Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# DETECTION OF COMMUNITIES IN COMPLEX GRAPHS BASED ON THE NORMALIZED COVARIANCE BETWEEN VERTICES

[1]ABDERRAHIM GHADI, [1]MOHAMED BEN AHMED, [1]A.ANOUAR BOUDHIR, [1]MOHAMED BOUHORMA
[1]LIST laboratory, Abdelmalek Essaadi University
[1]Computer Sciences department, Faculty of Sciences and Techniques of Tangier - Morocco
E-mail: ghadi05@gmail.com, benahmed@fstt.ac.ma, hakim.anouar@gmail.com, m.bouhorma@fstt.ac.ma

**ABSTRACT**

Communities detection in graphs has been the subject of many algorithms. Recent methods seek to optimize a function of modularity indicating a maximum of relationships within communities and a minimum of inter-communities relations.
This paper will be presented as follows, first, we present the state of the art in matters of detection methods of communities then we propose a method for detecting communities. This method works into two steps, the first step consist to split the graph into subgraphs by using the normalized covariance measurement and the second part allows to merge the sub-graphs by maximizing the modularity of the resulting graph.

**Keywords:** *Graph Theory, Complex Networks, Modularity, Clustering Coefficient, Assortativity.*

## 1. INTRODUCTION

Recent advances in the field of complex systems have highlighted the central role that the graphs play in many phenomena. These large graphs are used to model the interactions between the different actors of these complex phenomena involved in very many fields: computer science, sociology, biology, physics, etc. , Surprisingly these graphs have nontrivial common structural properties which have been the subject of many recent studies [1, 6, 7, 12, 21].
These properties allow to consider a new algorithmic for large graphs.

The work of this paper are part of this interdisciplinary context, focusing on the algorithmic method for communities detection, ie of the vertices of the graph can be easily grouped into sets of vertices such that each set of vertices is densely connected internally and weakly linked externally.

We will first present the large graphs (complex graphs or complex networks). Then we introduce the communities detection problem. A several works related to this subject will be mentioned, implemented in parallel with the work carried out in this paper.

## 2. TERMINOLOGIES

Graph theory provides a modeling support complex networks by generalizing the structure whatever their origin :

1. an element of the network (individual computer, proteins, .etc) is represented by a vertex or node of the graph;
2. a relationship or connection between two elements is represented by an edge of the graph.

This modeling allows to express the distinctive properties of complex networks, and to implement algorithms to solve the problems these ones raise.

**Graph**: A graph G is a discrete structure consisting of nodes (vertices) and lines joining the nodes (edges). Two vertices are adjacent to each other if they are joint by an edge. The edge joining the two vertices is said to be an edge incident with them.

We use $V_G$ (n = | V | is the order of the graph) and $E_G$ (m = | E |) is the size of the graph) to denote the set of vertices and edges of G respectively.

**Degrees of Vertices**: The degree of a vertex is the number of edges incident with it, except that

Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex u is denoted by d(u).

**Subgraphs**: A subgraph of a graph G is a graph H where $V_G \subseteq V_H$ and $E_G \subseteq E_H$

**Clique**: a complete subgraph of a given graph is called a clique of the graph.

**Density**:The density of a graph is defined as either $\frac{2m}{n(n-1)}$ : The ratio between the number of edges and the maximum possible number of edges given the number of nodes of the graph.

**Adjacency Matrix**: Let G be an graph. Suppose $u_1, u_2, \ldots, u_n$ are the vertices of G. Then the adjacent matrix with respect to this ordering of $V_G$ is the $nxn$ matrix $M=[m_{ij}]$ where

$$m_{ij} = \begin{cases} 1 & \text{when } u_i \text{ and } u_j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases}$$

**Paths and Cycles**: In a graph G, a path of length l from u to v is a sequence of l + 1 vertices $v_1, v_2, \ldots, v_l, v_{l+1}$ where $v_1 = u$ and $v_{l+1} = v$ , and $v_i v_j \in E_G$ .

A path is called a circuit when $v=u$

A path or a circuit is simple if it does not contain the same edge more than once.

**Connected Graphs**: A graph is connected if there is a path between every pair of distinct vertices of the graph. An edge $uv$ in a connected graph $G$ is called a bridge if $G-uv$ , the graph obtained by deleting uv from G, is not connected.

