4	2
6	3	15	4	2	15	4	16	3	2
7	4	8	3	2	16	3	17	4	2

Step2: Detecting the communities.

Step2 aims to generate the communities among the adjacent tuples was detected in previous step. First by detect the communities' seeds and then uses these seeds to generate the final communities.

A. Detect communities' seeds

LC-BDL algorithm creates communities' seeds by selecting the sub cliques without duplicates among the adjacent list was created in the previous step, then retrieve back the main maximal clique instead of adjacent sub clique.LC-BDL algorithm detects four seeds in adjacent tuples *{1, 4, 5, 7}*.

B. Generate communities

Merge each two communities' seeds if they share one adjacent sub clique and loop until no possible merge to produce the largest community among these communities' seeds. LC-BDLresults for *flexible community scale*with depth level L equal one and threshold Kequal four is $|C| = 2$, C_1= *{N11, N12, N13, N14}* an C_2 = *{N10, N2, N3, N6, N7, N8, N9}*.

3.2.3. "Power community scale" - Maximum depth level

The depth level value in *power community scale* is maximum, This scale based on detect the communities among all the maximal cliques' adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique whose size is larger than three to check that no sub cliques of a maximal clique belongs to series of adjacent sub cliques for another maximal clique. It helps to detect the largest communities in a given network without restriction for threshold size K as the restricted community scale where depth level equal zero or even in the flexible community scale where depth level is variant.

Step 1: Generate adjacent list

The algorithm in the *power community scale* aims first to generate the test maximal cliques among the maximal cliques was detected in first phase and according to threshold of size K where $K = 3, 4, 5 \dots t < \infty$ and depth level L where $k - L \geq 3 \; \forall \; L = 0, 1, 2, .. r < \infty$. We denote test maximal cliques byTMC is the set of maximal cliques of size greater than or equal K.

$$TMC = MC$$

As a simple illustration, we use *TMC*set equal *MC* set was created in first phase it means that the algorithm will select all the seven maximal cliques was detected in previous

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

phase for the given sample network, TMC = *{ N11, N12, N13, N14},{ N12, N14, N15},{ N11, N12, N17},{ N10, N2, N3, N9},{ N10, N3, N6, N8, N9},{ N3, N4, N6} and { N3, N6, N7, N8}*. Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$is all possible sub cliques of size equal 3, where TMC_i is the i^{th} tuple in TMC, using the power set $P(.)$ we can write$TMCSi = \{D_i \in P(TMC_i)||D_i| = 3\}$.As an illustration for TMC_1 =*{N11, N12, N13, N14}* the algorithm uses only all the four adjacent sub cliques of size equal three $TMCSi$ = *{(N11, N12, N13), (N11, N12, N14), (N11, N13, N14), (N12, N13, N14)}*.Now define a set

$$TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\}$$

Therefore the depth level L equal max in the power community scale thus $TMCS$ consists of twenty five sub cliques for the seven maximal cliques. $TMCS$ = *{(N11, N12, N13), (N12, N14, N15), (N11, N12, N17), (N10, N2, N3), (N10, N3, N6), (N3, N4, N6), (N3, N6, N7), (N11, N12, N14), (N10, N2, N9), (N10, N3, N8), (N3, N6, N8), (N11, N13, N14), (N10, N3, N9), (N10, N6, N8), (N3, N7, N8), (N12, N13, N14), (N2, N3, N9), (N3, N6, N8), (N6, N7, N8), (N10, N3, N9), (N10, N6, N9), (N3, N6, N9), (N10, N8, N9), (N3, N8, N9), (N6, N8, N9)}*. Consider the set

$$BT = \{(TMCS_1, TMCS_1), (TMCS_2, TMCS_2), \dots, (TMCS_h, TMCS_h)\}.$$

Then LC-BDL algorithm define adjacent test list by the set ATas a list of testing each sub clique of $TMCS$ with the rest of $TMCS$.

$$AT = \{AT \in P(TMCS)||AT_i| = 2\}$$
$$= \{(AT_{g1}, AT_{u1}), (AT_{g2}, AT_{u2}), \dots, (AT_{gq}, AT_{uq})\}$$

