Abstract

There are many community detection algorithms nowadays, but no hiding community algorithms. If communities are not protected, hackers can easily focus on the important community and malfunction the networks. Therefore, this thesis focuses on solving the problem. We devised six hiding approaches to hide the community in undirected networks. Two of them take lower degree into account to add or remove edges. Other methods add edges randomly. Since there are various networks, we selected two types of networks: real network and random networks as testing data in order to analyse the relationship between types of network and hiding results. Furthermore, we use two community detection algorithms, OSLOM and infomap to examine if different community detection algorithms have an effect on hiding results.

- The project developed six methods to hide community structure in undirected networks, see pages 11-20.
- There is only one method that removes edges using modified BFS algorithms
- There are some limitations to hiding methods, such as no use in homeless nodes, see pages 11-20.
- This project did experiments on the relationship between hiding results and hiding methods, types of network: real network and random network, and different community detection algorithms: infomap and OSLOM, see pages 20-49.
- The hiding results are quite different in real networks and random networks, see pages 48-49.
- The performance of hiding methods gets worse when the network becomes bigger, see pages 48-49.
- Different community detection algorithms impact hiding results greatly, see pages 48-49.
- There is no method that is the best to all size of network in our hiding approaches, see pages 48-49.

Contents

I. Introduction

There is a variety type of networks in the world. According to different topics, network can be divided into social network, biological network such as food web, metabolic network, or technological network such as WWW (Girvan and Newman, 2002). One of the most interesting properties of networks is that they are composed of multiple communities, and a community is consisted of many vertices. There are edges connected these vertices. Some connected neighbouring or relevant communities can be grouped together to form an independent cluster. With few connected edges between multiple clusters, a network will be generated. (See Fig1)The connectivity inside a community is very dense but sparse between clusters. Besides, communities can be subdivided into groups of groups. This is called hierarchical structure. (Clauset, Moore, and Newman, 2008). (See Fig2)

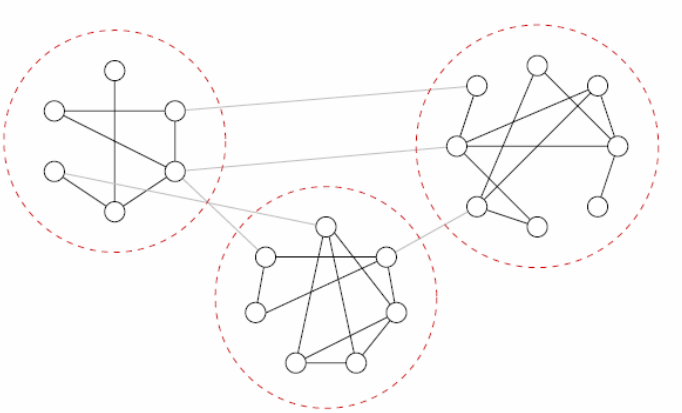


Figure1. Community Structure (Newman and Girvan, 2004, p.1)

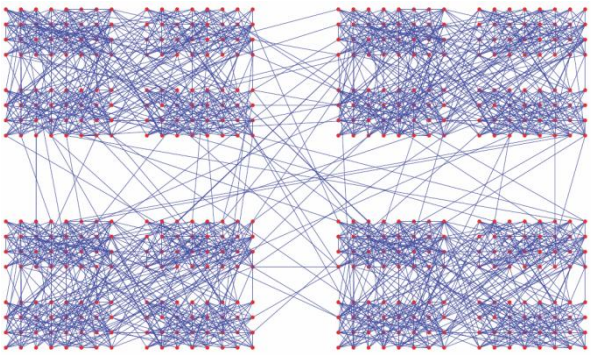


Figure2. Hierarchical structure (Lancichinetti, Fortunato, and Kert´esz, 2009, p.3)

Vertices or communities which are grouped in the same clusters often share the same common properties. They have more interaction with each other than other clusters. For instance, in biological network, proteins in the cells are clustered together with the same function. In the citation network, pages are in the same group because they have the related topic (Girvan and Newman, 2002). In social networks, bar associations are a set of people they are all lawyers.

In networks, communities have become more important because they can improve the efficiency and contribute to the better performance. For example, in social networks, if customers are classified based on their interests in online bookstores, websites can suggest them the proper books that they prefer. It is helpful to achieve the sales target rather than recommend everything to customers (Fortunato, 2010). Therefore, how to divide networks into communities or how to detect communities in networks has become a popular research topic.

Nowadays, an increasing number of community detection algorithms have been developed. There is no algorithm that can implement in all kinds of networks and also has the best performance. They are suitable for different size and the density of networks. Furthermore, according to different ideas and different aims of each algorithm, the use and the results are quite distinct. For example, modularity maximization is a popular technique for community detection algorithms. Each community usually has the property of high modularity which is one of possible ways to indicate the good quality of community. However, there are some critical problems based on the size of networks in this method, such as merging the small clusters, or breaking the large clusters. These lead to incorrect and inaccurate detection results (Lancichinetti, and Fortunato, 2011). The other example is called CONGA (Cluster-Overlap Newman Girvan Algorithm) which is an algorithm that extend Girvan–Newman algorithm. It has the ability to find overlapping communities through using the new idea of split betweenness to split the vertices between communities. However, although the result is good, the running time is so large because it inherits the attribute of Girvan-Newman algorithm. Therefore, this algorithm is appropriate to small networks (Gregory, 2008). In brief, through applying different community detection algorithms, we can partition different types of networks into the communities with the different properties.

Since we identify the community structure in networks, networks can be displayed by the combination of several communities. It indicates that these communities can be discovered in networks. In other words, everyone can see them in networks. For instance, Rosvall and Bergstrom (2007) use infomap detection community algorithm to divide numbers of journals into a wide variety of modules (See Fig3). We can find that “Power Systems” is classified in “Physics” community. It represents a serious problem. If “Physics” is a security department and “Power Systems” is confidential information in networks, it will expose security department to dangerous situation since hackers can easily malfunction the networks through attacking this community. As a result, communities are not safe anymore. They confront security and privacy problems.

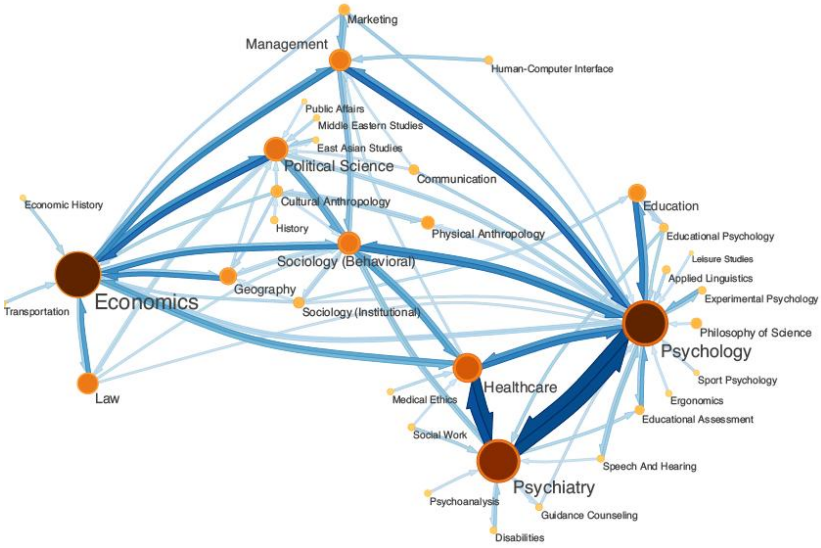


Figure3. Community Structure Example ((Rosvall and Bergstrom, 2008, p.1121)

In order to solve the problems, if we can block or hide specific and important communities such as “security department” as mentioned above, people cannot search them in networks anymore and they can be protected properly.

Therefore, different to other works, we are going to develop efficient hiding community approaches in this project. In other words, the aims of our project are not only hiding communities but also evaluating which method is the most suitable in the different kinds of environments. We devise four possible methods to achieve the target. The first one is adding edges based on the degree of each node which means the number of neighbours of each node. (See Fig4) The second is removing redundant edges. The third is adding edges based on the combination of degrees of nodes and random. The fourth one is adding edges randomly.

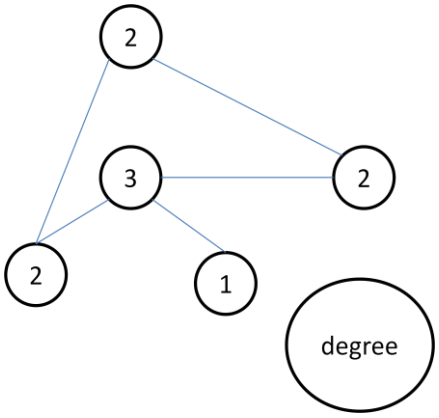


Figure4. The degree of each node

We have to detect communities before using hiding methods. Thus, we choose two different already-known community detection algorithms at first. Then, we implement four hiding methods to the communities after the division of the network. Next, we use this modified network as an input data of the detection algorithm, and apply such

algorithm again. Finally, we compare the original community structure with the latest community structure to check if the specified community still exists. (See Fig5 and Fig6)

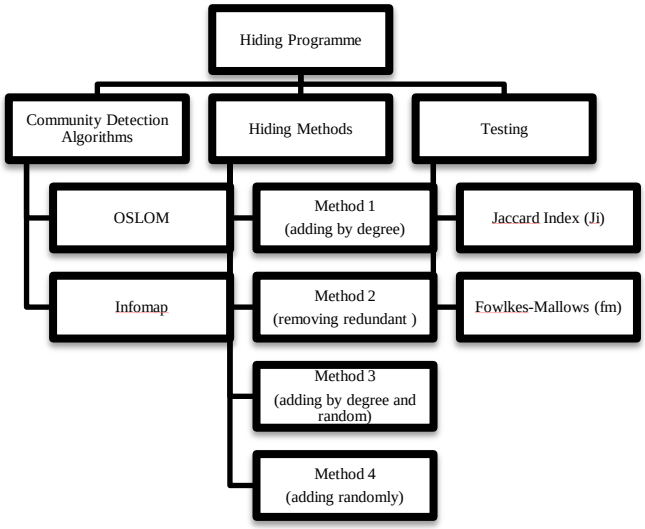


Figure5. Simple System Framework

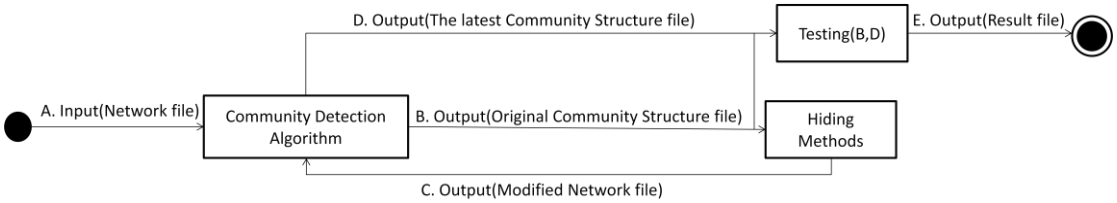


Figure6. Simple Process Diagram

In this project, we use different size and types of networks as input files to check the relationship between hiding methods and them. Furthermore, we provides two community detection algorithms to test if the hiding results will be affected. Finally, in testing part, we use two common comparing criteria to evaluate the efficiency of each hiding methods.

Currently, there are no algorithms for hiding community but many for destroying community. Destroying community is much different from hiding communities. In destroying community approaches, we can remove or add any nodes or edges randomly (Holme, Kim, Yoon, and Han, 2002). On the contrary, we cannot do such thing in hiding because it is possible to break the connection between nodes. It is a crucial point while developing hiding community approaches.

In the section II, we are going to analyse a variety type of networks, because types of network may associate with which hiding approaches we should apply. Besides, we will also introduce some background information of hiding methods, the way to check results, and some properties of networks. Section III is going to state the ideas of each hiding method in detail. Section IV is devoted to analysing the experimental data of each hiding method in each network in each community detection algorithm, and then doing

comparison and evaluation between them. In section V, we are going to summarize the hiding approaches and what we are going to implement in the future.

II. Related Work

1. Analysis of different types of networks

In introduction part, we realised the differences between destroying and hiding methods. However, the testing of destroying approaches actually provides a good direction to test the limitation of hiding methods. Since further research (Albert, Jeong, and Barabási, 2000) shows that applying different destroying algorithms to different types of networks will lead to different results, we can speculate that it also has an impact on hiding methods. Besides, understanding the structure of networks contributes to develop efficient hiding methods and find their defaults.

Following is going to introduce different kinds of networks including their formation and properties. These networks will be a data set for hiding community programme in section IV.

A. Random Graph

Recent research (Fortunato, 2010) states that random graph is a disordered graph. The link probabilities between the random pairs of vertices are equal for all vertices. The distribution of edges among vertices is homogenous. It means that, the degree of a vertex is similar or equal to other vertices in the same network. (Crucitti, Latora, Marchiori, and Rapisarda, 2004) There are different genres of random graph such as ER model, BA model, and WS model.

a. ER model

Erdős–Rényi (ER) model produces the network with exponential tail. (Albert, Jeong, and Barabási, 2000) At the beginning, there are only nodes without any links. Edges are randomly selected and added to random pairs of vertices. One feature of this model is that there are no multiple edges between nodes. (Holme, Kim, Yoon, and Han, 2002) Besides, the degree of each node is similar. (See Fig7(a))

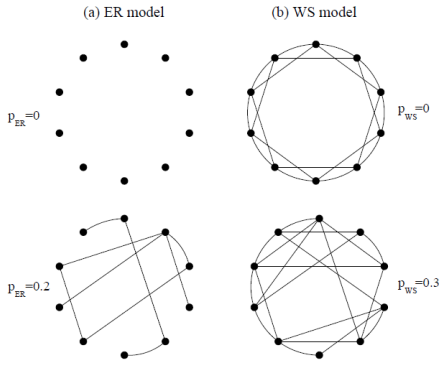


Figure7(a). The formation of ER model (b) The formation of WS model (Barab´asi, Albert, and Jeong, 1999, p.5)

b. BA model

Barabási–Albert model (BA model) is a scale-free network which is the network with power-law tail. It means that most of the nodes have few neighbours. (Barabási–Albert, 1999) (See Fig8) play important roles in scale-free network. (Albert, Jeong, and Barabási, 2000) The network is generated by the process of preferential attachment. (Dekker, and Colbert, 2004) At first, there are no edges but m_0 nodes. At each time step, a node is added progressively, and the edges are added according to the degree of nodes that they connect. The BA model leads to low clustering in networks. (Holme, Kim, Yoon, and Han, 2002) It definitely affects the result of community detection algorithms.

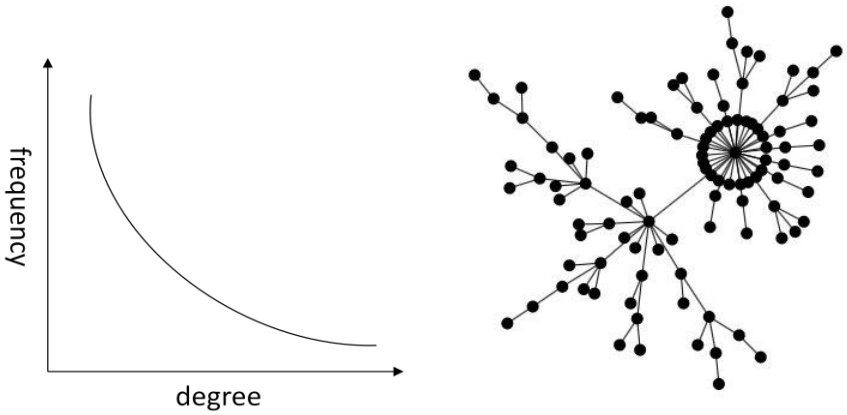


Figure8(a). Power-low distribution

(b). BA model (graphstream-project.org, n.d.)

c. WS model

Watts-Strogatz model (WS model) represents the properties of small-world networks which are the characteristics of real networks. (See Fig9 and Fig7(b)) WS model consists of numerous clusters. The density of each cluster is really high and there are some shortcuts between clusters. Therefore, one of the features of this model is that the connection between nodes in the same clusters is much higher than that between clusters. (Liu, 2012)

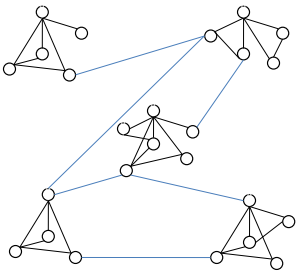


Figure9. WS model (Liu, 2012, p.1)

B. Real network

On the contrary, compared to random graph, real networks are inhomogeneous distribution. There are huge differences between the degrees of vertices. The networks contain the nodes with low degree and the nodes with high degree. One of well-known cases of real network is called 'karate club' study of Zachary. (Girvan, and Newman, 2004) (See Fig10)

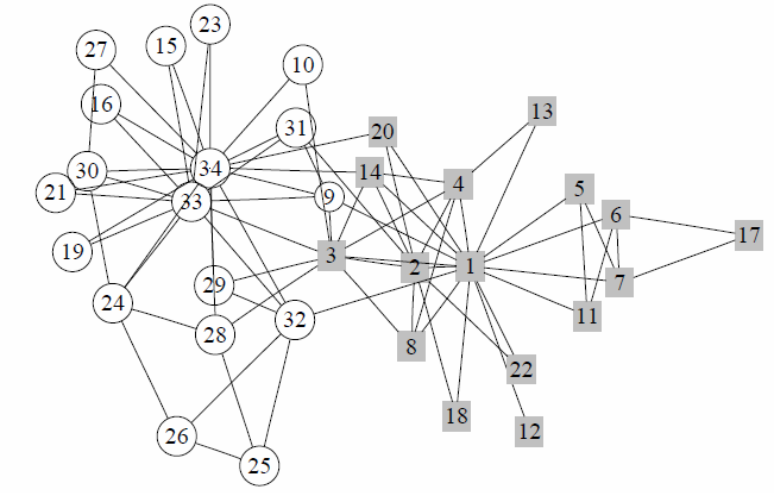


Figure10. The karate club study of Zachary. (Newman and Girven, 2004, p.10) From node1 to node33 are the administrators and instructors. Grey square displays the node connects with instructors and white circle means the node connects with administrators.

In contrast to real network, random graph has some different properties. For example, although WS model represents one of real networks features, the degree distribution is still impractical, not as real network with scale-free inhomogeneous distribution. On the other hand, even if BA model is scale-free distribution, it cannot produce high clustering like WS model and real networks. (Holme, Kim, Yoon, and Han, 2002) Through these different properties, we can analyse that hiding methods is suitable for which types of networks and find its restriction for each networks.