**distance**: The distance between two vertex is the length of the shortest path connecting them.

**Edge Betweenness Centrality**: Betweenness centrality of an edge e is the sum of the fraction of all-pairs shortest paths that pass through $e$ :

$$C_B(e) = \sum_{s,t \in V} \frac{\sigma(s,t \mid e)}{\sigma(s,t)}$$

where $V$ is the set of nodes, $\sigma(s,t)$ is the number of shortest $(s,t)$ -paths, and $\sigma(s,t \mid e)$ is the number of those paths passing through edge $e$ .

$$Q = \frac{1}{2m} * \sum_{i,j} (m_{ij} - \frac{d(u_i)d(u_j)}{2m}) * \delta(u_i, u_j)$$

$$i = 1 \ldots n \text{ and } j = 1 \ldots n$$

**Modularity** $Q$ : Modularity represents the difference between the value of adjacency between two nodes of the same community $m_{ij}$ and the probability that those are connected. $Q$ modularity of a graph is defined by:

where m =| E | and n =| V |, $m_{ij}$ is equal to 1 if $u_i$ and $u_j$ are adjacent, $m_{ij} = 0$ otherwise. $d(v_i)$ is the degree of the vertex ui and $\delta$ is the Kronecker symbol equal to 1 if $u_i$ et $u_j$ belong to the same community and 0 otherwise.

## 3. SIMILARITY & CLUSTERING

In Mathematics and computing, similarity is an important criterion for the identification of subgroup in a group of objects, values (numerical or not), data (known or recognized) in a ”space” or a system In classification, it is called clustering (or clustering) to describe data partitioning, and a cluster is then a set of data or materials with similarities.

There are several detection communities methods. We will present here those receiving the most interest from the scientific community:

1. Hierarchical methods
2. Methods based on optimization of an objective function
3. Methods based on a model

Before presenting these three methods, we will clarify the relationship between the problem of community detection and graph partitioning. Indeed, the graph partitioning is to group the nodes of a graph in a generally predetermined number of homogeneous subgraphs by minimizing the number of links between the different groups while the community detection does the same with or without requiring a priori knowledge of the number of communities.

### A. hierarchical methods

Hierarchical classification algorithms seek to consolidate the nodes of a graph (network) in different communities, so that the nodes of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

same community as similar as possible so that nodes from different communities are the different possible. For hierarchical methods, we distinguish two distinct approaches namely the agglomerative algorithms and the divisive algorithms:

1) Agglomerative algorithms:

- The agglomerative approach starts with a structure in which each node of the graph represents a
- community. We start with n communities (where n =| G |).
- We begin by calculating the distances between communities and merge the two closest communities to form a new community.
- At each step, we recalculate all distances between communities and we fuse two communities.
- When there is only one graph representing the entire community, there is more distance to calculate.
- the different stages of this process can be represented by a tree-form called dendrogram. The leaves are the communities with a single node and the root represents the entire graph.

Walktrap is an example of agglomerative algorithm.

2) divisive algorithms: The algorithms for division are to divide a network into communities by iteratively eliminating the links between nodes. they start with a single community until to obtain n communities with a single node representing the leaves of the dendrogram.
In each iteration, all connected graph is considered a community. Existing methods differ in the choice of links to be removed and by the weight given to the links.
Edge Betweenness is an example of divisive algorithm.
To find the inter-community links, Edge Betweenness algorithm gives each link a measure of betweenness centrality (Edge Betweenness Centrality).

**B. Methods based on optimization of an objective function**

There are a number of algorithms based on heuristics to define the community structure of networks. This type of algorithm consist to define an objective function whose value varies according to the identified communities. The function is maximum for the best community structure. An example of this type of algorithm is Fast Greedy Newman.

**C. Methods based on a model**

The algorithms based on the model are unsupervised classification algorithms, using methods based on prototypes expressed in a model formalism. For each type of data, a learning model adapted to the nature of the data is proposed.

Label Propagation is an example of methods based on a model. This algorithm is based on the principle that each node change the community according to the community containing its neighbors. A node is part of the community that contains the largest number of neighboring nodes.

This process is the learning model for Label Propagation, which runs on all nodes in each iteration. At the beginning of the algorithm, each node is in one community. Then, the nodes change their communities while respecting the learning model. With this method, a group of nodes, strongly linked, ends up in the same community.

Fast Greedy is not a hierarchical algorithm. It gives no score or setting links. Add to that it does not measure the distance or similarity between communities to merge. Similarly, the major drawback of this method is its classification quality that is less good compared with other algorithms using modularity eg Edge Betweenness.