Two main enhancement used in the algorithm to enhance the performance and reduce computation cost. The LC-BDL algorithm avoids generating a test case between two sub cliques if they belong to the same $TMCS_i$. The algorithm also uses the smallest adjacent sub cliques for each maximal clique instead of testing all adjacent sub clique which increase the time and computation cost and therefore make the suitable to use on real networks. Finally to generate the adjacent list the algorithm perform the needed computation for each tuple in $AT$by detecting the cardinality for each sub clique andcalculate the adjacently vertices between these two sub cliques for the i^{th}tuple AT_i to be considered adjacent if one of two sub cliques share greater than or equal its cardinality -1. LC-BDL algorithm produces forty one adjacent sub cliques among the two hundred and thirty seven tuples in AT as final adjacent list for the *power community scale* using the sample network as shown in Table10.

Table10. A sample of step1 result for adjacent tuples in *Power community scale* using the sample network where the cardinality of the first sub clique is$|AT_{g1}|$the cardinality of the second sub clique is$|AT_{u1}|$, $d(AT_{g1}, AT_{u1})$ is the adjacent vertices between the two sub cliques.

AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$
1	3	3	3	2
2	3	8	3	2
2	3	16	3	2
3	3	8	3	2
4	3	5	3	2
4	3	10	3	2
4	3	20	3	2

Step2: Detecting the communities.

Step2 aims to generate the communities among the adjacent tuples was detected in previous step. First by detect the communities' seeds and then uses these seeds to generate the final communities.

A. Detect communities' seeds

LC-BDL algorithm creates communities' seeds by selecting the sub cliques without duplicates among the adjacent list was created in the previous step. Then retrieve back main maximal clique instead of sub cliques.LC-BDL algorithm detects seven seeds in adjacent tuples *{1, 2, 3, 4, 5, 6, 7}*.

B. Generate communities

Merge each two communities' seeds if they share one adjacent sub clique and loop until no possible merge to produce the largest community among these communities' seeds.LC-BDLresults for *flexible community scale*is$|C| = 2$, C_1= *{N11, N12, N13, N14, N15, N17}* and C_2 = *{N10, N2, N3, N4, N6, N7, N8, N9}*.

IV. EXPERIMENT

To evaluate the efficiency and simplicity of the proposed algorithm, Zachary's Karate Club Network isa standard real world networks generally considered as benchmark for community detection are used as experimental dataset for the algorithm implementation.It is an undirected, un-weighted network having 34 nodes and 78 edges as shown in Table11. [11]

Table11. Zachary's Karate Club Network edges. [11]

Zachary's Karate Club Network edges						
2, 1	7, 6	11, 6	18, 1	28, 25	33, 3	33, 32	34, 24
3, 1	8, 1	12, 1	18, 2	29, 3	33, 1	34, 9	34, 27
3, 2	8, 2	13, 1	20, 1	30, 24	33, 15	34, 10	34, 28
4, 1	8, 3	13, 4	20, 2	30, 27	33, 16	34, 14	34, 29
4, 2	8, 4	14, 1	22, 1	31, 2	33, 19	34, 15	34, 30
4, 3	9, 1	14, 2	22, 2	31, 9	33, 21	34, 16	34, 31
5, 1	9, 3	14, 3	26, 24	32, 1	33, 23	34, 19	34, 32
6, 1	10, 3	14, 4	26, 25	32, 25	33, 24	34, 20	34, 33
7, 1	11, 1	17, 6	28, 3	32, 26	33, 30	34, 21	
7, 5	11, 5	17, 7	28, 24	32, 29	33, 31	34, 23	

4.1. Phase1: Enumerate Maximal Cliques

NMC method discovers thirteen maximal cliques in a Zachary's Karate Club network. As a final result for phase one, MC =*{1, 13, 4}, {1, 14, 2, 3, 4}, {17, 6, 7}, {1, 18, 2}, {1, 2, 20}, {1, 2, 22}, {25, 26, 32}, {24, 28, 34}, {29, 32, 34}, {24, 30, 33, 34}, {31, 33, 34, 9}, {1, 6, 7} and {1, 2, 3, 4, 5}*.