2. Robustness

Recent research (Scellato, et. al. ,2011) argues that robustness indicates the degree of the community that it can tolerance when under attack. Moreover, Albert, Jeong, and Barabási (2000) have expressed a similar view that robustness is the ability of nodes to be unaffected after ruining their community. It is closely related to one of our hiding methods that deletes edges. As we mentioned in section I., we cannot remove edges randomly because it might cause the disconnection between pairs of nodes. Consequently, after using this hiding method, the ability of nodes should not have a huge impact. Moreover, if communities suffer from low level of attack (the small fractions of

vertices/edges removed), rewiring some edges can increase robustness effectively against the attack. (Beygelzimer, Grinstein, Linsker, and Rish, 2005) In brief, we can infer that there is a connection between hiding method about deleting edges and robustness of networks.

The measure of robustness of the networks is associated with *edge connectivity*, and *node connectivity.* Edge connectivity is the minimum number of edges that have to be removed among two vertices in order to leave no path between them. However, edge connectivity cannot display the removed edges are very important. (Matossian, 2007) On the other hand, node connectivity is the minimum number of removing nodes which leads to the disconnection in networks. (Dekker and Colbert, 2004) In here, we only consider edge connectivity.

3. Community detection algorithms

There are numbers of kinds of community detection algorithms nowadays. We selected two of them that their results and ideas are quite different in order to test whether the detection algorithms have a heavily influence on hiding methods. One is infomap (Rosvall and Bergstrom, 2007) and the other is called Order Statistics Local Optimization Method (OSLOM) (Lancichinetti, Radicchi, Ramasco, and Fortunato, 2011).

A. Infomap

Infomap is successful, efficient and much more accurate than most of community detection algorithms.(Ahn, Bagrow, and Lehmann, 2010) The time complexity is O(m). The idea of Infomap is compressing the information flows on networks, in other words, compressing the information of a dynamic process on networks, which is called random walk (Lancichinetti, and Fortunato, 2010). The process is as follows (See Fig11): At the beginning, the algorithm uses Huffman code to name each node in order to describe the path of a random walk. This code can save space by encoding with short length of the name. After giving the name, the path can be displayed in a series of encoded nodes. For example, in figure 11, the path started from node 1111100 and ended to the node 00011. Next, through implementing a two-level description, which is done by using greedy search and simulated annealing to optimising the quality function: the minimum between the original node and compressed node, major groups receive the unique name, but the name of each node in different group is reused. Because of the names of clusters, the random walk can be shown with a series code again but the size is much smaller than previous one. Finally, it represents only the modules name. (Rosvall, and Bergstrom, 2008)

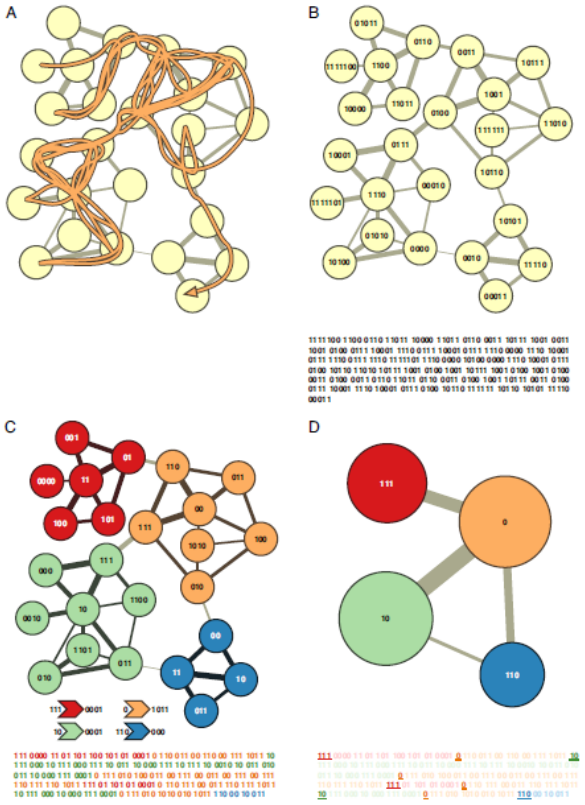


Figure11. The process of Infomap community detection algorithm
(Rosvall and Bergstrom, 2008, p.1119)

B. OSLOM

OSLOM is a multi-purpose method than can be widely used for overlapping communities, hierarchy communities and dynamic communities. Moreover, it has good performance in artificial networks and real networks. The main idea of this method is searching the significant cluster through using local optimisation of a fitness function to compile statistics data so that we can find the most important cluster. Following is the process:

- *First, it looks for significant clusters, until convergence;*
- *Second, it analyzes the resulting set of clusters, trying to detect their internal structure or possible unions thereof;*
- *Third, it detects the hierarchical structure of the clusters.*
  (Lancichinetti, Radicchi, Ramasco, and Fortunato, 2011, p.5)

The partition result displays in Figure12.

There are some characterisations of OSLOM algorithm that affect the community structure and the input network. First one is “significant clusters”. The significance of clusters is decided by fitness measure. Because of a parameter P which is set as constant in initial and indicates, it is difficult to detect random networks. In other words, the result might just contain few clusters. Second one is “homeless vertices”. If the input network is a random network, most of the nodes

are seen as homeless. That is to say, nodes are not divided into any communities. The third is "overlapping communities". The communities might intersect with each other and share some nodes. These nodes could be classified into numbers of communities. The fourth is "cluster hierarchy". Because OSLOM makes analysis of the hierarchy structure of the clusters, and computes depths in each branches, some nodes becomes homeless after implementing the algorithms in order to display hierarchical structure. (Lancichinetti, Radicchi, Ramasco, and Fortunato, 2011)

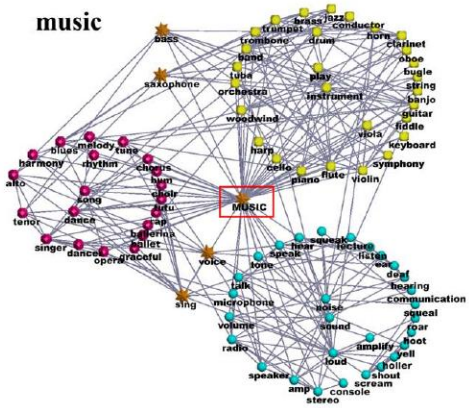


Figure12. Partitioning real network with OSLOM. The star icon represents the overlapping nodes. (Lancichinetti, Radicchi, Ramasco, and Fortunato, 2011, p.15)

4. Comparison with Two Communities

Nowadays, there are three main types of comparing clusters methods: counting pairs, set matching and variation of information. We select counting pairs because set matching is not proper for comparing the clusters that their sizes are quite different, and counting-pair is simple and easily defined. (Traud, Kelsic, Mucha, and Porter, 2010)

In this section, we are going to introduce two counting-pairs criteria which are commonly used for compare clusters. The first one is Jaccard Index (Altman et. al., 2001) and the second one is Fowlkes-Mallows index (Fowlkes and Mallows, 1981). The reason for using both is that they are easily affected by evaluating the differences based on only thinking of the nodes belonging to which communities. (Bae, Bailey, and Dong, 2006) Therefore, in order to have precise result, we consider both of them.

The principle of counting pair is counting the nodes in which communities. In general, given two communities C_1 and C_2, counting pairs consists of four parameters: N_{11}, N_{00}, N_{10}, and N_{01}.

N_{11} is the number of pairs of nodes that in the same clusters (C_1, C_1)

N_{00} is the number of pairs of nodes that in the same clusters (C_2, C_2)

N_{10} is the number of pairs of nodes that in the different clusters (C_1, C_2)

N_{01} is the number of pairs of nodes that in the different clusters (C_2, C_1)

$$N_{11} + N_{00} + N_{10} + N_{01} = \frac{n(n-1)}{2}$$

n: the number of nodes

A. Jaccard Index

$$Percentage\ of\ similarity(C_1, C_2) = \left(\frac{N_{11}}{N_{11} + N_{10} + N_{01}a}\right) \times 100\%$$

B. Fowlkes-Mallows Index

$$Percentage\ of\ similarity(C_1, C_2) = \left(\frac{N_{11}}{\sqrt{(N_{11} + N_{10})(N_{11} + N_{01})}}\right) \times 100\%$$

Since understanding the construction of different networks and their features, we can find their weakness and then develop the efficient hiding strategies based on these points. Furthermore, the robustness provides a new direction of devising hiding methods. Besides, detection community algorithms also play an important role in hiding methods. Compared to OSLOM, infomap has a good performance in community structure and the running time, but cannot classify networks into overlapping communities. On the contrast, OSLOM is suitable for overlapping communities, but has a bad result in random networks. These differences make the community structure quite distinct. Finally, we evaluate the result through observing the percentage of Jaccard Index and Fowlkes-Mallows Index. The higher percentage of them, the more similar the two communities are.

III. The approaches to hide communities

Hiding communities is a new research in network, so there is little information related to this topic. We started to come up with the hiding approaches through considering our purposes. As mentioned in above, the aims of this project are not only to hide the community but also find which hiding method is "efficient" in which kinds of network. We give the definition of "efficient" in the project as the number of adding or removing edges. Although adding all edges between nodes in one community to other communities can easily reach the target due to the structure of community (See Fig13), the number of adding edges is really quite huge. This wastes much time and space in doing this in many cases. (See section IV.) Therefore, we have devised three methods that take the efficiency into account, and one method that just adds edges randomly. Besides, we modified two of these methods to test in section IV. Since the community structure is that the connection between nodes is denser in the same cluster than that in the different clusters, we add inter-communities edges, and remove the intra-community edges to hide community. Moreover, we set parameters to control the performance of each method in order to hide efficiently.

Before introducing hiding methods, we divide them into two classes based on

different main ideas. Besides, in this section, if we mentioned community 0 or the targeted community, it indicates the community we would like to hide.

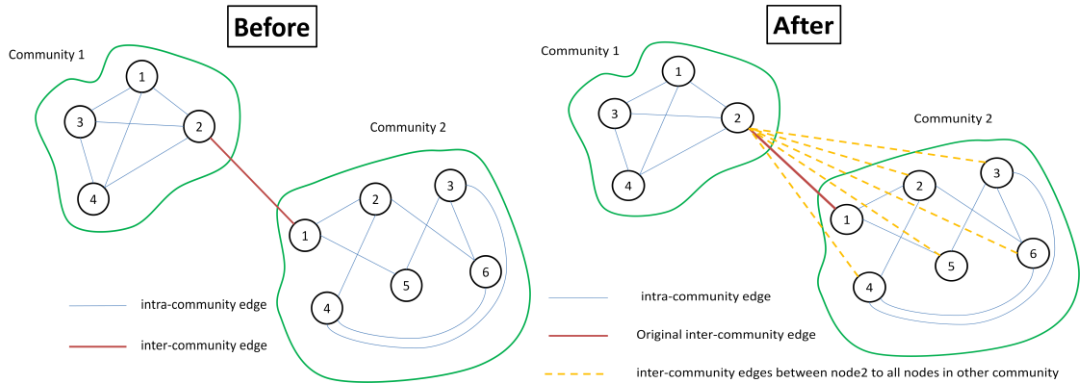


Figure13. (a)(b) The example of add all inter-communities edge between each pair of nodes in networks.

(a) Before adding inter-community edges, the community structure is still well-defined that nodes in the same community interact with each other more frequently than with those outside the community.

(b) After adding inter-community edges, node2 links with more nodes in other community than in the same community. Therefore, the structure of community 1 is broken.

1. Low Degree

In this section, we take the degree of nodes for the purpose of decreasing the number of adding edge. The reasons are as follows:

If the nodes with lower degree, it shows that the node does not have many neighbours. It is more possible to break the community structure by the fewer number of adding edges. (See Fig14)

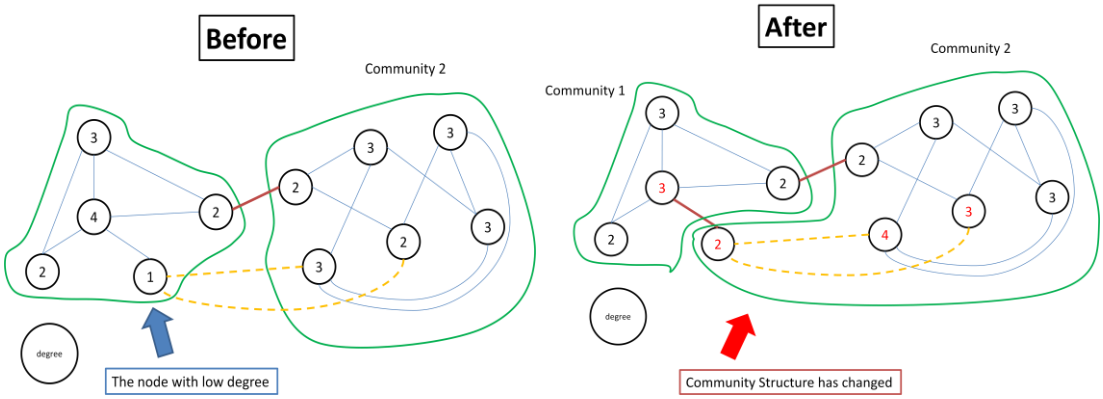


Figure14. (a)(b) The example of add inter-communities edge based on lower degree.

(a) Before using community detection algorithm, we added two inter-community edges based on lower degree of node.

(b) After using community detection algorithm, the node1 is classified into the other community. The structure of community1 and community2 has changed.

On the other hand, if the node with higher degree, we usually have to add more inter-community edges to break the community structure than the nodes with lower degree, since the higher degree can be one of criteria to judge the importance of point. (Freeman, 1979) (See Fig15)

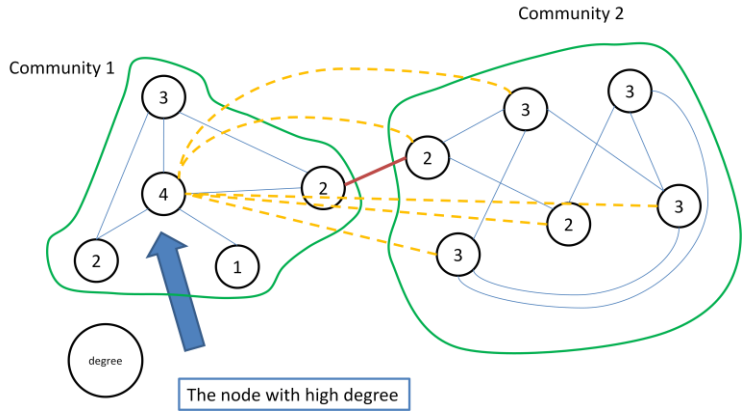


Figure15. The example of add inter-community edges based on high degree.

If we compare between figure13, figure14 and figure15, referring to hiding the same community, we can find that adding edges from the nodes with lower degree only connects fewer numbers of edges, while adding all inter-community edges and adding edges from the node with higher degree connect more number of edges.

A. Method 0

Method 0 is one of the hiding approaches that use the concept of degree of node. We have to consider how to select the end points in the other community since the start point is definitely in the targeted community. We use a parameter called range to settle this problem. Range is defined by the degree of the start point in targeted community. If the degrees of nodes in the other community are between the degree of the start point plus range, and the degree of the start point minus range, we build the edge between the start point and these nodes.

$assuming\ x$ is the degree of the start point (1)

assuming y is the degree of nodes in the other community (2)

If the node in the other community satisfies following condition,

If $x - range > 0$, $x - range < y < x + range$ (3)

If $x - range \leq 0$, $1 \leq y < x + range$ (4)

Then, we link the edge from x to y

Following is the example,

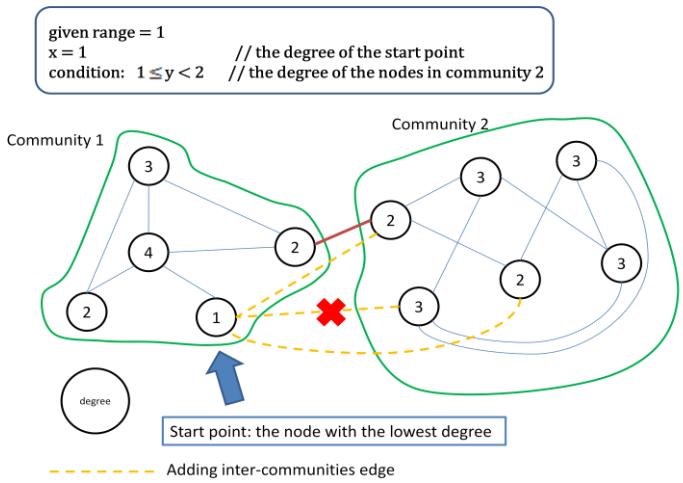


Figure16. The example of range

As we just mentioned above, the algorithm starts from the node with the lowest degree. Since it is nearly impossible to implement hiding method 0 in one time to reach the target, we have to define another parameter called percentage. Percentage means the top percent of lower degree in the targeted community. For example, from Figure 17, if we would like to extract the top forty percentage of the lower degree, node 1 and node 5 will be selected since times is equal to 2. Times is how many numbers of nodes should be selected. It is defined by,

Times = percentage × (the number of nodes in community 0)

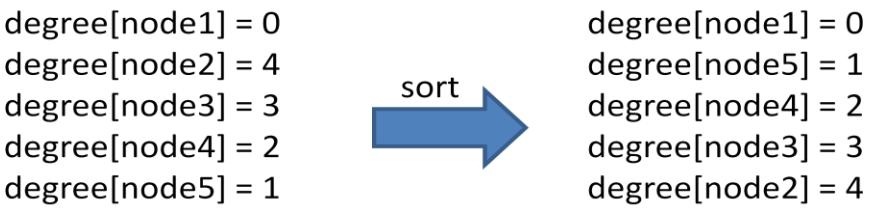


Figure17. The example of using percentage.