Label Propagation is the only one to be non-deterministic because its execution with the same network can give several results. When the number of inter-community links is low, all these algorithms end up having the same communities structure. By increasing the ratio of links inter-communities, community detection results will differ, and the quality of the clustering degrades.
In summary, the most popular algorithms for communities detection are:

- Edge betweenness (Girvan-Newman link centrality-based approach),
- Walktrap (Pons-Latapy random walk-based approach),
- Leading Eigenvectors (Newman's spectral approach),
- Fast Greedy (Clauset et. al modularity optimization),
- Label Propagation (Raghavan et. al),

Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Louvain (Blondel et. al, modularity optimization),
- Spinglass (Reichardt-Bornholdt, modularity optimization),
- InfoMap(Rosvall-Bergstrom, compression-based approach).

## 4. PROPOSED ALGORITHM

The covariance is a meters to assess the direction of change of two random variables and qualify the degree of independence. So if the bigger (respectively smaller) values a variable correspond mainly to large (respectively smaller) values of the other variable, then the covariance is highthe two variables tend to show similar behavior. Otherwise,

when the superior values of a variable primarily correspond to inferior values of the other, the variables tend to show opposite behavior, and the covariance is low (The covariance values are between -1 and 1).

The higher the covariance will be close to zero more can say that the two sets of variables are independent. the contrary, the more this one will be high the more link Independence will be weak.

We chose to use this measure of covariance on the links connecting the vertices of the graph(network). Indeed, if there is a connection between two vertices with a large number of common neighbors, then this link has a high covariance and represents intra-community link. Otherwise, if a connecting two vertices with a small number of common neighbors, then this link has a low covariance. This is called an inter-community link.

To calculate the covariance links, we must first calculate the adjacency matrix of the graph (network). If two vertices $u_i$ and $u_j$ of this graph are directly connected, $M_{ij}$ is equal to one and is zero otherwise.

The covariance of a link $e_{ij}$ connecting nodes $u_i$ andis high if the ones and zeros of the vector $M_i$ (the ith of the adjacency matrix) correspond to ones and zeros of vector $M_j$ . in other words, the nodes $u_i$ and $u_j$ have the same neighbors and are probably in the same community).

The covariance is low if the ones and zeros of $M_i$ correspond to the zeros and ones of $M_j$ . In the latter case, the nodes $u_i$ and $u_j$ have not the same neighbors and they are therefore in

$$cov(e_{ij}) = \frac{1}{n-1}\sum_{l=1}^{n}(M_{il} - \bar{M}_i)(M_{jl} - \bar{M}_j)$$

different communities.

The covariance of a edge $e_{ij}$ is define as follow :

Where $\bar{M}_i$ is the average of the vector $M_i$ and n is the graph.

The sign of the covariance therefore shows the tendency in the linear relationship between the variables. The magnitude of the covariance is not easy to interpret. The normalized version of the covariance, the correlation coefficient, however, shows by its magnitude the strength of the linear

$$cov_N(e_{ij}) = \frac{\sum_{l=1}^{n}(M_{il} - \bar{M}_i)(M_{jl} - \bar{M}_j)}{(n-1)\sigma_{M_i}\sigma_{M_j}}$$

relation. For this reason we will use the normalized form of the covariance:

$\sigma_{M_i} = \frac{1}{n-1}\sum_{l=1}^{n}(m_{il} - \bar{M}_i)^2$ Where is the standard deviation of $M_i$ (with Bessel's correction).

After several experiments, we used the covariance as a measure of similarity between two vertices and thus links. Indeed, this measure has given good results and good performance.

Interclass inertia: This measure is used to study a set of data, the grouping into several subsets so that the elements of the same group (community) are as similar and that two distinct groups are as heterogeneous as possible. We use this measure to determine the link aggregation: links with high covariance and links with low covariance.

The inter-class inertia between C1 and C2 communities is defined as follows:

Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$I_{C_1,C_2} = | C_1 | (\mu_1 - \mu)^2 + | C_2 | (\mu_2 - \mu)^2$$

- | $v_i v_j \square E_G$ | and | $^1$ | are respectively the number of links in both $C_1$ and $C_2$ communities.
- $\mu_1$ , $\mu_2$ and $\mu$ are, respectively, the average covariance for $C_1$ , $C_2$ and all communities.

The main idea of the algorithm is to remove a set of links in each iteration. This process of elimination is repeated until a number of sub networks greater than $\xi = \sqrt{}$ . $\xi$ number reflects a stop test for the algorithm. Indeed, Newman proved that modularity is maximal when the number of communities is close to $\xi$ .