4.2. Phase2: Discover the communities

4.2.1. "Restricted community Scale" - Zero depth level

Step 1: Generate adjacent list

The algorithm uses $K = 4$ and depth level $L = 0$ in *restricted community scale* and according to $TMC = \{MC_i||MC_i| \geq K\}$, TMC is all maximal cliques its size greater than or equal four among the thirteen maximal cliques was detected in previous phase for Zachary's Karate Club network TMC =*{{1, 14, 2, 3, 4}, {24, 30, 33, 34}, {31, 33, 34, 9} and {1, 2, 3, 4, 5}*. Then LC-BDL algorithm generates adjacent test list by first generates

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

$TMCS_i$, where $TMCS_i = \{D_i \in P(TMC_i) || D_i| = |TMC_i| - L\}$. Therefore the depth level $L = 0$ thus $TMCS_i = TMC_i$ = *{{1, 14, 2, 3, 4}, {24, 30, 33, 34}, {31, 33, 34, 9} and {1, 2, 3, 4, 5}}*. Then LC-BDL algorithm define adjacent test list by the set $AT$and produce only five adjacent sub cliques among the ten tuples in AT as final adjacent list for the *restricted community scale* using the Zachary's Karate Club network as shown in Table12.

Table12. Step1 result for *restricted community scale* using the Zachary network where the cardinality of the first sub clique is$|AT_{g1}|$, the cardinality of the second sub clique is$|AT_{u1}|$, $d(AT_{g1}, AT_{u1})$ is the adjacent vertices between the two sub cliques and the R column equal 0 for non-adjacent tuples and 1 show the adjacent tuples.

AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$	R
1	5	1	5	5	1
1	5	3	4	0	0
1	5	4	4	0	0
1	5	5	5	4	1
3	4	3	4	4	1
3	4	4	4	2	0
3	4	5	5	0	0
4	4	4	4	4	1
4	4	5	5	0	0
5	5	5	5	5	1

Step2: Detecting the communities.

A. Detect communities' seeds

LC-BDL algorithm detects four seeds in adjacent tuples in AT set *{2, 10, 11, 13}*.

B. Generate communities

LC-BDL results for *restricted community scale*with depth level $L = 0$ and threshold $K = 4$is$|C| = 3$, C_1= *{1, 2, 3, 4, 8, 14}*, C_2 = *{24, 30, 33, 34}* and C_3 = *{9, 31, 33, 34}*.

4.2.2. "Flexible community Scale" – variant depth level

Step 1: Generate adjacent list

The algorithm uses $K = 4$ and since that the triangle structure or 3-clique is a basic sub-structure of any clique whose size is larger than three the depth level $L = 1$. And according to $TMC = \{MC_i || MC_i| \geq K\}$, TMC = *{{1, 14, 2, 3, 4}, {24, 30, 33, 34}, {31, 33, 34, 9} and {1, 2, 3, 4, 5}*. Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ is $TMCS_i = \{D_i \in P(TMC_i) || D_i| = |TMC_i| - L\}$, Now define a set $TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\}$ then $TMCS$ =*{(1, 14, 2, 3), (24, 30, 33), (31, 33, 34), (1, 2, 3, 4), (1, 14, 2, 4),(24, 30, 34), (31,33,9),(1, 2, 3, 8), (1, 14, 3, 4), (24, 33, 34), (31, 34, 9),(1, 2, 4, 8), (1, 2, 3, 4), (30, 33, 34), (33, 34, 9), (1, 3, 4, 8), (14, 2, 3, 4), (2, 3, 4,8)}*. Consists of eighteen sub cliques for the four maximal cliques when $K = 4$ and $L = 1$. Then LC-BDL algorithm define adjacent test list by the set $AT$and produce only seventeen adjacent sub cliques among the one hundred and twenty tuples in AT as final adjacent list for *flexible community scale* using the sample network as shown in Table13.

Table13. Step1 result for *flexible community scale* using the Zachary network where the cardinality of the first sub clique is$|AT_{g1}|$the cardinality of the second sub clique is$|AT_{u1}|$, $d(AT_{g1}, AT_{u1})$ is the adjacent vertices between the two sub cliques.

AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$
1	4	4	4	3
1	4	8	4	3
3	3	10	3	2
3	3	14	3	2
4	4	5	4	3
4	4	9	4	3
4	4	13	4	4
4	4	17	4	3
5	4	12	4	3
8	4	13	4	**3**
9	4	16	4	3
10	3	15	3	2
12	4	13	4	3
13	4	16	4	3
13	4	18	4	3
14	3	15	3	2
17	4	18	4	3

Step2: Detecting the communities.