Following provides the simple process of method 0 and its pseudo code. (See Fig18 and Fig19)

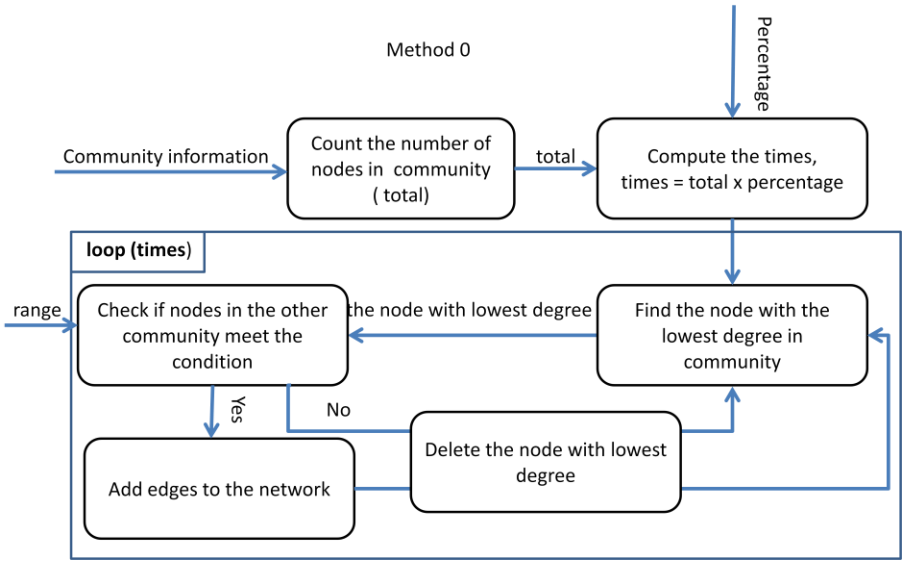


Figure18. Simple Process Diagram for method 0

```
Set percentage
Set range
Set  total to the number of nodes in the community
Set  times to total multiplied by percentage
FOR i = 1 to times
        find the node u with the lowest degree in the community
        Set x to be the degree of node u
        FOR j = 1 to the size of the other community
                Set y to be the degree of node j
                IF ( x– range < 1 )
                        IF ( 1 <= y < x + range)
                                Add edge between node u and node j
                        ENDIF
                ELSE
                        IF ( x – range < y < x + range )
                                Add edge between node u and node j
                        ENDIF
                ENDIF
        ENDFOR
        Set y to zero
ENDFOR
```

Figure19. Pseudo Code for method 0

There is a limitation for this method that it cannot be used in networks that is divided into only one community. The idea of this method is adding inter-communities edge. Consequently, if there is no more than one community, it is impossible to add edges.

B. Method 1

Method 1 is the only one removing edges in hiding methods. Since hiding is different from destroying, deleting edges without definite plan might result in disconnection in the community. Accordingly, researching the redundant edges to delete is the main idea of this method.

As we mentioned in II.2., robustness represents the edge tolerance which means

that community keeps original even though suffering from attack. The measure of robustness is related to edge connectivity which is the minimum number of removing edges causes the independence between nodes. In the other words, we have to find the pairs of nodes which have more than one path between them so that they will not be affected even if deleting edges. For instance, from Figure20 (a), we can know that there are five paths between node 1 and node 2. If we delete edge(1,2), node1 can still interact with node2 via other paths. As a result, in this case, edge(1,2) is a redundant edge. Besides, through observing the redundant edges, we discovered that they appear if there is a cycle. (See Fig20 (b))

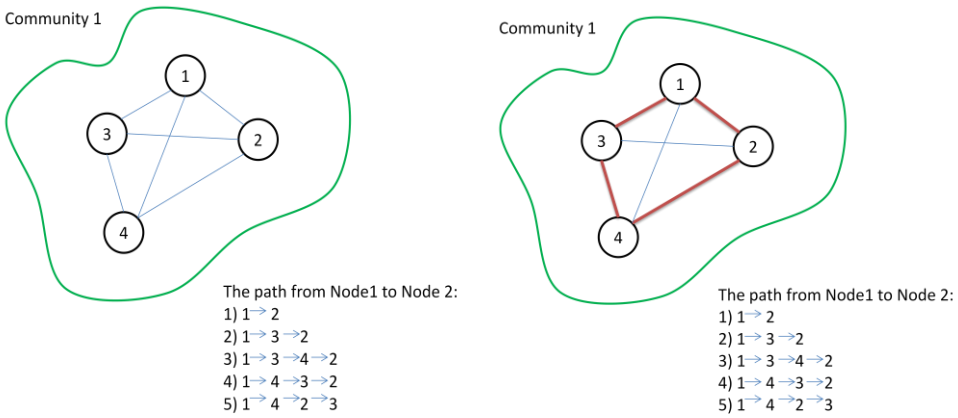


Figure20 (a) The example is numerous paths between pairs of nodes

(b) The example of cycle in community

Hence, the most important is finding cycle and all possible paths between nodes. These can be done through implementing modified BFS. (See Fig21) The different part is that, if we notice this point has been walked again, it points out there is a cycle. So, we can delete the edge between this point and last point. (See Fig22)

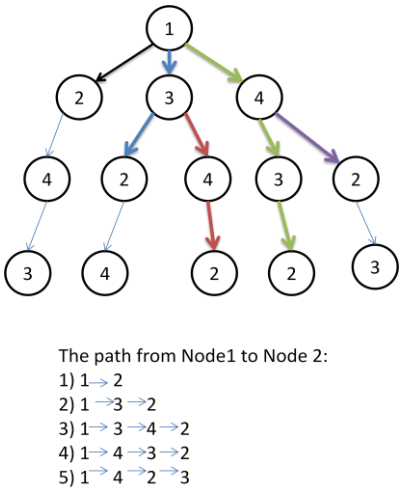


Figure21.BFS

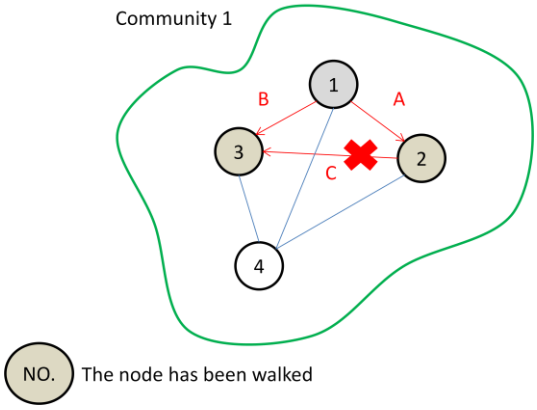


Figure22. An example of deleting edge in BFS

Since we realize how to find the paths, there is a question that how many times BFS should run. Therefore, in this method, we also set a parameter called percentage_1 to decide how many edges we would like to delete.

Following provides the simple process of method 1 and its pseudo code. (See Fig 23 and Fig24)

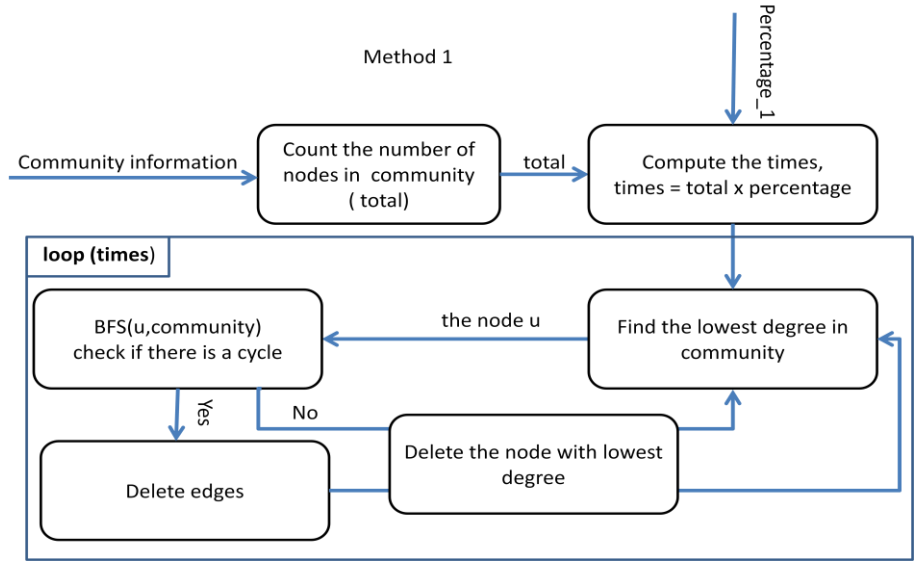


Figure23. The process of method 1

```
Set percentage_1
Set  total to the number of nodes in the community
Set  times to total multiplied by percentage_1
FOR i = 1 to times
            find the node u with the lowest degree in the community
            CALL BFS(u, community)
ENDFOR

void BFS(u, community)
{
   FOR each neighbour of the node u
            IF neighbour v is already been WALKED
            THEN
            delete edges between u and the v
            ENDIF
   ENDFOR
}
```

Figure24. Pseudo Code for method 1

The limitation of this method is that it cannot implement in the community that only has one node because the main idea of this is using intra-communities edges. If there is no intra-communities edge, it is impossible to use this method.

2. Random

In this section, we use another way to hide community, random. Random does not seem to be a efficient method compared with method 0 and method 1. However, it is a good way to do even if the efficiency might not so high. Such approach is direct, and common. According to Holme, Kim, Yoon, and Han(2002), they use two different ways

to design the attack strategies. One is targeted attack. The other is random failure. The former one attacks the community based on some properties such as degree. The latter one attacks nodes or edges randomly. But the hiding result is excellent. This is one of reasons that we develop the random method. Besides, such approaches are also a good test if compared to the target hiding, method 0 and method 1.

We are going to introduce two hiding methods based on random as follows:

A. Method 2

Method 2 starts from the point in the targeted community. We select the node in the community randomly and then link the node with other nodes in other communities randomly. (See Fig25) The number of adding edges is decided by a parameter, percentage_2 which is the percentage of the number of edges in this community.

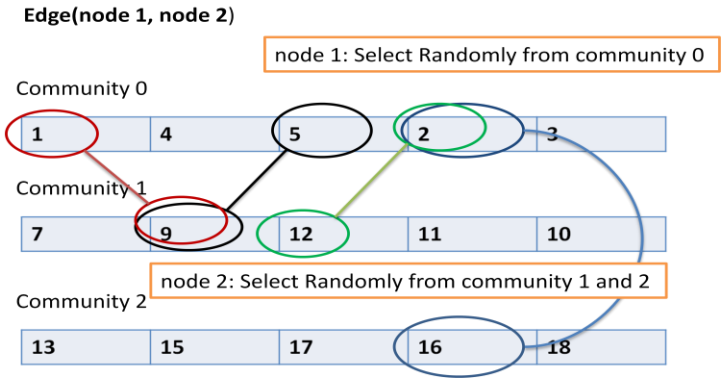


Figure25. An example of method 2

Following provides the process of method 2 and its pseudo code. (See Fig26 and Fig27)

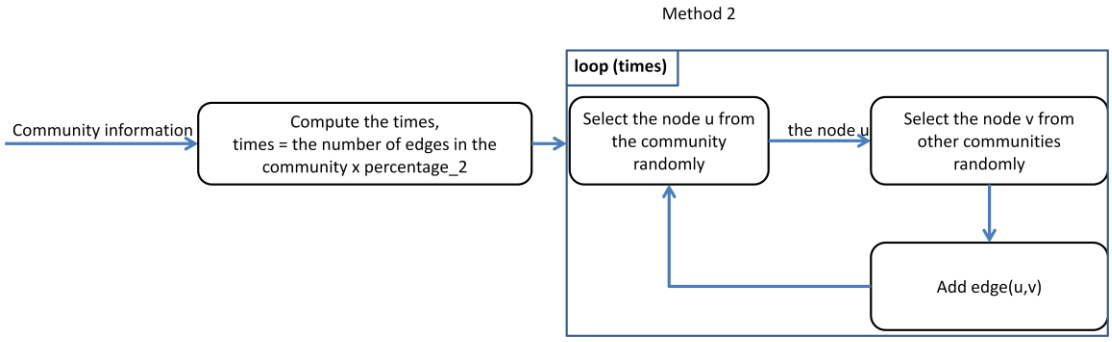


Figure26. The process of method 2

```
Set percentage_2
Set  total to the number of edges in the community
Set  times to total multiplied by percentage_2
FOR i = 1 to times
            find the node u in the community
            find the node v in other communities
            Add Edge(u,v)
ENDFOR
```

Figure27. Pseudo Code for method 2

The limitation of method 2 is that it cannot use in the network that only contains one community since the total edges in the targeted community is zero.

B. Method 3

Compared to method 2, method 3 is totally random. It selects pairs of nodes to build randomly. (See Figure 28) There is also a parameter called percentage_3 that is the percentage of the total number of edge in the network which control how many edges should be added.

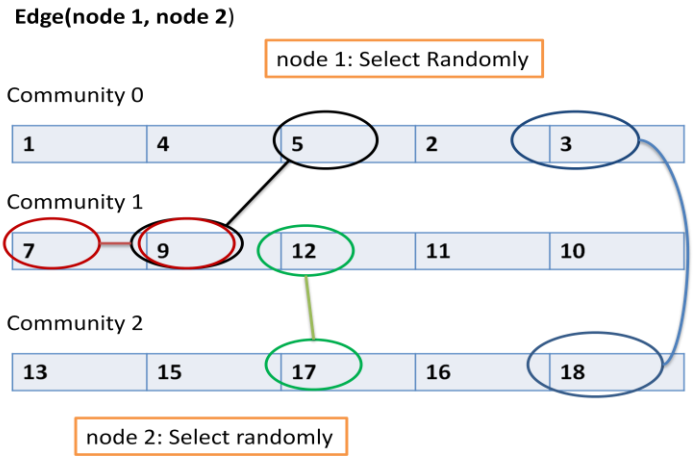


Figure28. An example of method 3

Following provides the process of method 3 and its pseudo code. (See Fig 29 and Fig30)

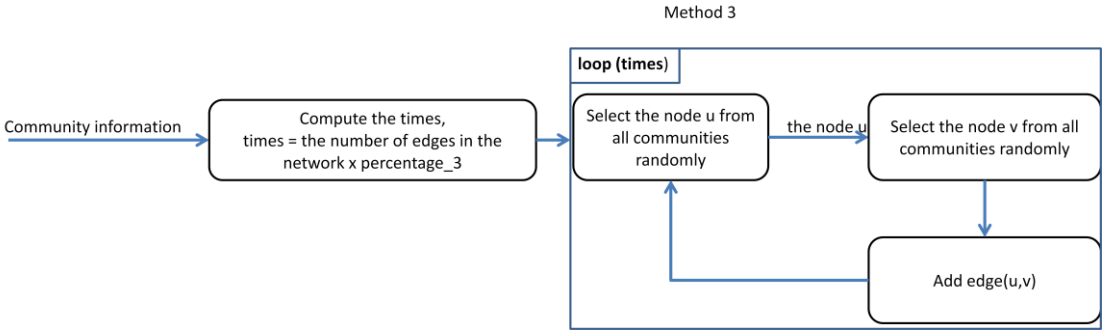


Figure29. The process of method 3

```
Set percentage_3
Set total to the number of edges in the network
Set  times to total multiplied by percentage_3
FOR i = 1 to percentage_3
        find the node u in all communities
        find the node v in all communities
        Add Edge(u,v)
ENDFOR
```

Figure30. Pseudo Code for method 3

Method 3 does not have any limitation. It might add both inter-communities edges and intra-communities edge.

C. Modified methods

We designed modified method 2 and modified method 3 after doing the testing. Since the number of adding and removing edges was not enough in some cases, the similar percentages were up to 100% that means the original community is almost the same as the community after running hiding methods. Accordingly, we modified the times in method 2 and method 3 to improve the performance. Due to the lack of the number of times, we change the times to a series of values before implementing the hiding methods. Besides, according to the main idea of method 2, it is usually more efficient than method 3. Therefore, the values for method 2 are smaller than method 3. There is no limitation anymore for modified method 2 compared the original method 2.

These methods are really different in different starting points, the way to connect edges, and the number of adding and removing edges. Method 0 uses lower degree as start point and range to decide which point to link. Furthermore, the number of adding edge is the number of nodes with the top percent lower degree. Method 1 also selects the lower degree as start point, and it removes edge through using BFS. The number of removing edges is determined as the method 0. Method 2 links from the nodes in the targeted community to the nodes that select randomly in other communities. The number of adding edges is the percentage to the number of edges in the community. Method 3 link pairs of nodes in the network randomly. The number of pairs of nodes to be linked is the percentage to the number of edges in the network. The difference between the original methods and modified methods is the number of adding edges. In modified methods, this is defined by a series of fixed values.

Besides, there are some restrictions in hiding approaches. Method 0 and Method 2 cannot implement in the network that only contain one community. Method 1 cannot use in the community that only has one node.

The differences as mentioned above have a profound impact on not only hiding results but also the efficiency. We will represent these in the next section.

IV. Evaluation and Experimental Results

After devising hiding methods, we are going to test them and analyse the results. As described in section II.1., different kinds of networks affect different results. Therefore, the testing networks we are going to use can be divided into two parts. The first one is real network. Since there are thousands of networks nowadays, we picked three popular networks. Their size can be classified into small, medium, and huge. So, we can evaluate if the size of network has an impact on hiding methods. The second one is random graphs. This part can be divided into three types, BA, ER, and WS model. Due to their different formation, we can examine if all hiding methods can be used in these networks and which method is the most helpful. Furthermore, because the structure of random graphs is quite distinct from real networks, we can compare the result of random graphs

to real networks,

Besides, as mentioned in II.3., we introduced two community detection algorithms. One is OSLOM, and the other is infomap. It is notable that these two methods will create different community structure after each time even if there is no change in networks. Moreover, the size of network has an impact on this difference. If they are implemented in a small size of network, the results will make slight differences while the results will have a drastic change in a huge size of network. However, referring to random results in the same size of networks, the possibility of the community structure to be changed each time with using infomap detection algorithms is much more than using OSLOM. Consequently, we ran our programme fifty times for each hiding method and each network and then balanced the data to get final results.

In section III., we have introduced four hiding methods. All of them have parameters which take control of the number of removing and adding edges or the ways to link edges. Accordingly, we modify such parameters each time while testing. Besides, because there are two parameters in method 0, we have to evaluate their relationship to find the most appropriate pair of values, so that we can use in the following testing.

Due to the reasons described above, we will do testing on two parameters in method 0 at first, and then use different values of parameters for each hiding method to test in each type of networks and community detection algorithms. Finally, we will make tables to compare the results of these methods and point out the most useful one.

As we mentioned in introduction, one of our aims is to create “efficient” approaches. Therefore, we regard the number of adding and removing edges as a criterion to decide such aim. The other target is to hide community. In section II.4., we stated two criteria to determine whether the community has been hidden. Hence, we define the success of our hiding methods that, if Ji is under 65% and Fm is under 70%, it indicates we hide the community since the lower the percentage represents the more dissimilar the two communities are.

Before evaluation, there are some terms to be explained. If we mention hiding result, it is the percentage of Ji and Fm. If we say efficiency, it usually refers to the number of adding or removing edges. Furthermore, the x axis of following scattered graphs indicates the parameter for each hiding method. If the name of y axis is percentage, it represents the percentage of Ji and Fm. On the other hand, if the name is size, it stands for the number of adding ore rewiring edges. Furthermore, the targeted community is the community we would like to hide.