At the beginning of the algorithm, the number of edges is | $E$ |. The number of subgraphs is equal to one. After, the algorithm at each iteration eliminates a set of edges, and calculates the number of subgraphs. If this number is less than $\xi$ , it still eliminates other edges. Otherwise, the algorithm stops and returns the graph $G=(V,E)$. The algorithm begins by calculating the covariance of all the edges and put them in $\Gamma_{cov_N}$ . Then the vector $\Gamma_{cov_N}$ is sorted in descending order. Then, the algorithm calculates different values of the inertia inter-class $I_{ic}$ . The index of the maximum value of $I_{ic}$ represents the number of edges not to eliminate. So the algorithm will remove | $E$ | $- I_{ic} + 1$ edges that have the lowest covariance values.

Creation communities using covariance and inter-class inertia:

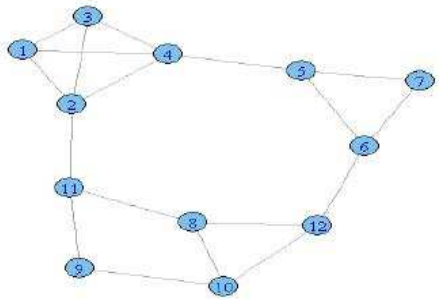
**INPUT**: $G(V,E)$
**OUTPUT**: $G'(V,E')$
**VARIABLES**: $\zeta_{sg}$: Number of subgraphs in $G'$
$\xi$: Minimal number of sub-graph to identified
$\Gamma_{cov_N}$:Vector of covariances of edges
$I_{ic}$: Vector of inter-class inertia
$I_s$: Number of edges to kept
$d(i)$ The vector of degree of $M_i$
$\Delta$ is the Matrix of Kronecker symbol
$((\delta(v_i,v_j))_{v_i,v_j \in V})$

**BIGIN**
$E = E'$
$\zeta_{sg} = 1$
$\xi = sqrt(n)$
WHILE $\zeta_{sg} < \xi$ DO
sort $\Gamma_{cov_N}$ in descending order
FOR $k = 1$ TO $| E' |$ DO
$I_{ic} \leftarrow Inertia(\Gamma_{cov_N}[1 \ldots k], \Gamma_{cov_N}[k+1 \ldots size(\Gamma_{cov_N})])$
ENDFOR
$I_s \leftarrow MAX(I_{ic})$
FOR each edge $e_{ij}$ in $E'$ DO
IF $cov_N(e_{ij}) \in \Gamma_{cov_N}[I_s \cdots | E' |]$ THEN
Delete $e_{ij}$ in $E'$ and $E' = E' - e_{ij}$
ENDFOR
$G' \leftarrow (V,E')$
$\zeta_{sg} \leftarrow subgraph(G')$
RETURN $G' = (V,E')$
**END**

### A. Illustrative example

Let the graph of Fig.1 composed of $n = | V | = 12$ vertices and | $E$ | = 17 edges:

Fig. 1.




Graph initial G (V , E )

The adjacency matrix of graph (Fig.1) is:

:)/

$2^* | G_t | ) * \Delta(j,k)$;
ENDFOR
ENDFOR
$m_2 \leftarrow \frac{m_2}{2*|G_t|}$
IF $(m_1 < m_2)$ THEN
Merge$(G_i, G_j, G)$ ;
ENDIF
ENDIF
ENDFOR
ENDFOR
retourn $\{G_1, G_2, G_3, \ldots, G_{\zeta_{sg}}\}$ ;
ENDWHILE
**END**

Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$M=\begin{vmatrix} 0&1&1&0&0&0&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0&1&0\\ 1&1&0&1&0&0&0&0&0&0&0&0\\ 0&1&1&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&1&1&0&0&0&0&0\\ 0&0&0&0&1&0&1&0&0&0&0&1\\ 0&0&0&0&1&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&1&1\\ 0&0&0&0&0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&1&1&0&0&1\\ 0&1&0&0&0&0&1&1&0&0&0&0\\ 0&0&0&0&0&1&0&1&0&1&0&0 \end{vmatrix}$$