A. Detect communities' seeds

LC-BDL algorithm detects four seeds from sub cliques was created in the previous step by retrieve back the maximal cliques instead of sub cliques in adjacent tuples in AT set*{2, 10, 11, 13}*.

B. Generate communities

LC-BDLresults for *flexible community scale*with depth level $L = 1$ and threshold $K = 4$is $|C| = 2$, C_1= *{1, 2, 3, 4, 8, 14}* and C_2 = *{9, 24, 30, 31, 33, 34}*.

4.2.3. "Power community scale" - Maximum depth level

Step 1: Generate adjacent list

According to $TMC = MC$, the algorithm select the thirteen maximal cliques was detected in previous phase , TMC = *{1, 13, 4}, {1, 14, 2, 3, 4}, {17, 6, 7}, {1, 18, 2}, {1, 2, 20}, {1, 2, 22}, {25, 26, 32}, {24, 28, 34}, {29, 32, 34}, {24, 30, 33, 34}, {31, 33, 34, 9}, {1, 6, 7} and {1, 2, 3, 4, 5}*. Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$where$TMCSi = \{D_i \in P(TMC_i) || D_i| = 3\}$. Now define a set $TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\}$. Therefore the depth level L equal max in the power community scale thus $TMCS$ consists of thirty seven sub cliques for the thirteen maximal cliques as shown in Table14.

Table14. Thirty seven test sub cliques adjacent sub cliques for the thirteen maximal cliques for *power community scale*

$TMCS$					
{1,13,4}	{14,2,4}	{1,2,22}	{30,33,34}	{1,2,4}	{2,4,8}
{1,14,2}	{1,3,4}	{25,26,32}	{31,33,34}	{1,3,4}	{3,4,8}
{1,14,3}	{14,3,4}	{24,28,34}	{31,33,9}	{2,3,4}	
{1,2,3}	{2,3,4}	{29,32,34}	{31,34,9}	{1,2,8}	
{14,2,5}	{17,6,7}	{24,30,33}	{33,34,9}	{1,3,8}	
{1,14,4}	{1,18,2}	{24,30,34}	{1,6,7}	{2,3,8}	
{1,2,4}	{1,2,20}	{24,33,34}	{1,2,3}	{1,4,8}	

Then LC-BDL algorithm define adjacent test list by the set $AT$and produces eighty tuples cliques among the five hundred and sixty four tuples in AT as final adjacent list

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

for the *power community scale* using Zachary's Karate Club network.

Table15.A sample of Step1 result for adjacent tuples in *power community scale* using the Zachary network where the cardinality of the first sub clique is$|AT_{g1}|$the cardinality of the second sub clique is$|AT_{u1}|$, $d(AT_{g1}, AT_{u1})$ is the adjacent vertices between the two sub cliques.

AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$
1	3	6	3	2
1	3	7	3	2
1	3	9	3	2
1	3	29	3	2
1	3	30	3	2
1	3	35	3	2
2	3	13	3	2
2	3	14	3	2
2	3	15	3	2
2	3	28	3	2
2	3	29	3	2
2	3	32	3	2
3	3	28	3	2
3	3	30	3	2
3	3	33	3	2

Step2: Detecting the communities.

A. Detect communities' seeds

LC-BDL algorithm detects thirteen seeds in adjacent tuples *{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}.*

B. Generate communities

LC-BDLresults for *power community scale*is$|C| = 5$, C_1*={1, 2, 3, 4, 8, 13, 14, 18, 20},*C_2*={1, 6, 7, 17},*C_3*={25, 26, 32},*C_4*={9, 24, 28, 30, 31, 33, 34}* and C_5*={29, 32, 34}.*