1. The Relationship between Percentage and Range in Method 0

Since there are two parameters that have an impact on the hiding result, we are going to test which one is more significant. Following are the experiment data: (See Figure31)

Given the network is Zachary's karate club, and the community detection algorithm is OSLOM,

Assuming range is constant, and equal to 1, (See Fig31(a))

Assuming percentage_0 is constant, and equal to 0.1, (See Fig31(b))

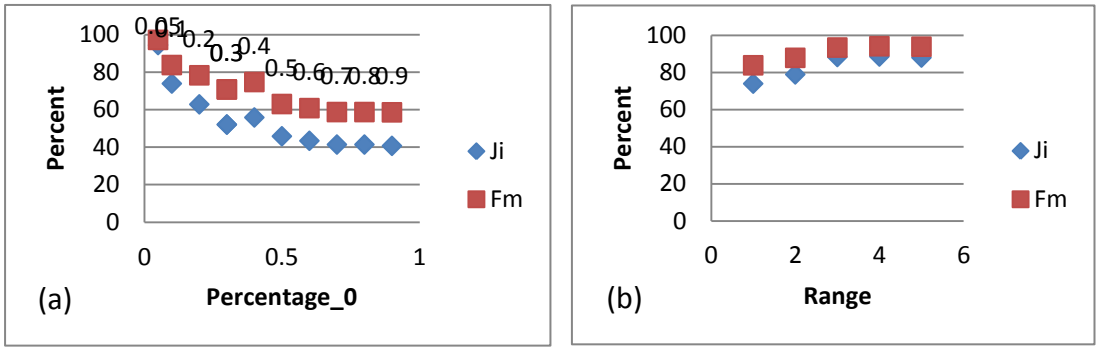


Figure31(a)(b). Find the relation between percentage_0 and range

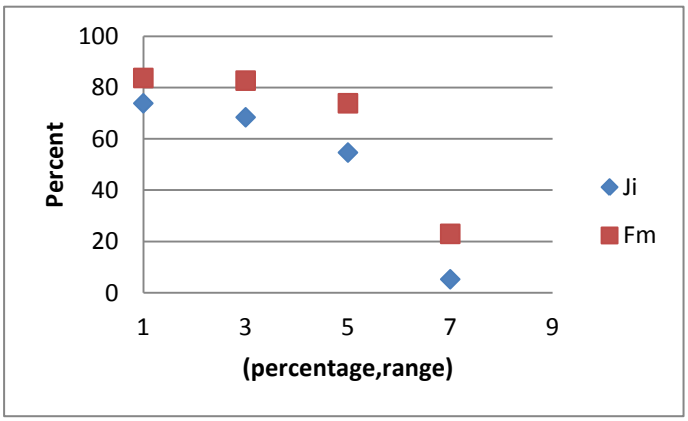


Figure31(c). Find the relation between percentage_0 and range

These charts show the connection between percentage_0 and range. There was a substantial difference between Figure31(a) and Figure31(b). Fi and Rm decreased steadily if range was constant. In comparison with Figure31(a), they increased unexpectedly if percentage_0 did not change. Consequently, we can know that percentage_0 has more effect than range. Besides, assuming range is equal to 5, Fi and Rm declined to 54.69% and 73.9% at percentage_0 0.5 while they went up to 87.91% and 93.75% at percentage_0 0.1. Therefore, we can speculate that, if range increase, it cannot indicate the community will be hidden. However, from Figure31(c), we found that if percentage_0 and range both rose, Fi and Rm fell down dramatically. As a result, if percentage_0 and range both goes up, it is efficient for hiding.

To summarize, the percentage to controls how many nodes in the targeted community should link is more important than the range that controls the number of nodes to be linked

in other communities. Besides, range cannot be use alone. Furthermore, if both parameters increased, it is more powerful and efficient than only one increased.

2. The Relationship between Real Networks and Hiding Methods in OSLOM

In this section, we are going to use different size of real networks to test if size will influence hiding methods. The experimental data is Zachary's karate club with 34 nodes and 78 edges (W. W. Zachary, 1977), American College football with 115 nodes and 613 edges (M. Girvan and M. E. J. Newman, 2002), and neural network with 297 nodes and 2148 edges(D. J. Watts and S. H. Strogatz, 1986). Each of them can be classified to small, medium, and big size of networks.

As described above, percentage_0 and range have to increase simultaneously. Accordingly, we select nine pairs of parameters for method 0 to test such as (0.1,1), (0.2,2), (0.3,3) …. to (0.9,9).

Following are the results of different hiding methods applying in different networks. We will analyse the feature of each methods at first, and then compare methods in each networks. Finally, we will select the most efficient method for each size of networks.

A. Zachary's karate club

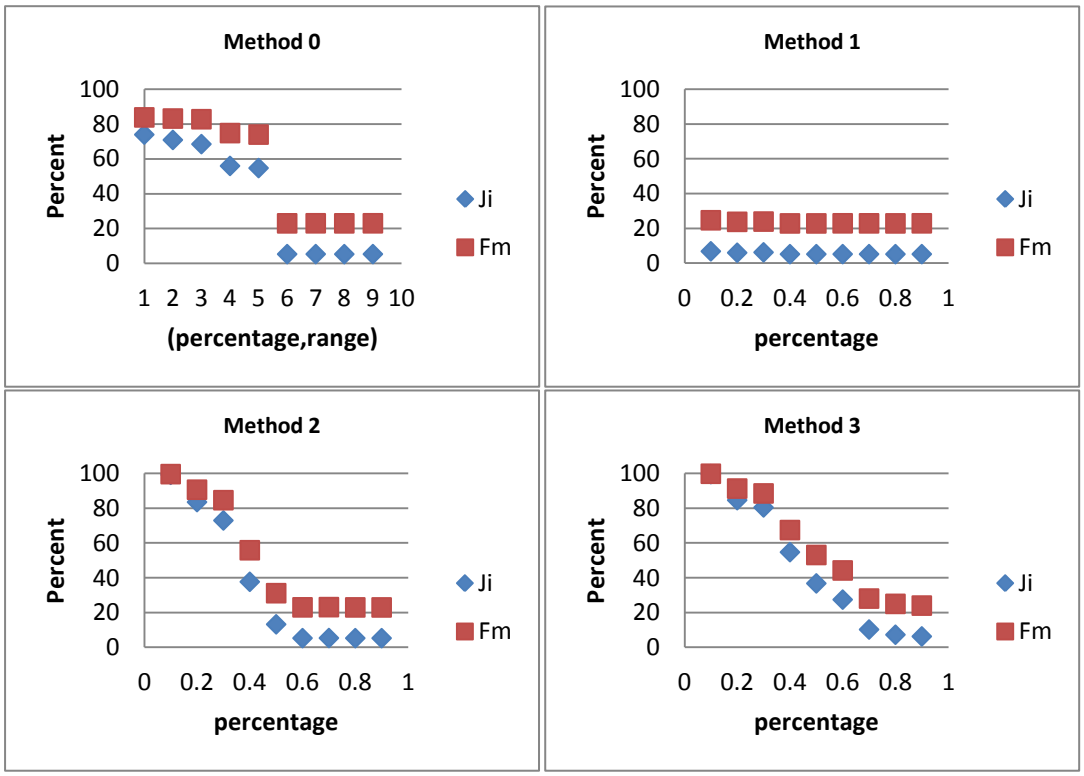


Figure32. The hiding results of each method in Zachary's karate club in OSLOM

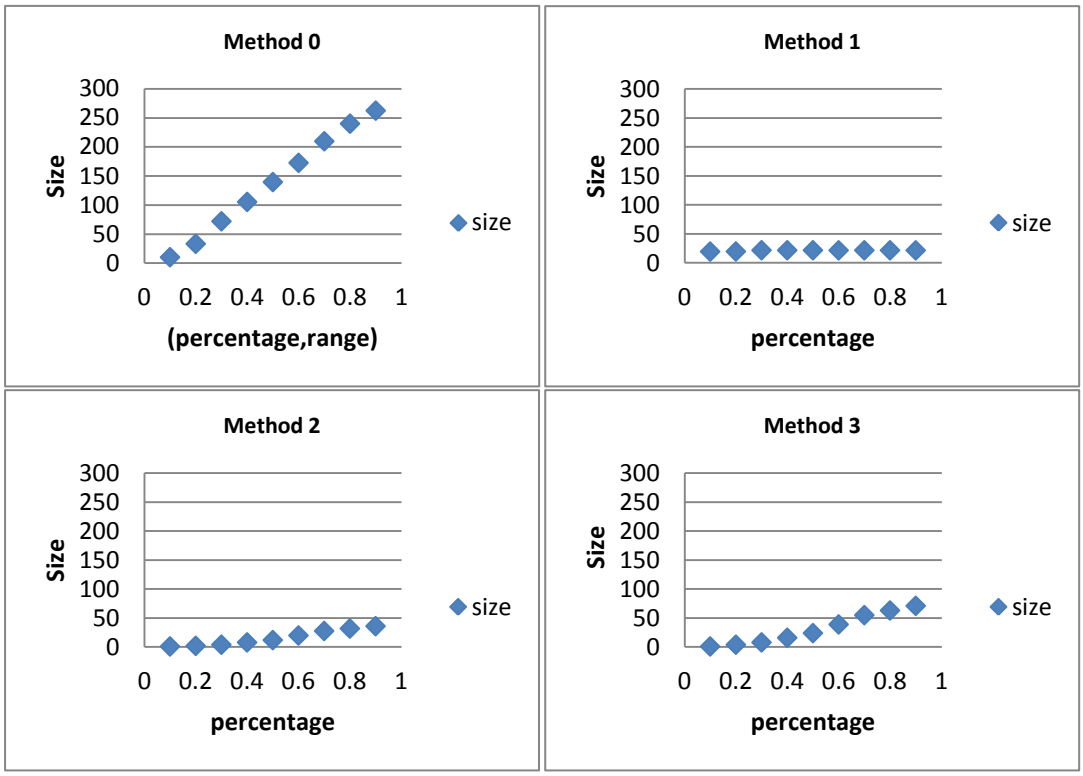


Figure33. The size of each method in Zachary's karate club in OSLOM

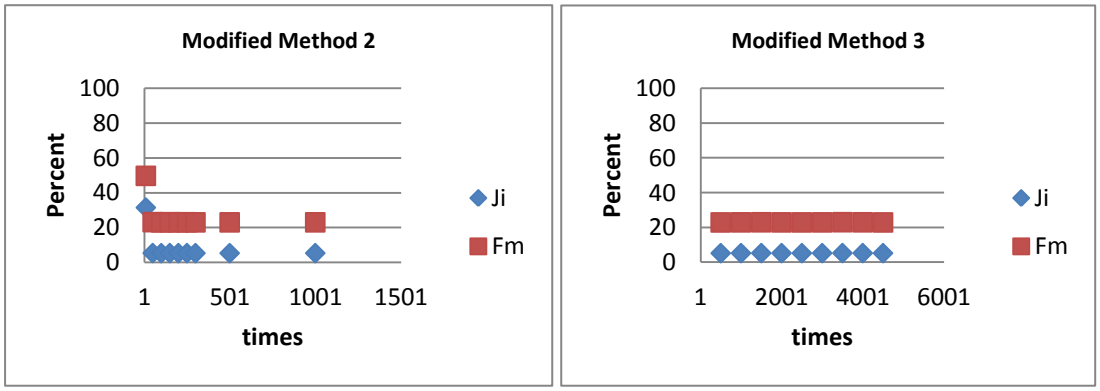


Figure34. The hiding results of modified method 2 and modified method 3 in Zachary's karate club in OSLOM

From Figure32, in method 0, Ji and Fm suddenly decreased to under 30% at (0.6, 6), but remained steady after this point. Since this method is based on adding inter-community edges, the number of nodes to be linked in the other community is almost selected. Hence, the number of such edges is already enough to combine the targeted one into the other community. In other words, the connection between nodes is denser between communities than in the targeted community. Therefore, no matter how many inter-community edges increase, there is no percentage in doing this. (See Fig32 and Fig33)

In method 1, this method can easily hide the targeted community when the percentage is 0.1. One of the reasons might be that Zachary's karate club is a small network. Thus, the possibility to have cycles inside the community is smaller.

Furthermore, the number of rewiring edges is defined by the percentage to the number of nodes in the community, so it is hard to see the change for Ji and Fm in this method. (See in section IV.2.B.) Besides, we found that it seems stable even if percentage increased. The explanation is that this method depends on removing intra-community edges. Hence, if the edges are removed by previous nodes, following nodes do not have edges to delete. It can be proved in figure33, after 0.1, the size kept to be 22 and did not alter.

In method 2 and method 3, we discovered that the hiding results levelled off after 0.6 in method 2, and so did method 3 after 0.8. Even if we added more inter-community edges, hiding results alter slightly. (See Fig32 and Fig33) The reasons are the same as the method 0. The network has become saturated. No matter how many edges are added, most of the nodes in the network already interacted with each other. Therefore, if we implement in the larger size of network, it is harder to achieve the aim. (See in the next section)

In modified method 2 and method 3, both of them reported powerful hiding results but added too many edges in this case. Therefore, in this case, efficiency was low. However, if such methods are taken into larger network, it will have obvious differences. (See in the latter section)

Referring to the number of adding or removing edges, method 0 is a slope as showed in figure33. There were many neighbours in the previous community that correspond to condition in this network so that the slope is higher than method 2 and method 3. On the contrary, method 1 is a horizontal line because no intra-community edges could be removed anymore. The size of method 2 and method 3 increased gradually..

Concerning to the start points with lower degree, the hiding results of method 0 and method 1 are nearly the same at percentage 0.6. However, method 1 is generally better than method 0 before 0.6 in this case.

Relating to add edges randomly, the results of method 2 and method 3 are similar but method 2 is slightly better than method 3. The reason is that method 2 only adds intra-community edges compared to method 3 that adds intra-community and inter-community edges. Method 2 seems to be better than method 3 in this case although there is only a marginal difference between them. (Figure33)

Analysing modified method 2 and method 3, because the network only contains 78 edges, the number of edges in modified method 2 and method 3 is hundreds times than the total edges in the network. Therefore, there is no difference between these methods in this case.

After examining the result in each method, we are going to evaluate which method is the most efficient in this network. In terms of hiding result, we are going

to design a table. If the hiding results of each method achieve one of the goals that Ji is less than 65% and Fm is less than 70%, we will select the best hiding results with the minimum of the size in the scattered graph. Finally, we take the other aim into account that, the fewer the number of adding and removing edges, the better it is. Thus, we chose the method with the lowest size to be the most efficient one

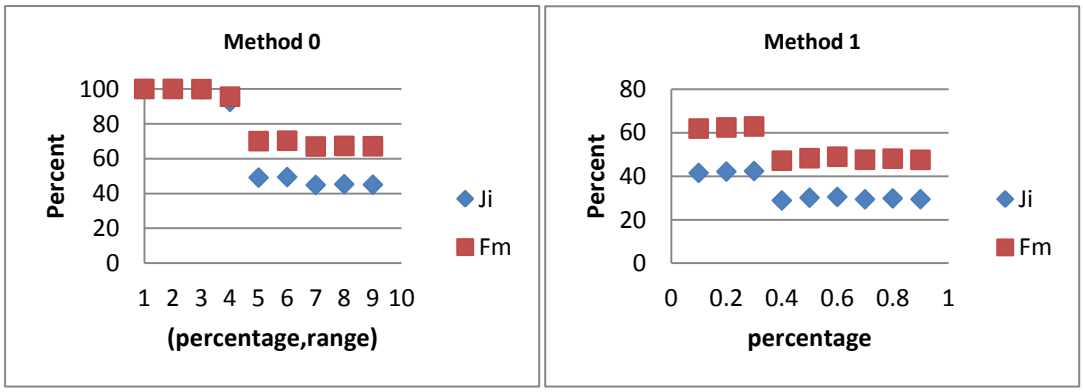
For example, method 0 could reach the target (Ji under 65% and Fm under 70%) at (0.6,6) and the percentage of Ji and Fm are (5.27%, 23%); method 1 did at (0.1) and the percentages are (6.91%, 24.74%), method 2 did at (0.4) and the percentages are (37.68%, 55.82%) , and method 3 did at (0.4) and the percentages are (54.71%, 67.42%). (See table1)

Method	(Percentage, Range)	(Ji, Fm)	size
0	(0.6,6)	(5.27%, 23%)	172
1	(0.1)	(6.91%, 24.74%)	20
2	(0.4)	(37.68%, 55.82%)	8
3	(0.4)	(54.71%, 67.42%)	16
Modified method 2		(31.43%, 49.64%)	10
Modified method 3		(5.28%, 22.93%)	500

Table1. The comparison between hiding methods for Zachary's karate club in OSLOM

From table1, under the premise that all methods meet the target(Ji <= 50% and Fm < 70%), we realise that even though the method 0 has the best hiding result in Ji and Fm, it removed 172 edges which are 20 times as many as method 2. Therefore, in this case, the performance from the best to the worst is: Method 2 > Modified Method 2 > Method 3 > Method 1 > Method 0 > Modified Method 3