Below the table summarizing the first iteration of the algorithm:

| Edges | Covariances | Normalised Co-variance in [0,1] | Interclass Inertia | Index |
|---|---|---|---|---|
| $e_{2,4}$ | 0,14 | 1,00 | 0,37 | 1 |
| $e_{2,3}$ | 0,09 | 0,78 | 0,54 | 2 |
| $e_{1,3}$ | 0,05 | 0,56 | 0,50 | 3 |
| $e_{2,11}$ | 0,05 | 0,56 | 0,52 | 4 |
| $e_{1,2}$ | 0,03 | 0,48 | 0,51 | 5 |
| $e_{3,4}$ | 0,02 | 0,44 | 0,49 | 6 |
| $e_{5,6}$ | 0,02 | 0,44 | 0,48 | 7 |
| $e_{5,7}$ | 0,02 | 0,44 | 0,50 | 8 |
| $e_{6,7}$ | 0,02 | 0,44 | 0,52 | 9 |
| $e_{8,10}$ | 0,02 | 0,44 | 0,56 | 10 |
| $e_{8,12}$ | 0,02 | 0,44 | 0,63 | 11 |
| $e_{10,12}$ | 0,02 | 0,44 | **0,72** | **12** |
| $e_{9,10}$ | -0,02 | 0,22 | 0,65 | 13 |
| $e_{9,11}$ | -0,02 | 0,22 | 0,61 | 14 |
| $e_{4,5}$ | -0,07 | 0,00 | 0,38 | 15 |
| $e_{6,12}$ | -0,07 | 0,00 | 0,18 | 16 |
| $e_{8,11}$ | -0,07 | 0,00 | 0,00 | 17 |

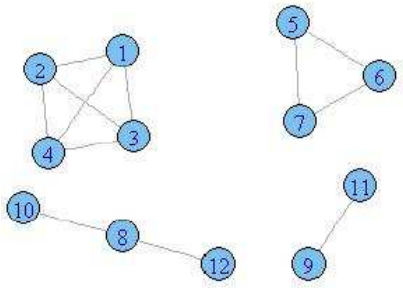
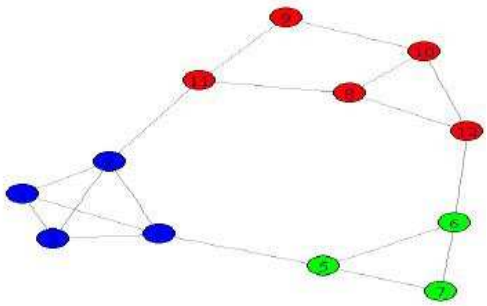



*Fig. 2. The Structure Of The Graph After The First Iteration Of The First Part Of The Algorithm.*

At the end of the firth iteration, the number of subgraphs $\zeta_{sg}$ is four and it exceeds $\zeta$ which is 3.46 ( $\sqrt{12}$ ). So this is the end of the first part of the algorithm whose output is the graph $G=(V,E')$ with $V$ representing twelve nodes of the graph and $E'$ containing eleven edges that have not been eliminated.

The resulting graph is shown in Fig.2 Before starting the second part of the algorithm ie the merging of sub graphs found, we must have a number of subgraphs greater than one. In our example, the end of the first part of the algorithm returned a set of 4 sub-graphs $G_1$ , $G_2$ , $G_3$ and $G_4$ representing the detected communities and the modularity of $G$ is $0,30$ . So we will calculate the $Q$ modularity for possible combinations (here 6) and we will hold the maximum values of modularity.



During the first iteration of the second part of the algorithm, the number of sub-graphs is equal to $4$ . we calculated the modularity $Q$ for the possible combinations. A maximum value of $Q$ equal to $0,35$ is obtained by merging the subgraph that contains the vertices $8,10$ and $12$ with the sub-graph that contains vertices $9$ and 11. The merger of these two sub -graph consist to add all links between nodes of these two subnets that are in the initial graph $G$ . the modularity of the graph in Fig.2 is $0,30$ while the modularity of the graph in Fig.3 is $0,35$ . In this case, we keep the structure of the graph in Fig.3 because its modularity is much superior. in the end of the algorithm, the final structure of our graph will be that of Fig.3.

*Fig. 3. The Structure Of The Graph After The First Iteration Of The Second Part Of The Algorithm.*

Example of calcul of modularity $(i=1,j=2)$

Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$M=\begin{vmatrix}
0&1&1&0&0&0&0&0&0&0&0&0\\
1&0&1&1&0&0&0&0&0&0&1&0\\
1&1&0&1&0&0&0&0&0&0&0&0\\
0&1&1&0&1&0&0&0&0&0&0&0\\
0&0&0&1&0&1&1&0&0&0&0&0\\
0&0&0&0&1&0&1&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0
\end{vmatrix}$$