V. RESULTS AND EVALUATION

Communities are detected for Zachary's Karate Club network using the proposed algorithm and CPM method, in literature it is found that generally value of k ranges from 3 to 6. Here k value is taken as 4. First part of the work success to overcome the shortfalls of CPM method as a result to the way of enumerating maximal clique, which based on brute force algorithm its cost is proportional to the number of candidate solutions. For instance, for a complete graph of only 100 nodes, the algorithm will generate at least 2^{99} - 1 different cliques. [5] The proposed method NMC succeeded to discover and enumerate maximal cliques producing the same result as the brute force algorithm of CPM method but without need to evaluate all the number of candidate solutions as brute force algorithm. NMC method discovers thirteen maximal cliques in a Zachary's Karate Club network, $MC = \{1, 13, 4\}, \{1, 14, 2, 3, 4\}, \{17, 6, 7\}, \{1, 18, 2\}, \{1, 2, 20\}, \{1, 2, 22\}, \{25, 26, 32\}, \{24, 28, 34\}, \{29, 32, 34\}, \{24, 30, 33, 34\}, \{31, 33, 34, 9\}, \{1, 6, 7\}$ and $\{1, 2, 3, 4, 5\}$.

Second part of the work success to overcome the shortfall of CPM method occurs due to the problem of missing out many vertices as a result to the restriction of using threshold K value. While CPM only considers the fully connected sub graphs of size k the neglect Sub graphs containing many cliques which may be part of existing community or generate new communities in the existing graph. This might give a not clear community structure and the poor nodes coverage problem. To overcome the shortcoming mentioned earlier the proposed algorithm produce three different scales with three different depth levels for the discoveredcommunities.First "*restricted community scale*" in which the depth level L value equal zero it means that it detects the communities among only maximal cliques of threshold size $K$and depth level L equal zero. Second "*flexible community scale*" in which the depth level L value is variant and flexible according to business target for detecting the communities, it means that it detects the communities among maximal cliques of threshold size K and its adjacent sub cliques of size equal maximal clique size till given depth L. This leads to enlarge the detected communities in *restricted community scale* by integrating these communities into larger communities and detect the hidden pattern of relation among these communities was discovered in *restricted community scale*. Third "*power community scale*"in which the depth level value is maximum to detect the largest communities could be reached by testing all the maximal cliques adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique of size is larger than three. This checks that no adjacent sub cliques for a maximal clique belongs to series of adjacent sub cliques for another maximal clique. This helps to detect the largest communities in a given network without restriction to threshold size $K$and help to avoid neglected nodes or cliques that may be part of existing community or generate new communities in the existing graph. CPM detects three communities$|C| = 3$, C_1*={1, 2, 3, 4, 8, 14},*C_2*={24, 30, 33, 34},*C_3*={9, 31, 33, 34}.* Vertices 33 and 34 are overlapped between last two communities. Total 22 vertices among 34 vertices are not included in any community though they are connected to the network. The covered nodes percentage equal 35.2%. The proposed algorithm cover this shortcoming by producing three different community scales as follow trying to include these vertices to the detected communities or even detect new communities among them on the basis of their depth level as discussed earlier. Initially with $k = 4$, LC-BDL *restricted community scale* produce same result as CPM method but without the cost of computing the adjacently matrix as CPM, it detects three communities$|C| = 3$ which are C_1*={1, 2, 3, 4, 8, 14},*C_2*={24, 30, 33, 34},*C_3*={9, 31, 33, 34}.* Vertices 33 and 34 are overlapped between last two communities. Total 22 vertices among 34 vertices are not included in any community and the covered nodes percentage equal 35.2%. *Flexible community scale* successes to integrate the three discovered communities in the CPM method and *restricted community scale* with threshold $k = 4$ and depth level $L = 1$ in two large communities, LC-BDL *flexible community scale* result is $|C| = 2$, C_1*= {1, 2, 3, 4, 8, 14}* and C_2 *= {9, 24, 30, 31, 33, 34}* with the same covered nodes percentage equal 35.2%, no overlapping nodes between the discovered communities. While the *power community scale* success to double the node covered ratio, make it equal 70.50% with total 24 vertices among 34 vertices are included in the discovered communities and also success to change the community

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

structure discovered by CPM method and the two previous community scales, LC-BDL *power community scale* result is$|C| = 5$, C_1={1, 2, 3, 4, 8, 13, 14, 18, 20},C_2={1, 6, 7, 17},C_3={25, 26, 32},C_4={9, 24, 28, 30, 31, 33, 34} and C_5={29, 32, 34} with 6 vertices are overlapped between discovered communities. The details of community structures detected by CPM and LC-BDL algorithm for Zachary's Karate Club network for k value as 4 are summarized and compared in Table16 and Fig. [1-3].