B. American College football



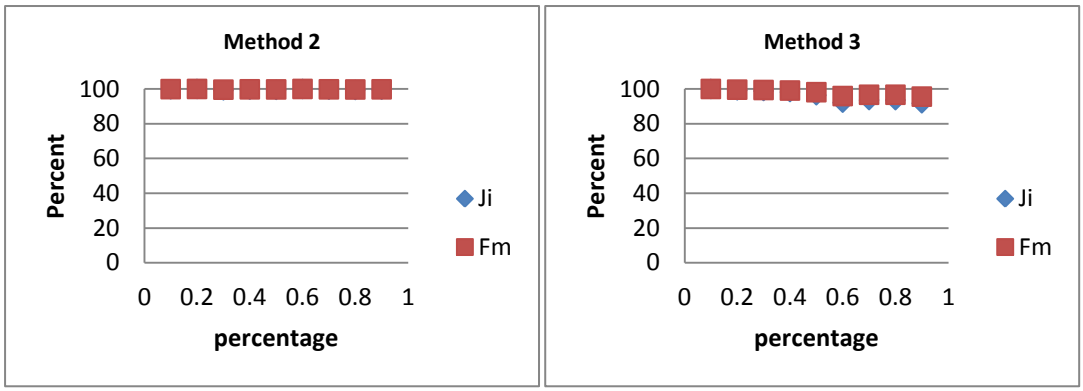


Figure35. The hiding results of each method in American College football

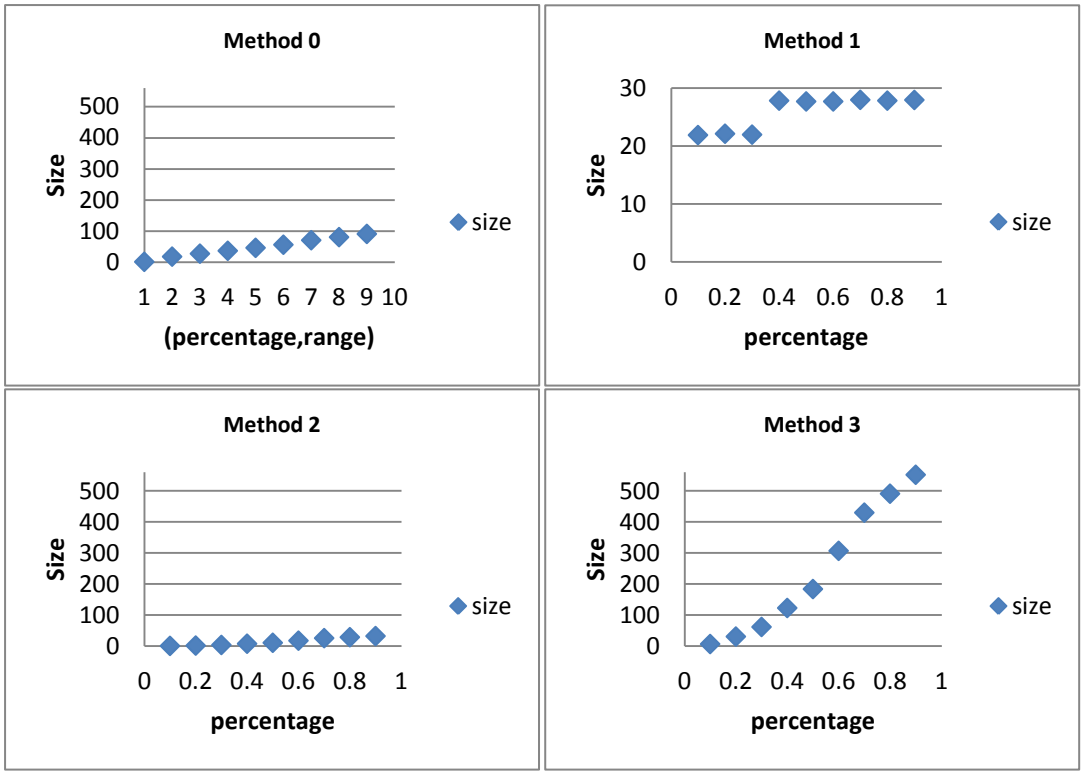


Figure36. The size of each method in American College football

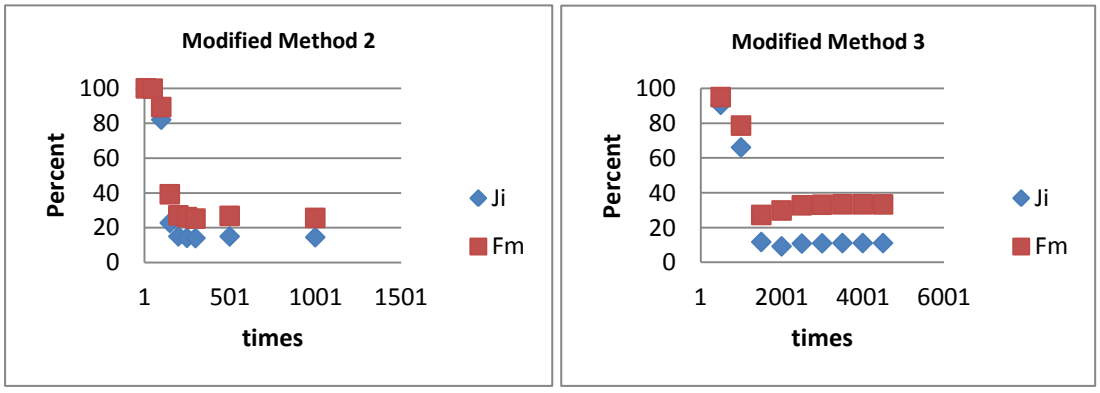


Figure37. The hiding results of modified method 2 and method 3 in American College football

From figure35, in method 1, as we described above, even if method 1 could still reach the target at percentage 0.1 as in Zachary's karate club, there was a slight

difference. Method 1 fell from (42.36%, 62.87%) to (28.84%, 47.14%) at percentage 0.4 compared to (Ji, Fm) remained constant from the beginning to the end in Zachary's karate club. In other words, this method kept deleting edges from percentage 0.1 to 0.4. After percentage 0.4, there is no edge to remove. (See Fig36) The main reason is as we mentioned above that the percentage to the total nodes in the community decide the number of inter-community edges to be deleted.

In method2, as we described earlier, when the size of network increase, the saturated situation is harder to reach. We discovered that Ji and Fm were never under 95%, even once. There are two reasons for that. The first is that the number of adding edges is not enough which is defined by the percentage of the total edges in targeted community. In this case, one possible condition is that, there are few inter-community edges in this community, so the number of intra-communities edges is also affected. The other possible one is that there are more inter-community edges when the networks become bigger. Therefore, it is more difficult to break the community structure. The second cause is that the nodes to be linked are selected randomly in other communities. Hence, if the networks become larger, there is a big chance that previous nodes link to the nodes in one community, and latter nodes connect to the nodes in the other community. In other words, it loses the opportunities to increase strong correlation between the targeted community and a certain community. Thus, in order to solve the second problem, we have to add more edges so that the possibility to connect to one community will increase.

Hence, in modified method 2 and method 3, because they almost added five times number of edges than the total edges in this network. We can see the hiding results are quite useful. If the number of edges is added around the certain number, the percentage will not increase and is in saturated status again. In this case, modified method 2 stopped decreasing the percentage at 500 times while modified method3 was at 1500 times.

For method 3, we realised that the percentage of the total edges seems not enough to hide community in this case. Although Ji and Fm decreased, the percentages are still too high to reach the aim. As a result in the next section, we are going to introduce modified method 3.

In method 3, due to the lack of adding edges, Ji and Fm were up to 100% that means the original community is almost the same as the community after running hiding methods. Since the number of adding random edges depends on the total edges in the network, there is a doubt if using total edges in the network is a right way to hide community.

After the analysis of hiding methods in the American College football network, we are going to compare hiding results. Because there is no percentage in method 2 and

method 3 which meet the hiding target that Ji is less than 65% and Fm is less than 70%, such methods cannot compare with other methods.

Method	(Percentage, Range)	(Ji, Fm)	size	note
0	(0.5,5)	(49.12%, 70.05%)	46	
1	(0.1)	(41.5%, 61.9%)	22	
2				Did not reach the target
3				Did not reach the target
Modified method 2		(22.75%, 39.16%)	150	
Modified method 3		(11.80%, 27.34%)	1500	

Table2. The comparison between hiding methods for American College football in OSLOM

Referring to the hiding result, modified method 3 is the most excellent one. However, the number of adding edges is too many and wastes much time and space. Compared to method 3, method 1 is more efficient and powerful. It both reached the hiding goal and removed at least number of edges in networks. Consequently, in American College football, the performance from the best to the worst is: method 1 > method 0 > modified method 2 > modified method 3.

C. Neural network

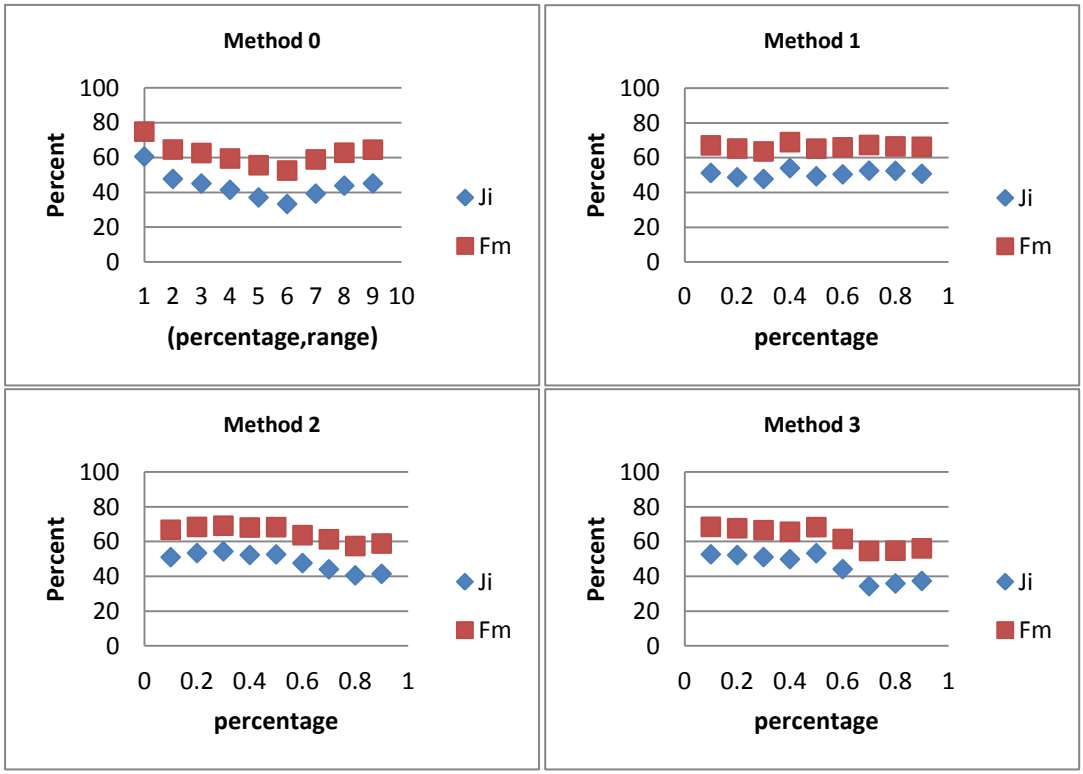


Figure38. The hiding results of each method in neural network in OSLOM

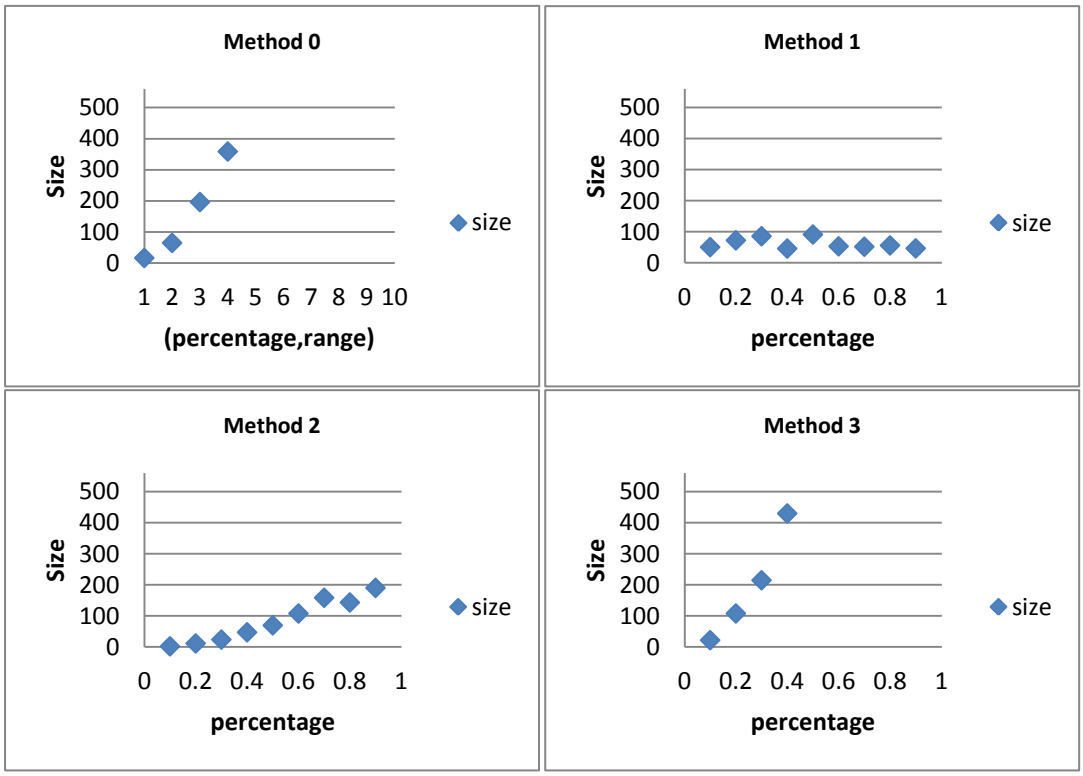


Figure39. The size of each method in neural network in OSLOM

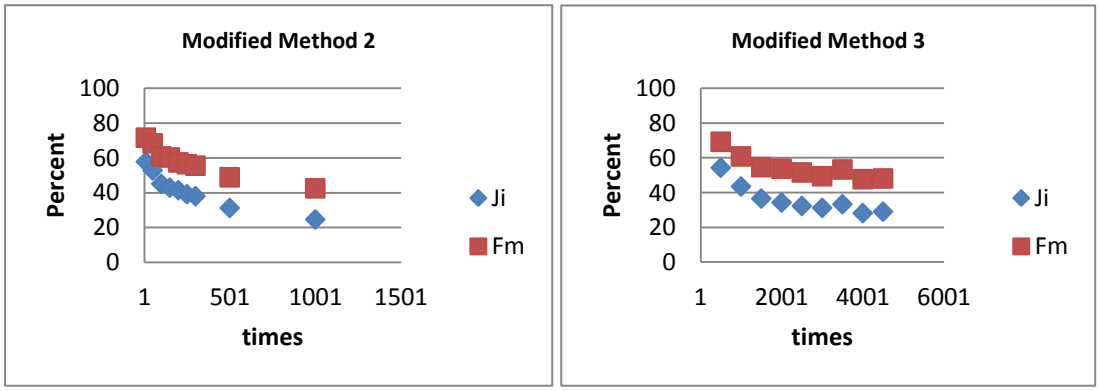


Figure40. The hiding results of modified method 2 and method 3 in neural network in OSLOM

From figure 38, in method 0, there is a fluctuation between (0.4, 4) and (0.9, 9). One of the reasons is as we mentioned above, OSLOM is a community detection algorithm that the network will be divide into different community structure each time. If the network is small, the influence is subtle. On the contrary, if the network is huge, OSLOM will have a profound impact on the classification of the network. Therefore, the results in this case were affected even if we got balance value. The other reason is the complication of network. Since the size of network has become larger, the connection in networks is more complex. For example, the nodes we link with might build several edges with other nodes we connect to as well. Hence, the more edges to be added have the higher possibility to the worse result. (See Fig35, Fig37. Fig38 and Fig40),

In method 1, the result is similar to implement in the Zachary's karate club that had removed all possible edges at percentage 0.1.

In method 2 and method 3, the hiding results are good. Both of them seem to be influenced by the reasons we stated in method 0. In fact, with the exception of modified methods that add thousand times edges than original methods, remaining hiding approaches have fluctuation in a period of percentage.

In modified method 2 and method 3, we found that the differences between their best hiding results and those in method 2 and method 3 are around 10%. Modified methods were much better than originals in this case since original ones cannot improve due to the restriction of their times and ways to connect edges. Therefore, if the size of network becomes bigger, modified method 2 and modified method 3 usually can get the better result than originals in a range of networks size.

Relating to efficiency, compared method 3 with modified method 3, we discovered that, if the percentages of Ji and Fm are the same in both methods (54%, 69%), their size is quite distinct. While method 3 added 22 inter-community edges, modified method 3 added 500 edges to reach the same target. However, even if the method 3 seems to be more effective than modified method 3, method 3 is useless in American College football. Hence, there is no one that is better than other method. It depends on different network.

After the evaluation of each method, we are going to select the most efficient and suitable one for neural network from table3. Even if method 1 hides community more completely than others, its size is 25 times more than method2. Consequently, in neural network, the performance from the best to the worst is: method 2 > method 3 > modified method 2 > method 1 > method 0 > modified method 3. The ranking is created based on the percentage of all hiding methods corresponding to the goal, and then ranked by the size.

Method	(Percentage, Range)	(Ji, Fm)	size
0	(0.2,2)	(47.72%, 64.57%)	64.6
1	(0.1)	(51.2%, 67.05%)	51.13
2	(0.1)	(51.08%, 66.82%)	2.27
3	(0.1)	(52.73%, 69.52%)	22
Modified method 2		(52.79%, 68.31%)	50
Modified method 3		(54.39%, 69.31%)	500

Table3. The comparison between hiding methods in neural network in OSLOM

To summarize, each method has different thresholds in different real network.The threshold is like the best performance that each method can achieve when employing in this network. In other words, the value means the network has reached the saturated situation. In method 0, method 2 and method 3, the threshold happened

because the nodes in other communities have been almost linked. The number of edges is enough to combine the targeted community into other communities. In method 1, the reason is that intra-community edges have been removed by previous nodes already. As a result, there is no edge that can be deleted anymore. In modified methods, their threshold happen in the limitation of the times. For modified method 2, it is 1000, while it is 4500 in modified method 3.After passing the threshold, the percentage of Ji and Fm remain steady and do not have big change to the end or until the next threshold. Each method does not restrict to one threshold. In the neural network, we compared method 2 with modified method 2 because the difference between them is only the size. From figure 38 and figure 40, Ji and Fm are 40.63% and 57.51% at percentage 0.6 in method 2. The threshold for method 2 is 0.6. On the other hand, modified method 2 has almost the same percentages as method 2 from 10 to 300 times. Therefore, the threshold for modified method 2 is 10 times. However, in the range from 500 times to the end, such values unexpectedly fell to around 31% and 40%. In other words, there is a second threshold in modified method 2 since the percentages should be stable after the threshold. Besides, we can infer that the number of threshold and the degree to achieve saturated situation depends on the size of the network since there are more edges that can be removed or added in big network. In neural networks, it is harder to real saturated status than in Zachary's karate club.

If the network is small, there is no substantially different result of these hiding methods. However, when the size of network becomes bigger, the hiding results of all method except for modified method all declined. Sometimes, method 2 and method 3 are useless. If the number of edges in networks are ten times than 1000 in modified method 2, and 4500 in modified method 3. These two methods are no use as well. On the contrary, if the number of edges under these values, the hiding results are more powerful than original ones, especially in the big network.

In method 0, percentage_0 influences more than range.

The times in method 3 indicate the percentage to total edges in networks. However, the total number of edges in networks seems not an efficient way to define the number of adding edges.

If the community detection algorithm is OSLOM, the results in all methods excluding modified methods will have more fluctuation in larger network because of the property of OSLOM and the complication in networks.

In the principle, method 2 and method 3 should be more efficient than modified ones because latter ones always add thousand times than former ones. However, it is not the truth if method 2 and method 3 are useless. Therefore, original methods and modified methods both have valid contributions towards different networks.

In these real networks, method 2 is most useful in Zachary's karate club and neural network and method 1 is the best in American College football.