The vector of degree is :

$$d=\begin{vmatrix}2&4&3&3&3&2&2&2&0&1&1&1\end{vmatrix}$$

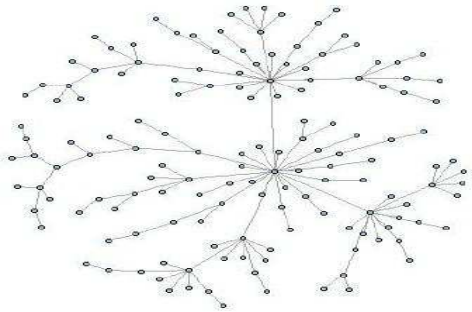
The matrix $\Delta(\delta(v_i,v_j))$ is:

$$\Delta=\begin{vmatrix}
1&1&1&1&1&1&1&0&0&0&1&0\\
1&1&1&1&1&1&1&0&0&0&1&0\\
1&1&1&1&1&1&1&0&0&0&1&0\\
1&1&1&1&1&1&1&0&0&0&1&0\\
1&1&1&1&1&1&1&0&0&0&1&0\\
1&1&1&1&1&1&1&0&0&0&1&0\\
1&1&1&1&1&1&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&1&0&1\\
0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&1&0&1\\
1&1&1&1&1&1&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&1&0&1
\end{vmatrix}$$

So modularity value in iteration $(i=1,j=2)$ is $Q=0,30$

## 5. SIMULATION AND RESULTS

### A. Simulation 1



Let Barabasi graph with $150$ vertices :

Fig. 4. Barabasi graph with $150$ vertices.

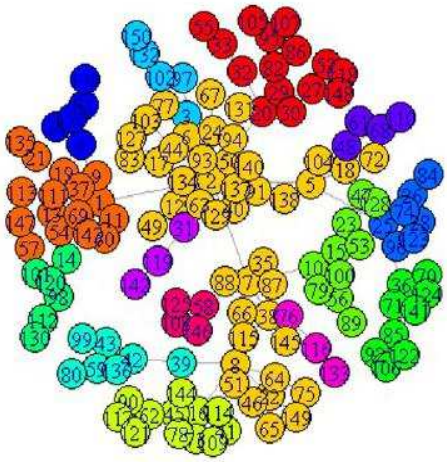


Fig. 5. Barabasi graph with $150$ vertices

The following table summarizes the community structure of the graph Fig.4 calculated with DCOMCOV algorithm and compared with five other algorithm.

| Algorithms \ Results | Number of communities | Modularity |
|---|---|---|
| Fast greedy | 11 | 0.8203234 |
| Edge betweenness | 11 | 0.8203234 |
| Spinglass | 15 | 0.8070808 |
| Dcomcov | 15 | 0.7804124 |
| Label propagation | 18 | 0.7770821 |
| Walktrap | 15 | 0.7561146 |

### B. Simulation 2

Let Watts graph with $500$ vertices :

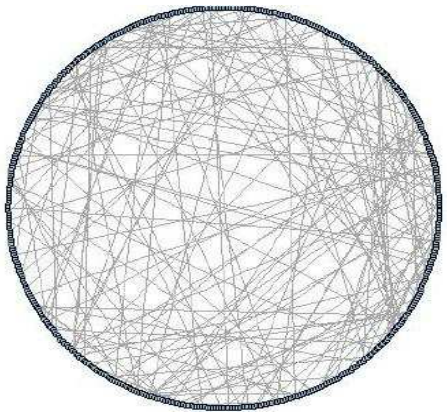


Fig. 6. Watts graph with $500$ vertices.

Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

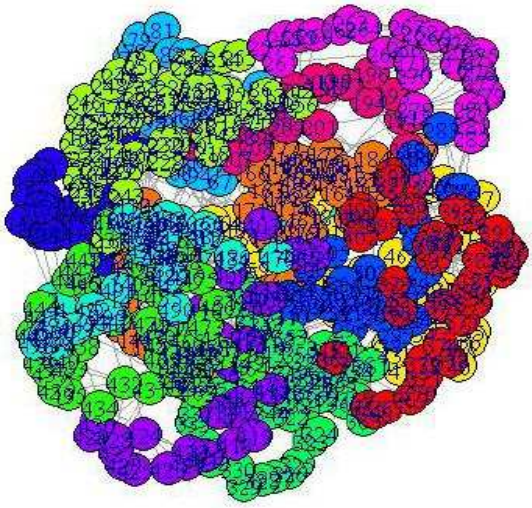


Fig. 7. Watts graph with 500 vertices.