Table16. $D.L$: depth level; K: threshold value; $|AT|$: tuples cardinality of adjacent test list; $|adj(AT)|$: adjacent tuples cardinality of adjacent test list; S: number of maximal clique seeds; $|C|$: number of discovered communities; CV: number of vertices covered; CR: % of nodes covered; UVC: Number of nodes uncovered; OV: Number of overlapped nodes

Algorithm		$D.L$	K	$\|AT\|$	$\|adj(AT)\|$	S	$\|C\|$	CV	UVC	OV	CR
CPM			4	-	-	-	3	12	22	2	35.2%
LC-BDL	R.C.S	0	4	10	5	4	3	12	22	2	35.2%
	F.C.S	1	4	121	17	4	2	12	22	-	35.2%
	P.C.S	Max	-	564	80	13	5	24	10	6	70.5%

Figure1.$|C|$ Number of discovered communities

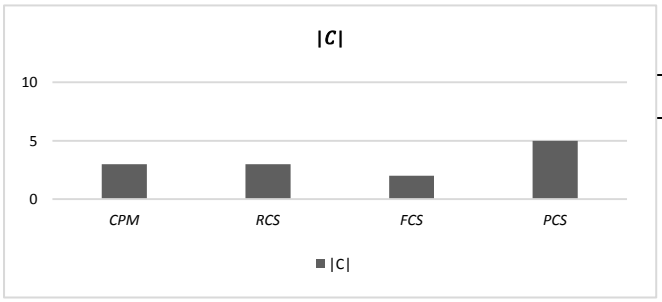
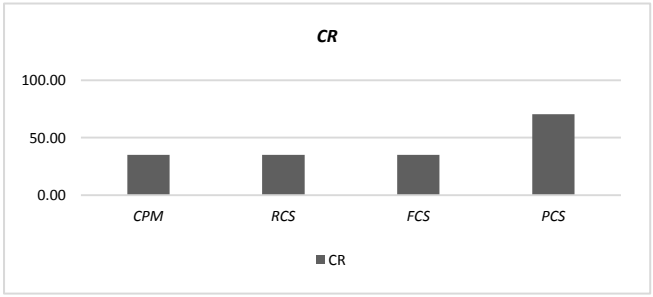


Figure2.CRdenote % of nodes covered



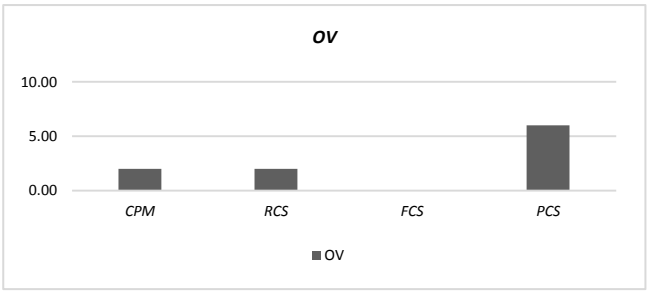


Figure3.OV Number of overlapped nodes

The LC-BDL set two types of parameters to ensure the quality of the goodness and performance metrics, while our algorithm detects cliques, adjacent k-cliques and overlapping communities, which all have clear definitions so the evaluation will depend on verify whether extracted communities satisfy the definition or not. While to evaluate the suitability and validity of our proposed algorithm in identifying the overlapping community detection in large scale networks, the average clustering coefficient and cluster density are used. For both CPM method and LC-BDL algorithm with k value as 4, LC-BDL algorithm and CPM have the same results in *restricted community scale*. And very high ratio for *flexible and power community scale*. The details are summarized and compared in Table17 and Fig. 4.