3. The relationship between the random graphs and hiding methods in OSLOM

In section II.3., we realised that OSLOM cannot use in the random graph. There is a big chance that it will classify the networks into homeless nodes. In other words, each community only has one node. Accordingly, we are going to analyse if different methods have an effect on different types of random graphs, BA, ER, and WS, and if the hiding methods can be adopted in this special division. The testing data of these types are generated from a well-known software called Gephi, which are usually used in networks or graphs. Furthermore, the sizes of the data are all similar. There are 150 nodes and 530 edges, 150 nodes and 530 edges, and 150 nodes and 300 edges in BA, ER and WS.

We did not implement method 1 and method 2 in this section since they are no use in the community that only includes one node as mentioned in the section III. Method 1 removes intra-community edges but there is no such edge in the community. Method 2 adds inter-community edges based on the number of edges in the community but the number of edges is zero. Due to these causes, it is impossible to run the methods in this kind of network.

A. Watts–Strogatz model

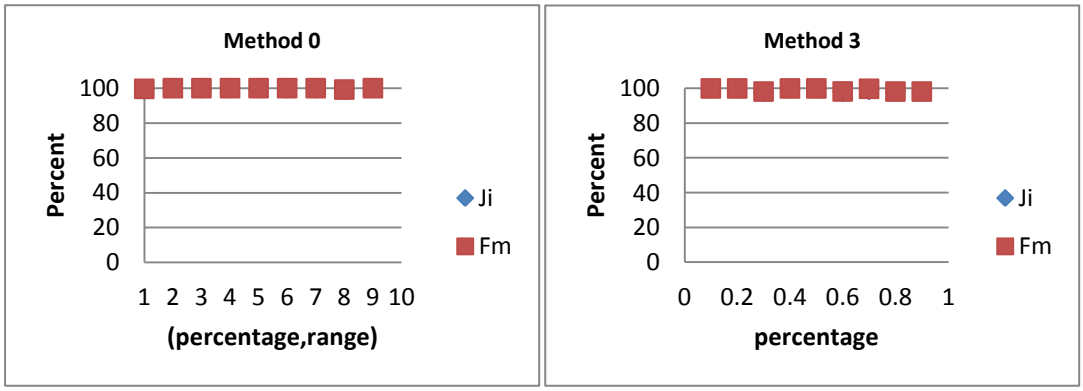


Figure41. The hiding results of method 0 and method 3 in WS model

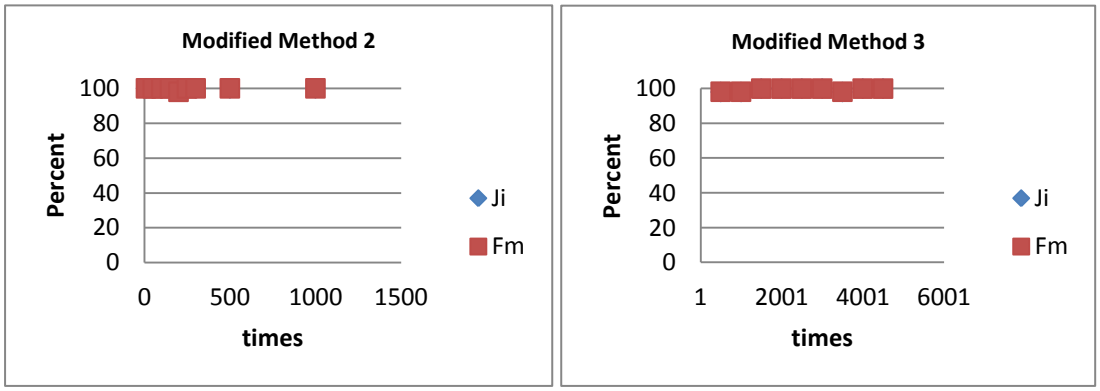


Figure42. The hiding results of modified methods in WS model

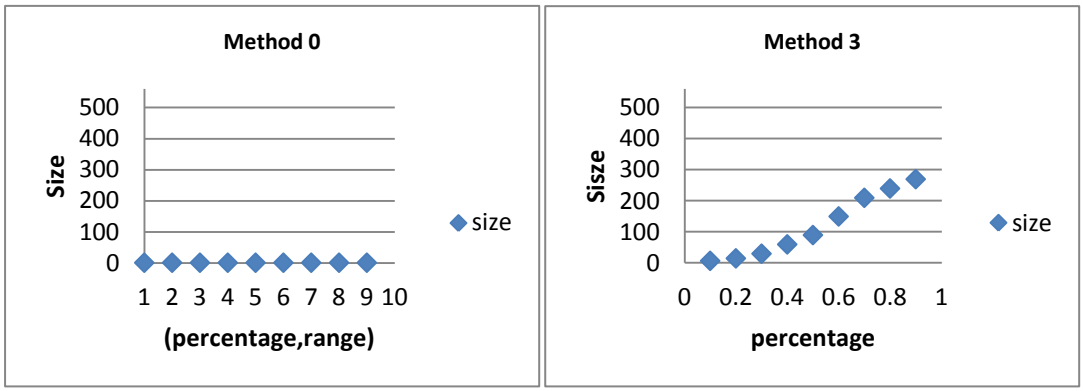


Figure43. The size of method 0 and method 3 in WS model

From figure 41 and 42, through observing the data of method 0 and modified method 2, we discovered that the hiding results are both really worse. In method 0 and modified method 2, Ji and Fm are almost 100% every time. The reason for method 0 is that the node in the targeted community only linked one edge with the node in the other community. (See Fig43) The number of adding edges is always 1. Modified method 2 only adds edges built from the node in the targeted community to those in other communities. This limitation might be the reason because the results of method 3 and modified method 3, which add edges randomly without any rule, changed. The remaining hiding methods are method 3 and modified method 3. There are both subtle fluctuations of Ji and Fm in these methods. However, method 3 vibrated two times more than modified method 3. Moreover, if the percentage is 98% and 98% in both methods, the size of method 3 is 7 while it is 500 in modified method 3. As a result, the method 3 is the best in this case even though it did not achieve the goal. Besides, there is a notable point that even if modified method 3 added around 4500 edges, there was no big difference in hiding results. We guess it because the community structure of WS model is too hard to break.

B. Erdős–Rényi model

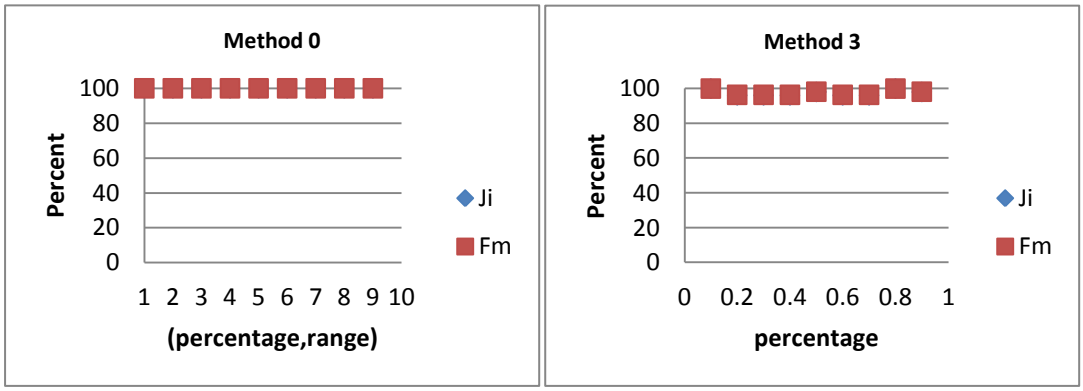


Figure44. The hiding results of method 0 and method 3 in ER model

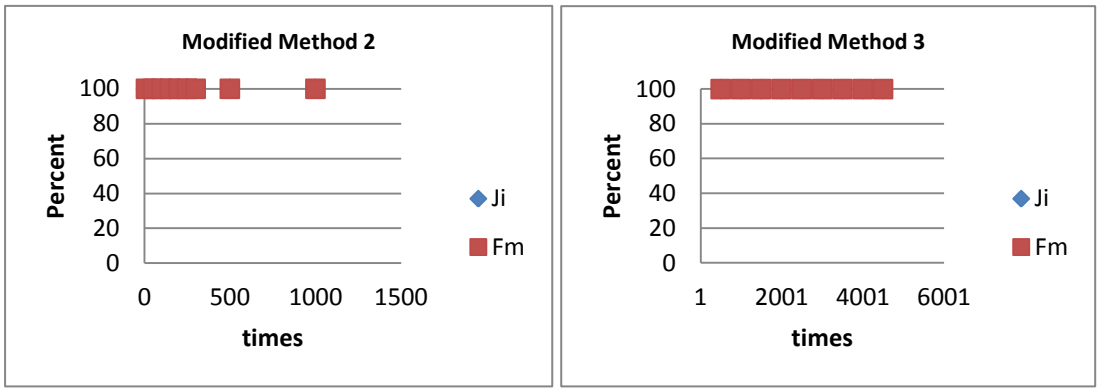


Figure45. The hiding results of modified methods in ER model

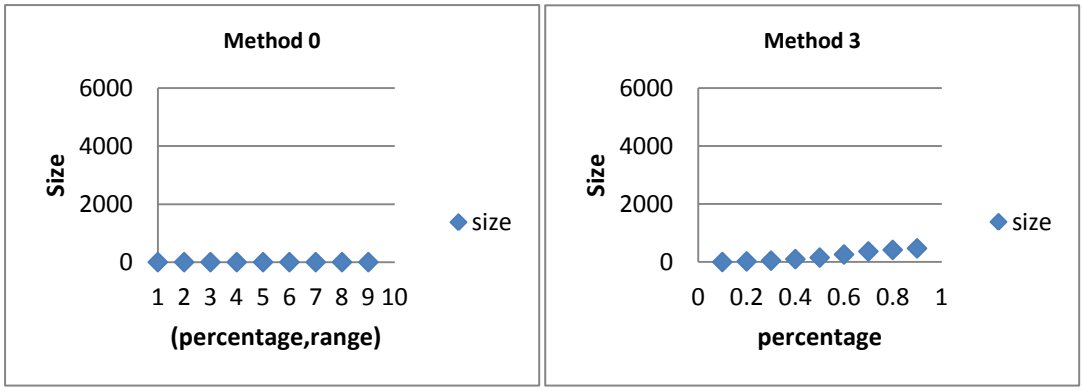


Figure46. The size of method 0 and method 3 in ER model

From figure 44 and figure 45, we discovered that method 3 is the most powerful because Ji and Fm sometimes decreased more than other methods with no change of percentage. Even though method 3 added 4500 edges, the result remained steady as WS model.

C. Barabási–Albert model

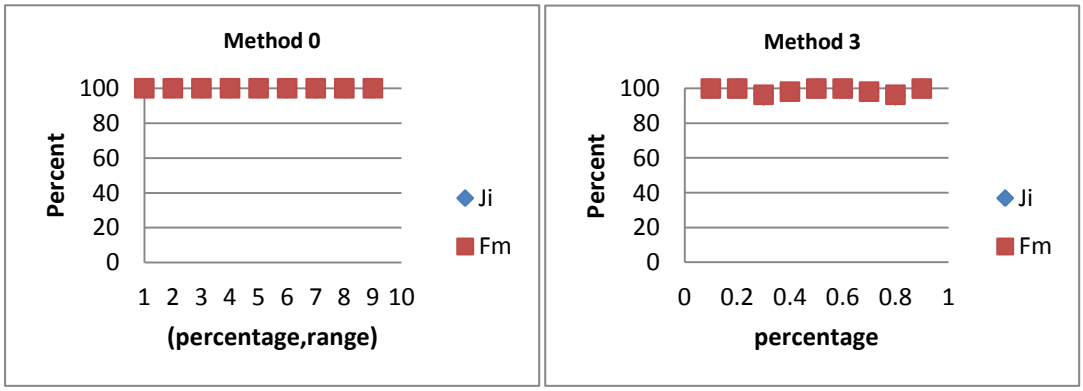


Figure47. The hiding results of method 0 and method 3 in BA model

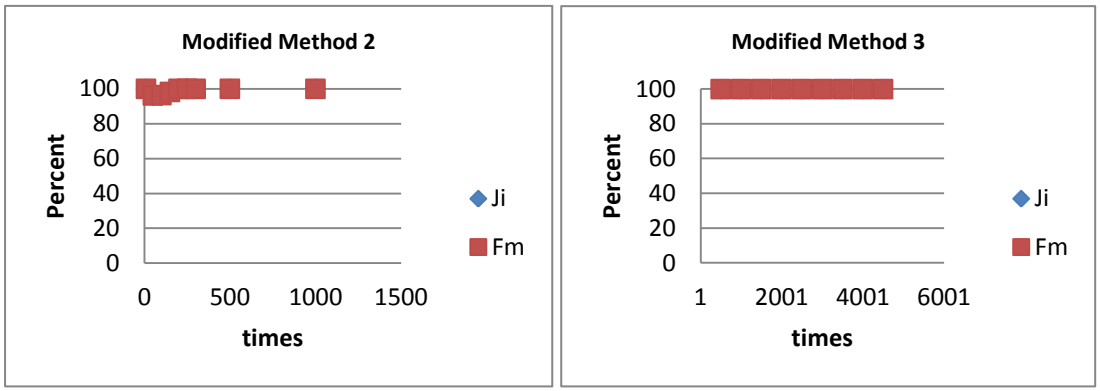


Figure48. The hiding results of modified methods in BA model

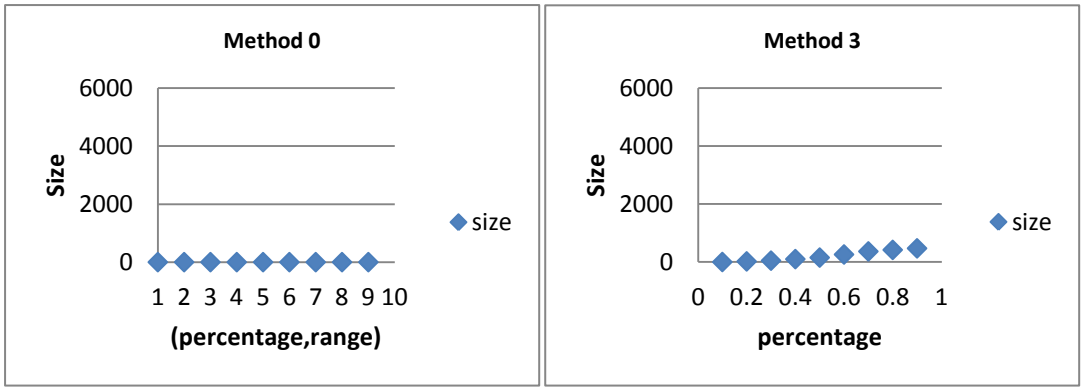


Figure49. The size of method 0 and method 3 in BA model

From figure 49, in modified method 2, we could find some drops of Ji and Fm around times 50 although it recovered to 100% after the times. In method 3, there is a slight fluctuation as ER model and WS model. Ji an Fm in method 3 alter one more time than modified method 2. However, the result of modified method 2 is better that method 3. Therefore, in BA model, it is hard to say which method is the best.

To summarize, method 0 is no use in all models. It points out that method 0 is not suitable for the size of both the targeted community and the pervious community equal to 1. Furthermore, method 3 is the most powerful one in almost WS, ER and BA model. Compared method 3 to modified method 3, even though the number of adding edges is much fewer than in modified method 3, the result of method 3 is still better than modified 3. Hence, we can speculate that, adding fewer edges is more efficient than adding more edges in WS, ER and BA model. Besides, the only difference between these three models is that modified method 2 is useful in BA model. In ER and WS model, even if modified method 2 added around 2000 intra-community edges, the results remained steady. It represents that even if the node in the targeted community links with all the rest of nodes in the network, the density is still no enough to hide the community. Finally, we guessed this only difference is because BA model has a slightly different structure from ER and WS model.

4. The relationship between the real networks and hiding methods in infomap

In section II.3., we understood the formation of infomap, and OSLOM. Therefore, in this section, we are going to use infomap as a community detection algorithm to test the relationship between the size of real networks, and then compare the results with OSLOM.

A. Zachary's karate club

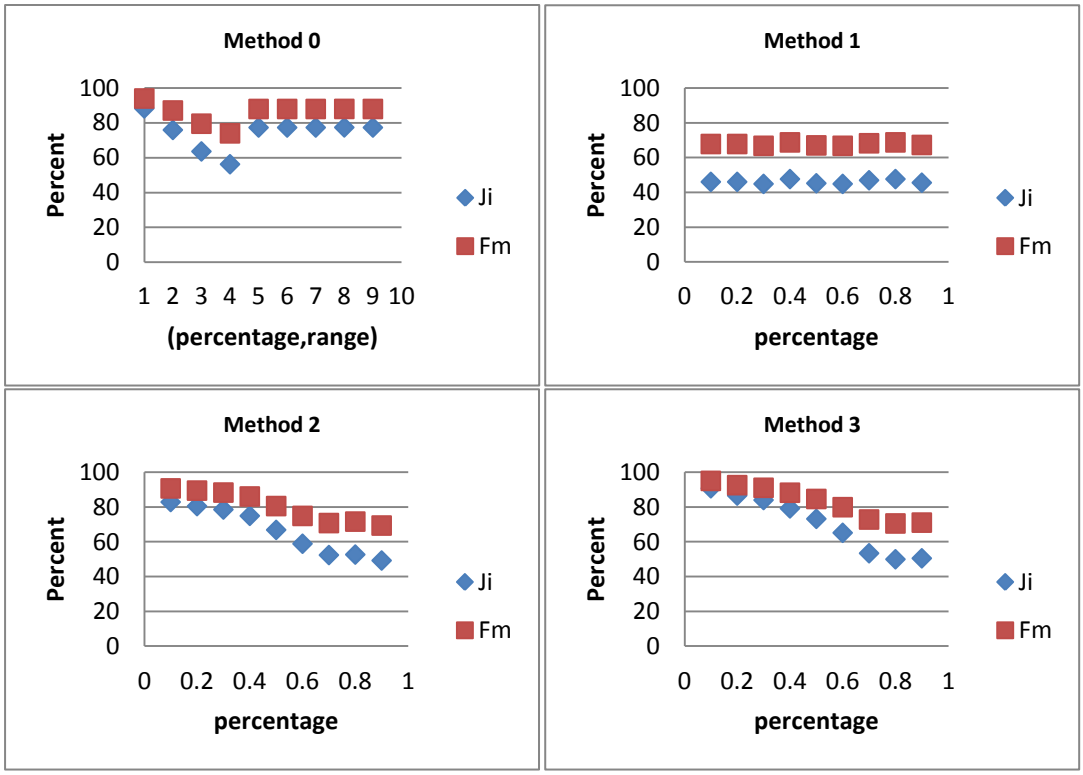
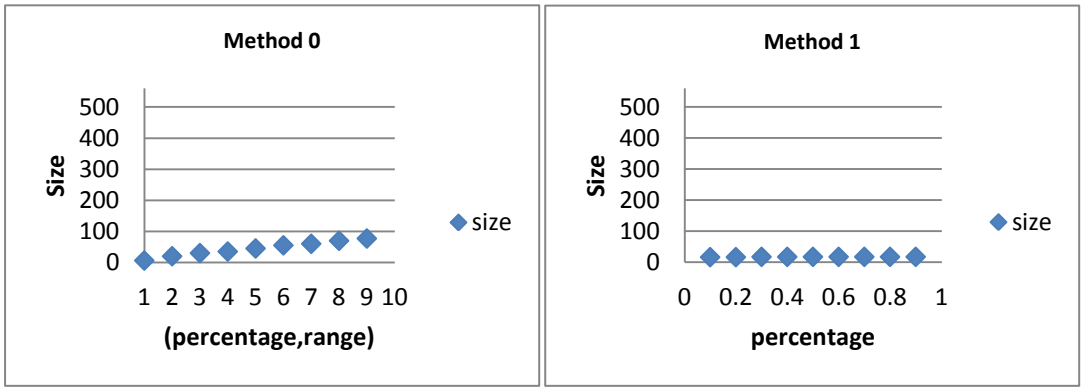