The following table summarizes the community structure of the graph Fig4 calculated with DCOMCOV algorithm and compared with five other algorithm.

| Algorithms \ Results | Number of communities | Modularity |
|---|---|---|
| Spinglass | 18 | 0.7930404 |
| Walktrap | 13 | 0.7864391 |
| Edge betweenness | 14 | 0.784115 |
| Dcomcov | 21 | 0.7705713 |
| Label propagation | 39 | 0.7279691 |
| Fast greedy | 7 | 0.7273706 |

### C. Simulation 3

Let Lattice graph with 64 vertices :

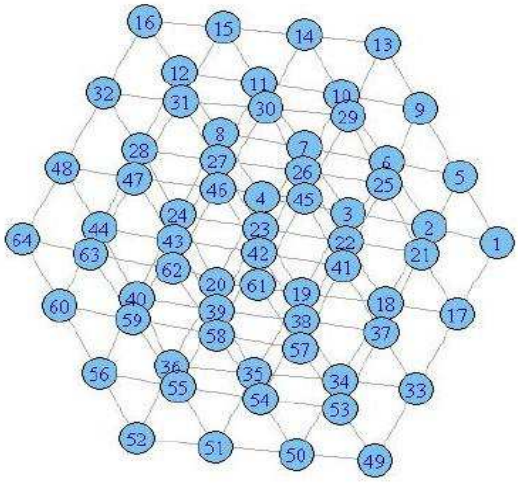


Fig. 8. lattice with 64 vertices

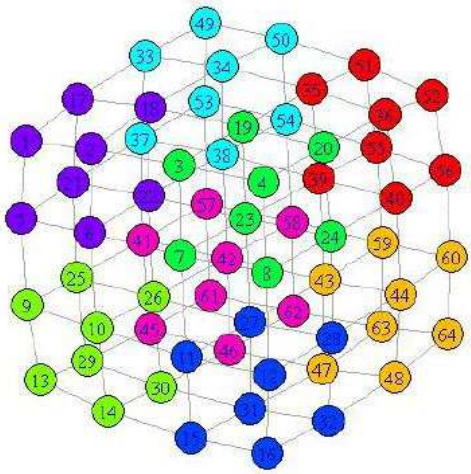


Fig. 9. Communities of 64-Lattice graph.

| Algorithms \ Results | Number of communities | Modularity |
|---|---|---|
| Walktrap | 8 | 0.5416667 |
| Edge betweenness | 8 | 0.5416667 |
| Spinglass | 6 | 0.5416655 |
| Dcomcov | 8 | 0.5254163 |
| Fast greedy | 6 | 0.5073061 |
| Label propagation | 6 | 0.4537037 |

### D. Simulation 4

Let Karate graph with 44 vertices :

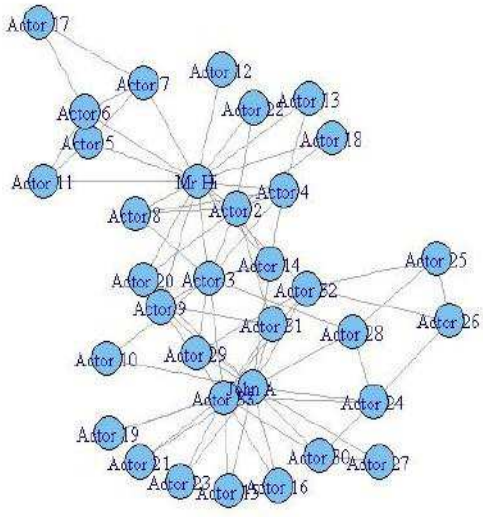
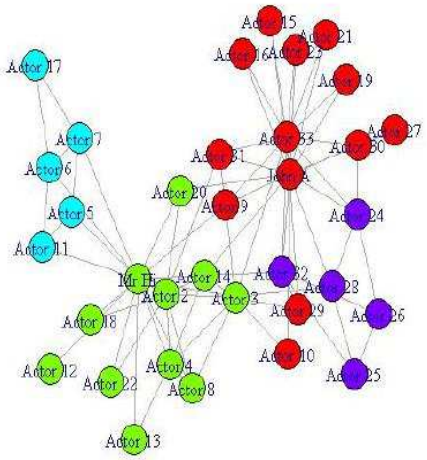


Fig. 10 : 44 -karate graph.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



| Algorithms \ Results | Number of communities | Modularity |
|---|---|---|
| Walktrap | 4 | 0.440181 |
| Fast greedy | 3 | 0.4345215 |
| Edge betweenness | 6 | 0.345299 |
| Dcomcov | 4 | 0.300155 |
| Spinglass | 4 | 0.2164971 |
| Label propagation | 2 | 0.08503401 |

Fig. 11. Communities of 44 - karate graph.