Table17.$|C|$denotes number of discovered communities; $Density$: % of the community density; $ClusterCoefficient$: % of the community cluster coefficient

$Algorithm$		$\|C\|$		$Density$	$Cluster\ Coefficient$
CPM		3	C_1	0.933	0.933
			C_2	1.000	1.000
			C_3	1.000	1.000
LC-BDL	R.C.S	3	C_1	0.933	0.933
			C_2	1.000	1.000
			C_3	1.000	1.000
	F.C.S	2	C_1	0.933	0.933
			C_2	0.733	0.867
	P.C.S	5	C_1	0.488888889	0.836
			C_2	0.833333333	0.833
			C_3	1	1.000
			C_4	0.619047619	0.819
			C_5	1	1.000

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

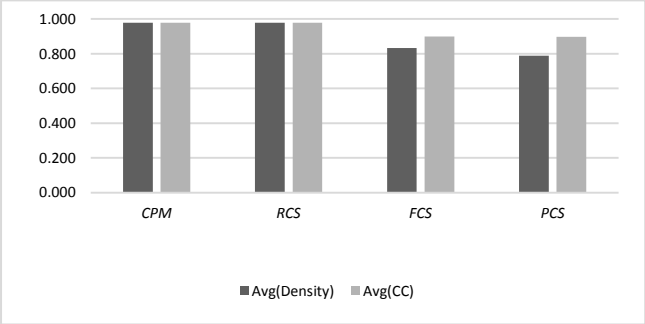


Figure4.$|C|$ % of the average cluster coefficient *(CC)* and density for discovered communities.

VI. CONCLUSION AND FUTURE WORK

In this work overlapping communities are identified in large scale online networks. New proposed clique based overlapping community detection algorithm has been studied. To quantify the discovered community structure the average clustering coefficient and cluster density are used. A benchmark classic real world network is used to test the algorithms. In the first part of work, a new method is proposed for the maximal clique problem which overcomes the problem of enumerating maximal clique in CPM method which based on brute force algorithm systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement. While this algorithm will always find a solution if it exists, its cost is proportional to the number of candidate solutions the brute force algorithm becomes impractical for large networks. For instance, for a complete graph of only 100 nodes, the algorithm will generate at least 2^{99} - 1 different cliques starting from any node in the graph. The proposed NMC method enhances the process of enumerating maximal cliques compared to the brute force algorithm by pruning specific nodes and edges. The proposed NMC method based on enumerating vertices maximal cliques by reducing the search vertices in two steps, first selects only the adjacent vertices for the tested vertex and excludes the vertices cannot be a part of existing maximal clique for this vertex. Then generate maximal clique among the rest vertices according to simple and native mathematical computation instead of test all candidate solutions as CPM method and make the process of enumerating maximal cliques fast and efficient. Hence, for large graphs many nodes and edges will be pruned, which will reduce the computation drastically. In second part of the work, The proposed algorithm efficiently detects overlapping communities using three different community scales based on three different depth level to detect the largest community in given network and assures high vertices coverage for connected network which overcomes the poor coverage problem of the CPM method. It is observed that based on the average clustering coefficient, cluster density and vertices coverage, proposed method gives better community structure compared to the clique percolation method. It can be concluded from the result that the community structure depends on the depth level for these communities and given threshold K. The community structure discovered by the CPM method integrated into larger communities and new communities discovered with high vertices cover ratio using LC-BDL algorithm.

Overlapping community detection is still a challenge. Though there are several proposed methods, but most of them not applicable to use for real large scale graphs due to the massive data for these graphs. Taking a huge amount of processing time. So emphasis should be given to effective algorithms which will be able to detect communities in large scale online networks in allowable time. In this work only un-weighted and undirected network has been taken into consideration. In future weighted and directed networks are needed to be considered for community detection. Also not covered vertices in the network may be assigned to the discovered communities using one of similarity measure to increase the vertices cover ratio. It is a suggested to apply the proposed LC-BDL algorithm using different data domains and study its accuracy and capacity on different scopes and natures.

REFERENCES

[1] Bedi, Punam, and Chhavi Sharma. "Community detection in social networks." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 6.3* (2016): 115-135.

[2] Cuvelier, Etienne, and Marie-AudeAufaure. "Graph mining and communities detection." *Business Intelligence.Springer Berlin Heidelberg,* 2012.117-138.

[3] Adedoyin-Olowe, Mariam, Mohamed MedhatGaber, and Frederic Stahl. "A survey of data mining techniques for social media analysis." *arXiv preprint arXiv:*1312.4617 (2013).

[4] Afsarmanesh, Nazanin, and MatteoMagnani. "Finding overlapping communities in multiplex networks." *arXiv preprint arXiv:*1602.03746 (2016).

[5] Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu. Social media mining: an introduction. *Cambridge University Press*, 2014.