Figure50. The hiding results of each method in Zachary's karate club in infomap



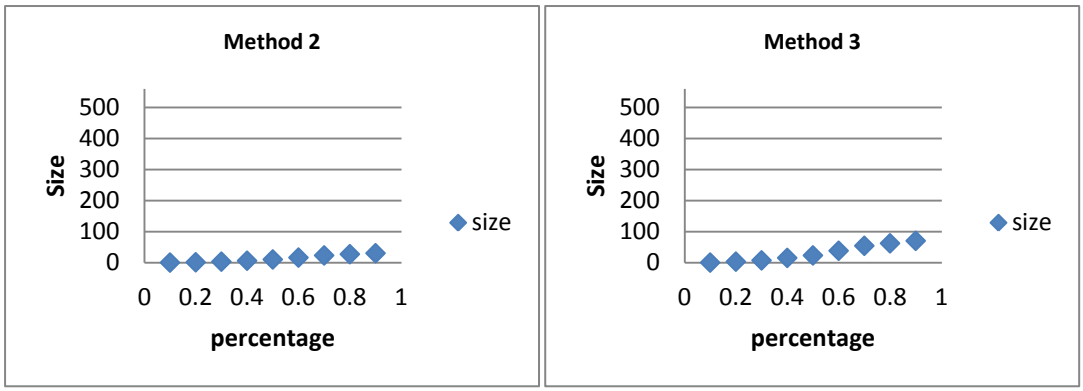


Figure51. The size of each method in Zachary's karate club in infomap

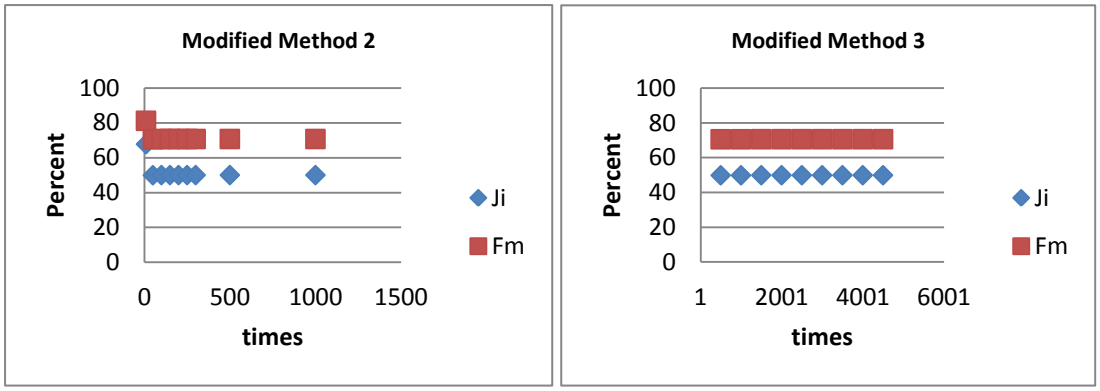


Figure52. The hiding results of modified methods in Zachary's karate club in infomap

From figure50, in method 0, there is no hiding result that meets the aim. On the contrary, the percentage of (Ji, Fm) in method 1 satisfied the goal. Furthermore, the results of Method 2 and method 3 are very similar since the size of network is small. The number of adding intra-community edges and adding all kinds of edges does not have major difference. (See Fig51)

Following is the table that measures the efficiency and the results of each method. From this table, we can rank methods based on the performance from the best to the worst: Method 1 > Method 2 > Method 0 > Method 3 > Modified method 2 > Modified method 3.

Method	(Percentage, Range)	(Ji, Fm)	size
0	(0.4, 4)	(56. 20%, 73.94%)	35
1	(0.1,1)	(46.13%, 67.90%)	17
2	(0.9,9)	(49.23%, 69.32%)	31
3	(0.1,1)	(49.94%, 70.54%)	63
Modified method 2		(49.92%, 70.56%)	50
Modified method 3		(49.94%, 70.67%)	500

Table4. The comparison between hiding methods for Zachary's karate club in infomap

In general, the best hiding result of each hiding method in OSLOM is approximately four times than in infomap. However, the size of each method in OSLOM is usually worse than in infomap. To sum up, through analysing table1 and

table4, we can found that, method 0 and method 1 in infomap are better than OSLOM due to the fewer number of size. On the contrary, remaining methods in OSLOM are more efficient than in infomap. Modified methods are both better used in OSLOM than infomap assuming the number of adding edges is the same. Besides, the method 1 is the most powerful and effective one in infomap, while it is method 2 in OSLOM.

B. American College football

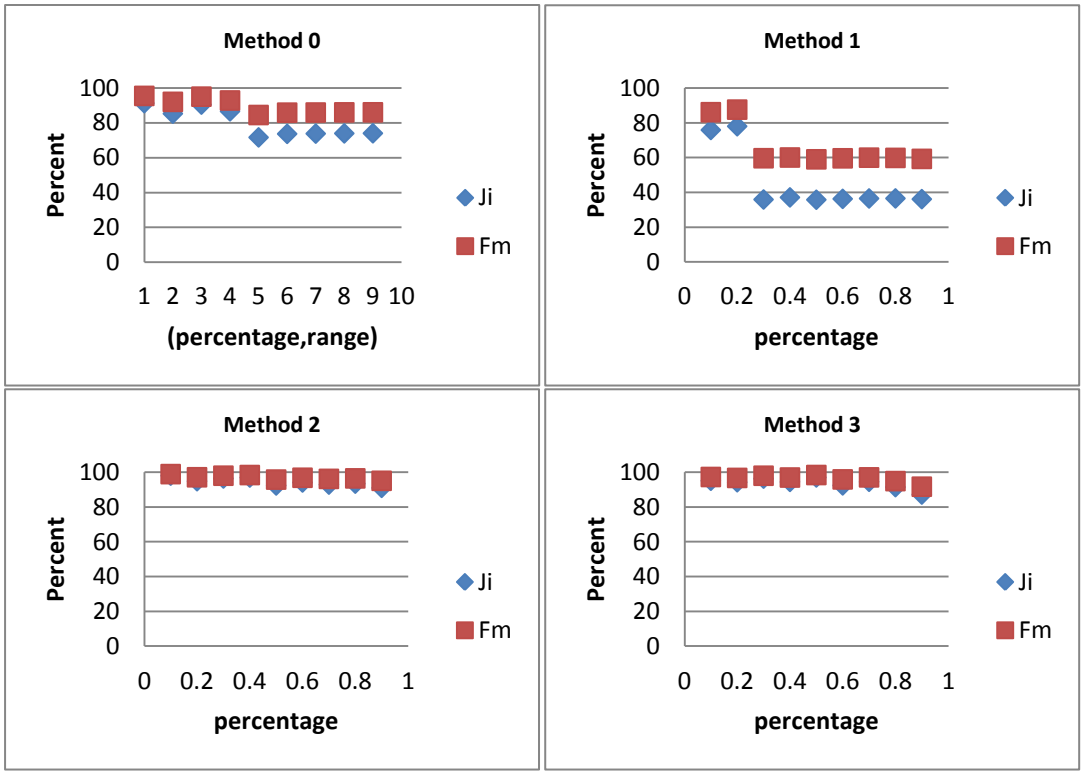
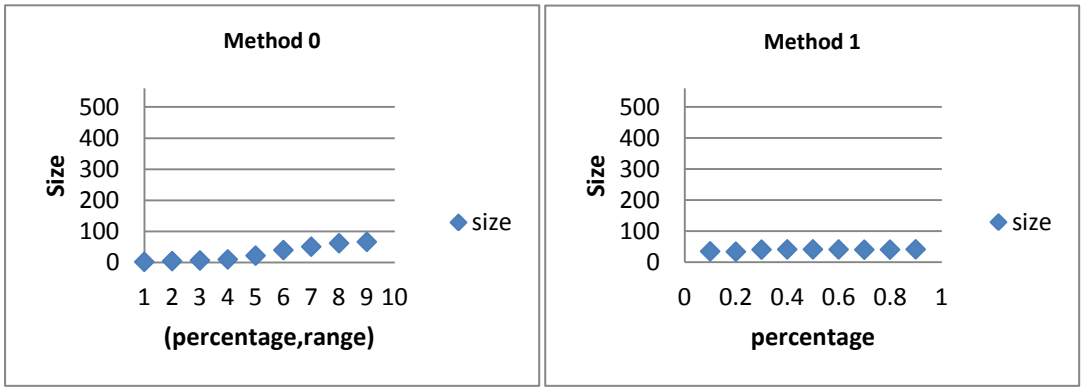


Figure53. The hiding results of each method in American College football in infomap



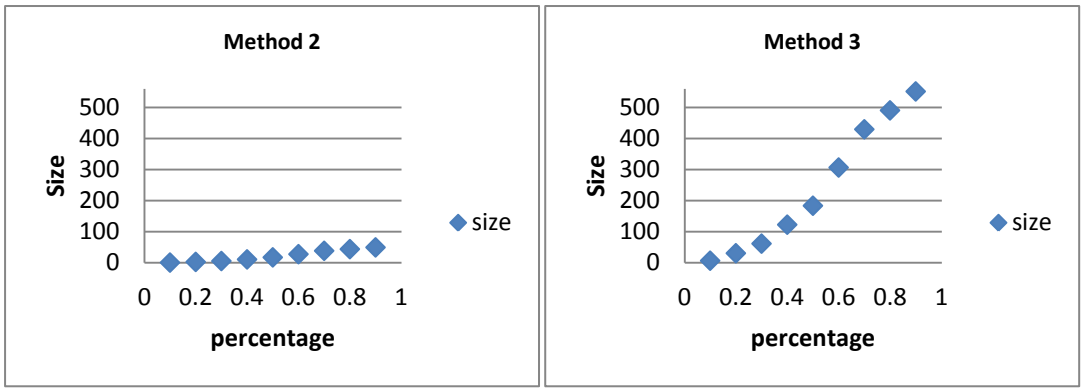


Figure54. The size of each method in American College football in infomap

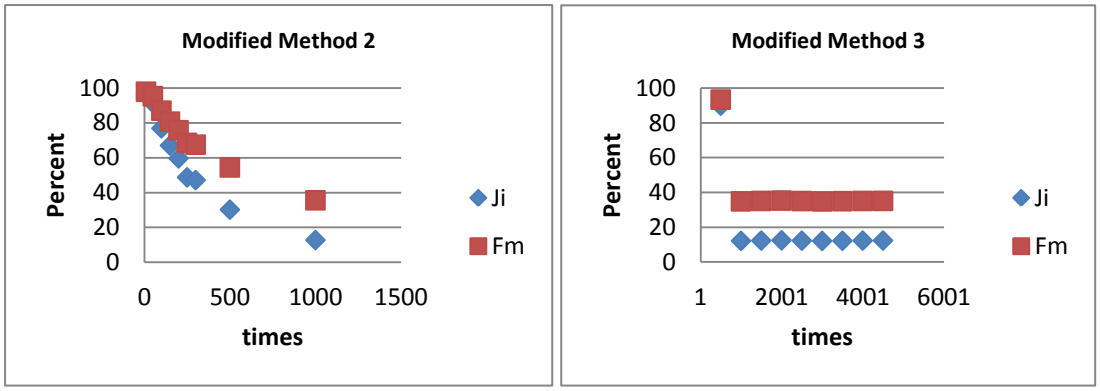


Figure55. The hiding results of modified methods in American College football in infomap

From figure 53, only method 1 reached the goal of hiding results because the percentage (Ji, Fm) were too high in the rest methods excluding modified methods. Besides, from figure 54, we could see that method 1 did not remove as many as numbers of edges as method 0, method 2 or 3. Therefore, method 1 has reached its best performance in this case. The ranking from the beast to the worst is: Method 1 > Modified method 2 > Modified method 3

Method	(Percentage, Range)	(Ji, Fm)	size	Note
0				Did not reach target
1	(0.3)	(36%, 59.72%)	41.39	
2				Did not reach target
3				Did not reach target
Modified method 2		(48.71%, 68.47%)	250	
Modified method 3		(12.23%, 34.96%)	1000	

Table5. The comparison between hiding methods for American College football in infomap

Through comparing figure 53 with figure 33, we found that method 0 was useless in infomap but useful in OSLOM. Neither method 2 nor method 3 is useful in both community detection algorithms. Moreover, relating to method 1, infomap have to remove more edges than OSLOM in this case, so method 1 in OSLOM is better than in

infomap. Besides, assuming the same number of adding edges, the hiding results of modified methods in OSLOM are still better than in infomap. Finally, method 1 is the most suitable one in the both community detection algorithms.

C. Neural network

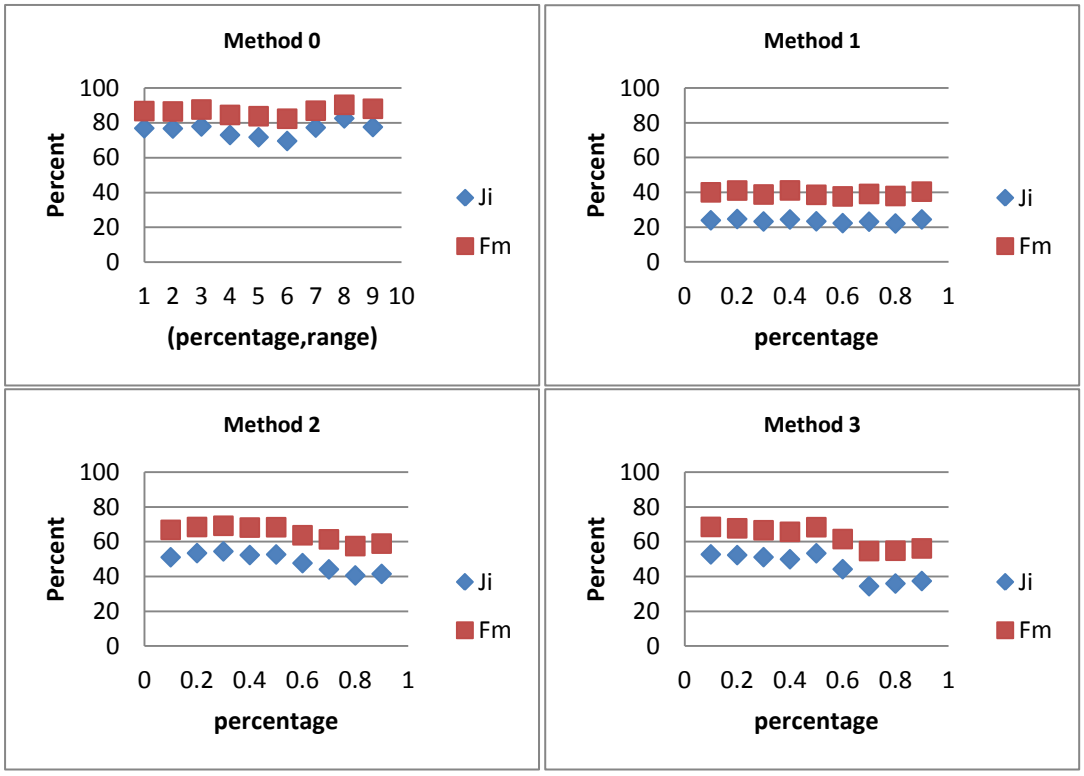


Figure56. The hiding results of each method in neural network in infomap

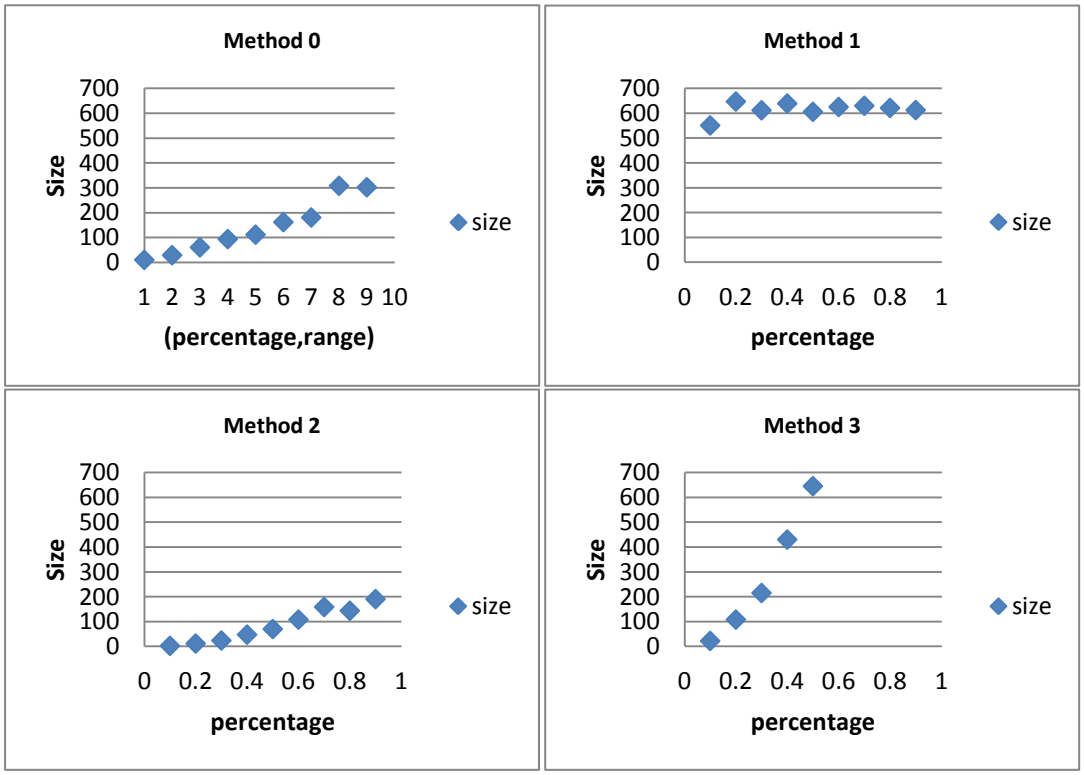


Figure57. The size of each method in neural network in infomap

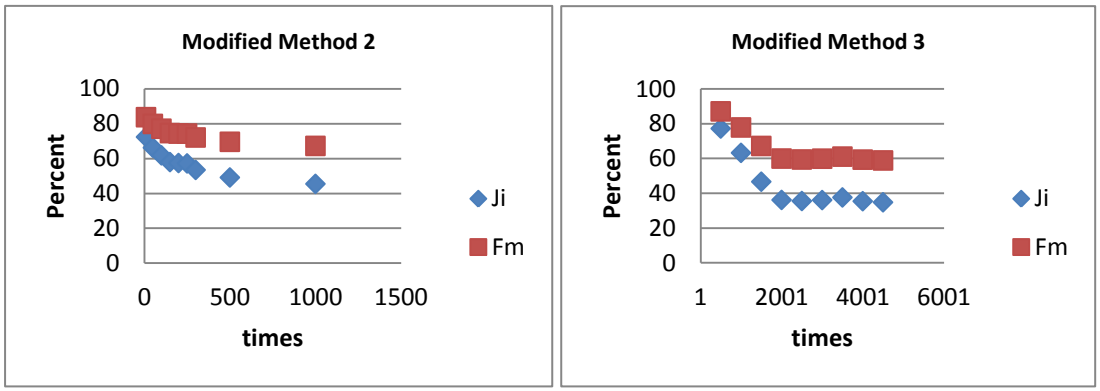


Figure58. The hiding results of modified methods in neural network in infomap

From figure 56, there is an interesting point that except for method 0, the remaining methods are all useful compared to the American football case we mentioned earlier. In that case, only method 1 can do. Following is the table describing the best result for each method. The performance from the best to the worst: Method 2 > Method 3 > Modified method 2 > Modified method 3.