## 6. CONCLUSION

We have presented a novel method for detecting communities in complex networks (in the form of a graph). The algorithm proposed in this method is divided in two parts.

In the first part, we used the normalized covariance measure to eliminate one or more inter-community edges and split the initial graph into sub-graphs. The second part is to find the optimal structure of communities keeping mergers between sub-graphs. In the last part, we used the concept of modularity, which is a measure of the quality of a partitioning vertices's graph into communities. The main idea, here, is that a good partitioning graph involves a number of significant intra-communities edges and a small number of inter-communities edges.

After several simulations on large graphs, we firstly constate, that the number of communities detected with the proposed algorithm is the average of the numbers found by the other fives methods. Secondly, the modularity is also the average modularities found by the cited methods. In add to the robust of our algorithm approach in their simplicity, we can also appreciate the considerable number of deleted links at each iteration conducting to a faster convergence.

This algorithm is applicable on undirected and unweighted graphs. Due to this, a perspective of this work open a new reflexions and discussions to improve this method in order to resolve the detection communities in weighted graphs.

## REFERENCES

[1] Barber M., "Modularity and community detection in bipartite networks", Physical Review E,vol. 76, n6, p. 1-9, 2007.

[2] Barber M. J., Clark J. W., " Detecting network communities by propagation g labels under constraints ",Physical Review E – Statistical, Nonlinear and Soft Matter Physics , vol. 80, n 2 Pt 2, p. 026129, 2009.

[3] Blondel V. D., Guillaume J.-L., Lambiotte R., Lefebvre E., " Fast unfolding of communities in large networks ",Journal of Statistical Mechanics : Theory and Experiment , vol. 2008,n10, p. P10008, October, 2008.

[4] Chen W., Liu Z., Sun X., Wang Y., " Community detection in social network s through community formation games ",roceedings of the Twenty-Second international joint conference on Artificial Intelligence - Vol. Three , IJCAI11, AAAI Press, p. 2576-2581, 2011.

[5] Crampes M., Planti M.,"Détection de communauts dans les graphes bipartis" ",IC 2012 Ingnierie des connaissance, 2012.

[6] Crampes M., Planti M., " Partition et recouvrement de communauts dans les graphes bipartis,unipartis et orients ", IC 2013 Ingnierie des connaissance , 2013.

[7] Davis G. B., Carley K. M., " Clearing the FOG : Fuzzy, overlapping groups for social networks ",Social Networks , vol. 30, n3, p. 201-212, 2008.

[8] Evans T. S., Lambiotte R., " Line Graphs, Link Partitions Overlapping Communities "Physical Review E, vol. 80, n1, p. 9, 2009.

[9] Fortunato S., " Community detection in graphs "Physics Reports,486, n3-5, p. 103, June, 2009.

[10] Freeman L. C., " Finding social groups : A meta-analysis of the southern women data ",Dynamic Social Network Modeling and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Analysis, National Academies Press, p. 39-97, 2003.
[11] Gregory S., " Finding overlapping communities in networks bypropagation "New Journal of Physics , vol. 12, n10, p. 103018, 2009.
[12] Stanley Milgram. The small world problem. Psychology Today, 1967
[13] Paul Erdôs and Alfred Rény. On random graphs. Publicationes Mathematicae, 6:290297, 1959.
[14] B la Bollobas and Oliver Riordan. Slow emergence of the component in the growing m-out graph. 2004.
[15] Duncan J. Watts and Steven H. Strogatz. Collective dynamics"small-world" networks. NATURE, Vol 393 :440442, 1998.
[16] Jon M. Kleinberg. Navigation in a small world. NATURE, Vol:845, 2000.
[17] Emmanuelle Lebhar. Algorithmes de routage et modèles alatoires pour les graphes petits mondes. PhD thesis, Ecole Normale Supérieure Lyon, 2005.
[18] U .Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17(4):395416, 2007.
[19] Guo-Qing Zhang et al."A universal assortativity measure for network analysis". APS Journals, 28 Dec 2012.
[20] G.Pandurangan. "Building Low-Diameter Peer-to-Peer Networks" JOURNAL ON SELECTED AREAS IN COMMUNICATIONS,21, NO. 6, AUGUST 2003.