[6] McCreesh, Ciaran, et al. "Clique and constraint models for maximum common (connected) subgraph problems." *International Conference on Principles and Practice of Constraint Programming.Springer International Publishing,* 2016.

[7] Reid, Fergal, Aaron McDaid, and Neil Hurley. "Percolation computation in complex networks." *Advances in Social Networks Analysis and Mining (ASONAM),* 2012 *IEEE/ACM International Conference on. IEEE*, 2012.

Volume: 09 Issue: 02 Pages: 3362-3375 (2017) ISSN: 0975-0290

[8] Palla, Gergely, et al. "k-clique Percolation and Clustering." Handbook of Large-Scale Random Networks.*Springer Berlin Heidelberg*, 2008.369-408.

[9] Wang, Jianxin, et al. "Identifying protein complexes from interaction networks based on clique percolation and distance restriction." *BMC genomics 11.2* (2010): S10.

[10] McDaid, Aaron, and Neil Hurley. "Detecting highly overlapping communities with model-based overlapping seed expansion." *Advances in Social Networks Analysis and Mining (ASONAM)*, 2010 *International Conference on. IEEE*, 2010.

[11] Zachary, Wayne W. "An information flow model for conflict and fission in small groups." *Journal of anthropological research 33.4* (1977): 452-473.

[12] Palla, Gergely, et al. "Uncovering the overlapping community structure of complex networks in nature and society." *arXiv preprint physics*/0506133 (2005).

[13] Nandi, G., and A. Das. "A survey on using data mining techniques for online social network analysis." *Int. J. Comput. Sci. Issues (IJCSI) 10.6* (2013): 162-167.

[14] Pattabiraman, Bharath, et al. "Fast Algorithms for the Maximum Clique Problem on Massive Sparse Graphs." *WAW*. 2013.

[15] Palsetia, Diana, et al. "Clique guided community detection." Big Data (Big Data), 2014 *IEEE International Conference on.IEEE*, 2014.

[16] Leskovec, Jure, Kevin J. Lang, and Michael Mahoney. "Empirical comparison of algorithms for network community detection." *Proceedings of the 19th international conference on World wide web. ACM,* 2010.

[17] Yang, Jaewon, Julian McAuley, and Jure Leskovec. "Community detection in networks with node attributes." *Data Mining (ICDM), 2013 IEEE 13th international conference on.IEEE,* 2013.

[18] Harenberg, Steve, et al. "Community detection in large-scale networks: a survey and empirical evaluation." *Wiley Interdisciplinary Reviews: Computational Statistics 6.6* (2014): 426-439.

[19] Lancichinetti, Andrea, Santo Fortunato, and JánosKertész. "Detecting the overlapping and hierarchical community structure in complex networks." *New Journal of Physics 11.3* (2009): 033015.

[20] Du, Nan, et al. "Community detection in large-scale social networks." *Proceedings of the 9th WebKDD and 1st SNA-KDD* 2007 *workshop on Web mining and social network analysis.ACM*, 2007.

[21] Shen, Huawei, et al. "Detect overlapping and hierarchical community structure in networks." *Physica A: Statistical Mechanics and its Applications* 388.8 (2009): 1706-1712.

[22] Evans, T. S., and Renaud Lambiotte. "Line graphs, link partitions, and overlapping communities." *Physical Review E 80.1* (2009): 016105.

[23] Evans, Tim S., and Renaud Lambiotte. "Line graphs of weighted networks for overlapping communities." *The European Physical Journal B-Condensed Matter and Complex Systems* 77.2 (2010): 265-272.

[24] Evans, Tim S. "Clique graphs and overlapping communities." Journal of Statistical Mechanics: *Theory and Experiment 2010*.12 (2010): P12037.

[25] Lee, Conrad, et al. "Detecting highly overlapping community structure by greedy clique expansion." *arXiv preprint arXiv*:1002.1827 (2010).

[26] Gregory, Steve. "A fast algorithm to find overlapping communities in networks." *Machine learning and knowledge discovery in databases* (2008): 408-423.

[27] Gregory, Steve. "Finding overlapping communities using disjoint community detection algorithms." *Complex networks* (2009): 47-61.

[28] Adamcsek, Balázs, et al. "CFinder: locating cliques and overlapping modules in biological networks." *Bioinformatics 22.8* (2006): 1021-10