Method	(Percentage, Range)	(Ji, Fm)	size	Note
0				Did not meet the target
1	(0.1)	(24.02%, 40.59%)	551	
2	(0.1)	(51.37%, 67.02%)	2.27	
3	(0.3)	(52.73%, 68.52%)	22	
Modified method 2		(49.10%, 69.58%)	500	
Modified method 3		(46.75%, 67.3%)	1500	

Table6. The comparison between hiding methods for neural network in infomap

Method 0 is useless in infomap while it useful in OSLOM. In method 1, the hiding result in infomap is powerful than in OSLOM but the number of removing edges is almost six times than in OSLOM. Therefore, method 1 in OSLOM is better than infomap. If the number of adding edges are the same, method 2 and method 3 in OSLOM are slightly better than in infomap. Referring modified methods, the finding results are better in infomap than in OSLOM. Method 2 is the most effective one in both community detection algorithms.

To summarize, hiding results of most of the methods in OSLOM are better than in infomap. On the contrary, there is no obvious relationship between community detection algorithms and the size. In other words, in the same case, the size of one method may be higher in OSLOM than in infomap but the other method may be lower in OSLOM than in infomap. Moreover, if a method is the most suitable one in a case in OSLOM, it cannot demonstrate such method is also the most proper one in the same case in infomap. Besides, under the same assumption that the size is the same, OSLOM is usually better than infomap in the most of the cases. Therefore, we infer that the community structure in infomap is harder and more complete than in OSLOM. (Discussed in the next section)

5. The relationship between the random graphs and hiding methods in infomap

The experimental data is the same as VI.3. It is notable that there must be significant differences between the OSLOM and infomap since OSLOM classified the networks into many homeless nodes. Therefore, the methods that we cannot use in OSLOM can be applied to here. Besides, it is a good example to compare if different kinds of networks have an impact on the results of hiding approaches. In this section, we are going to show the table and figure of hiding methods, and then analyse if there is the unexpected data in each hiding methods. Finally, we will do comparison with the results of real network in infomap, and random graphs in OSLOM.

A. WS model

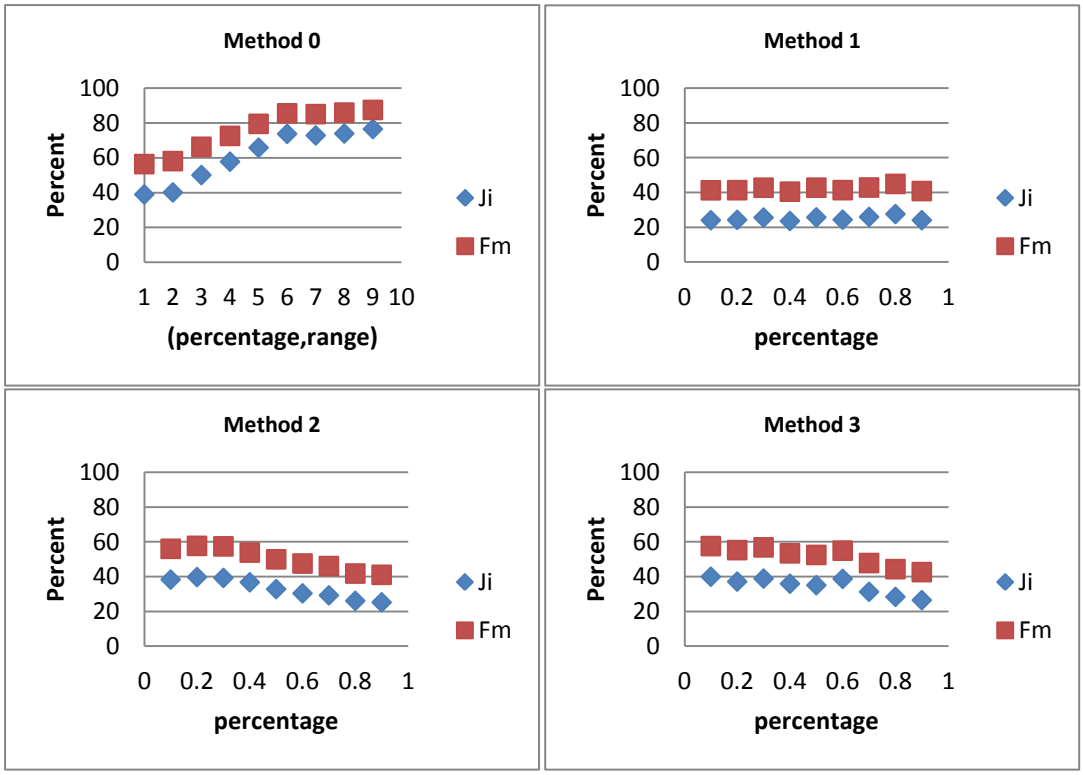
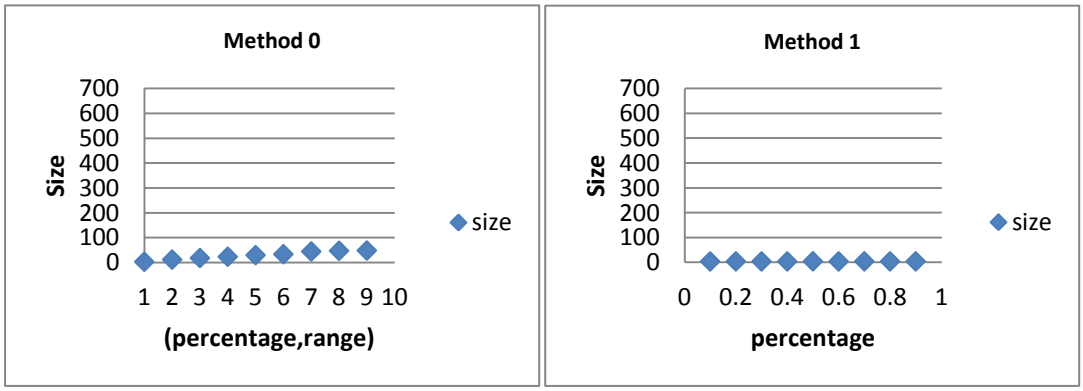


Figure59. The hiding results of each method in WS model in infomap



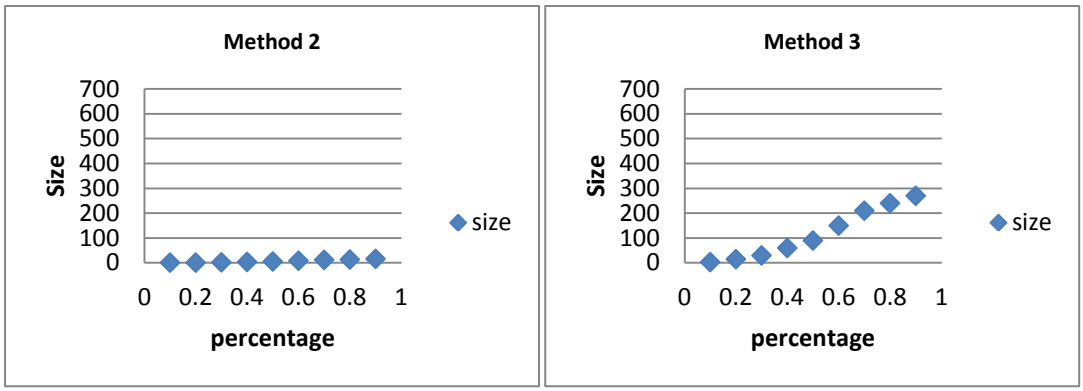


Figure60. The size of each method in WS model in infomap

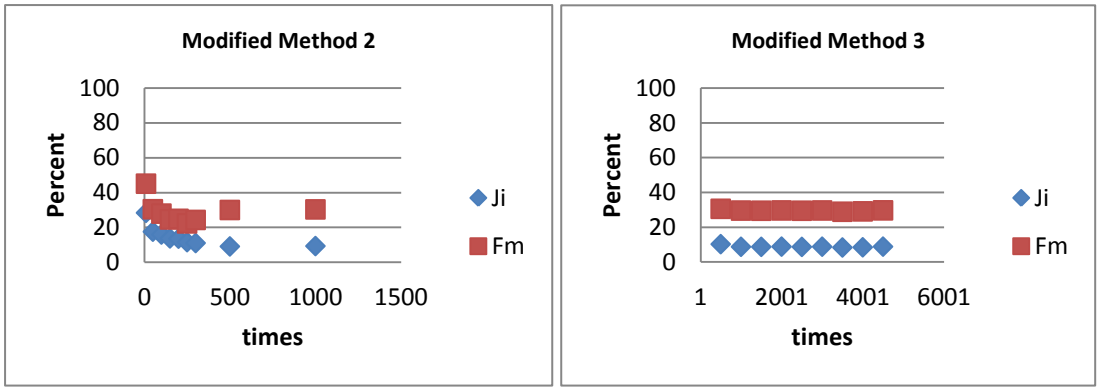


Figure61. The hiding results of modified methods in WS model in infomap

From figure60, we found there is a weird trend that the percentage became higher with the increasing percentage and range. Although it sometimes actually happens in other methods as well, it is the first time so obvious. The cause of this might be too many connections to the other community as we mentioned in VI.2.B. Due to the complicated connections in that community, the targeted community connect densely accidentally. However, this phenomenon seldom happened in previous cases.

Following is the table that presents their performance. From this table, the rank of the performance from the best to the worst: Method 2 > Method 0 > Method 3 > Method 1 > Modified method 2 > Modified method 3

Method	(Percentage, Range)	(Ji, Fm)	size
0	(1, 0.1)	(38.89%, 56.32%)	3
1	(0.1)	(24.11%, 41.34%)	4.3
2	(0.1)	(38.30%, 56%)	1
3	(0.1)	(39.95%, 57.57%)	3
Modified method 2		(28.33%, 45.07%)	10
Modified method 3		(10.37%, 30.69%)	500

Table7. The comparison between hiding methods for WS model in infomap

B. ER model

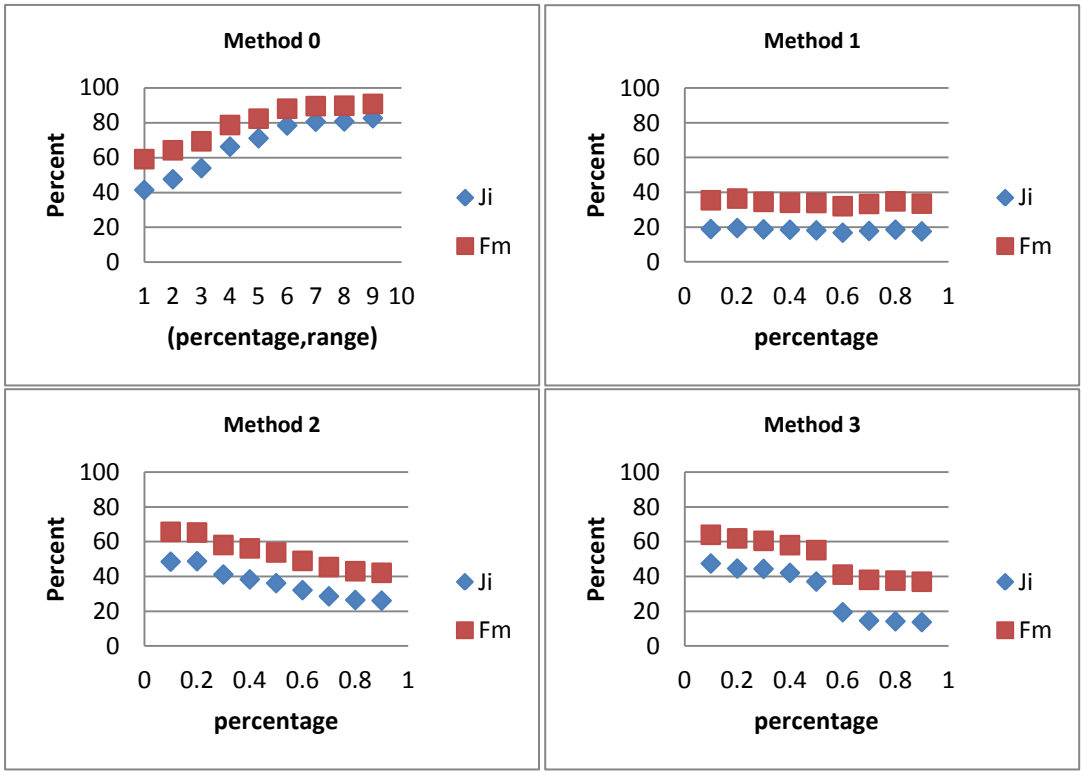


Figure62. The hiding results of each method in ER model in infomap

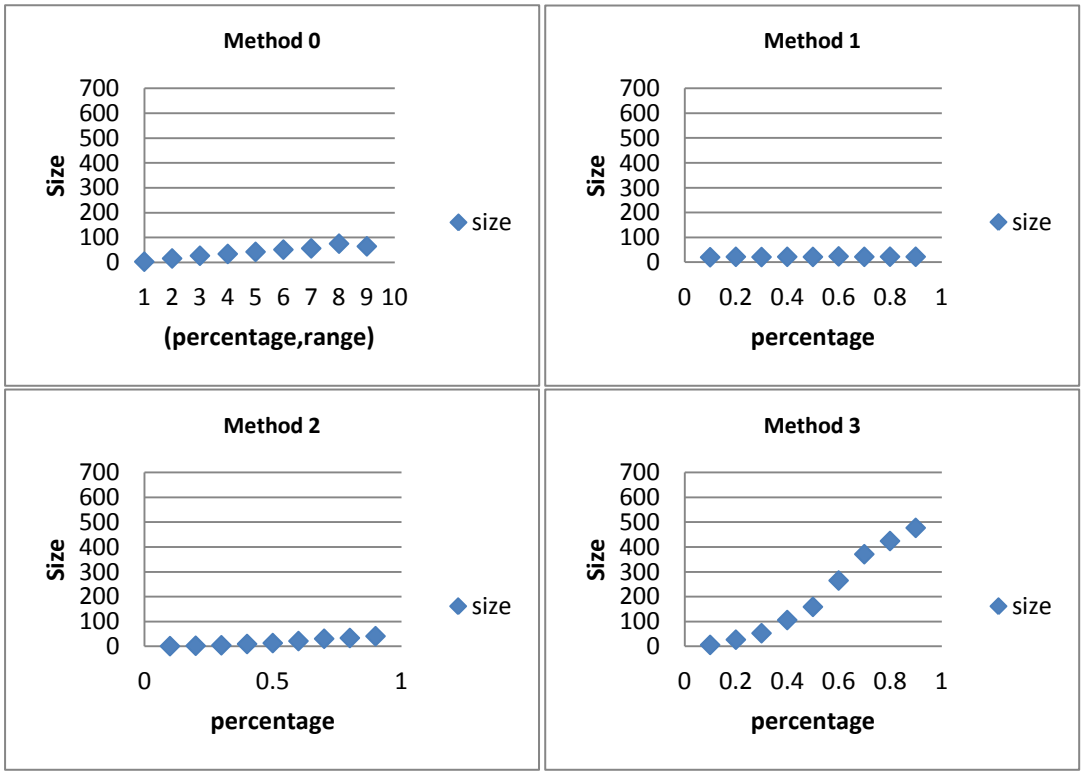


Figure63. The size of each method in ER model in infomap

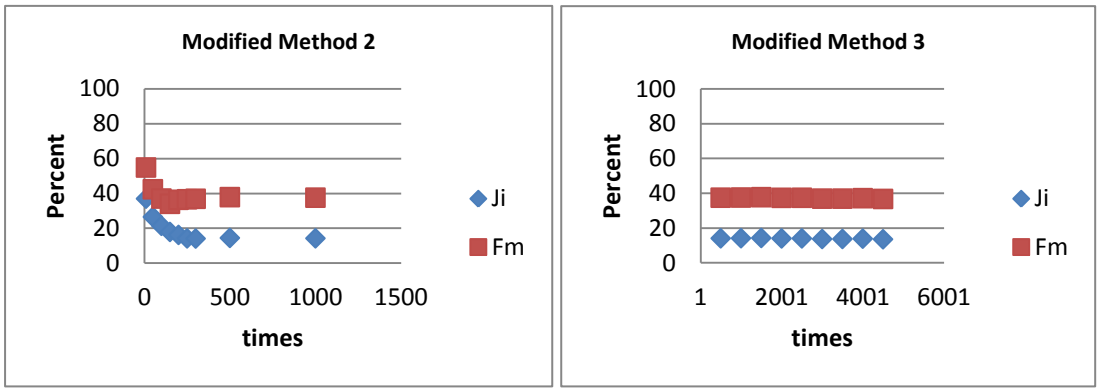


Figure64. The hiding results of modified methods in ER model in infomap

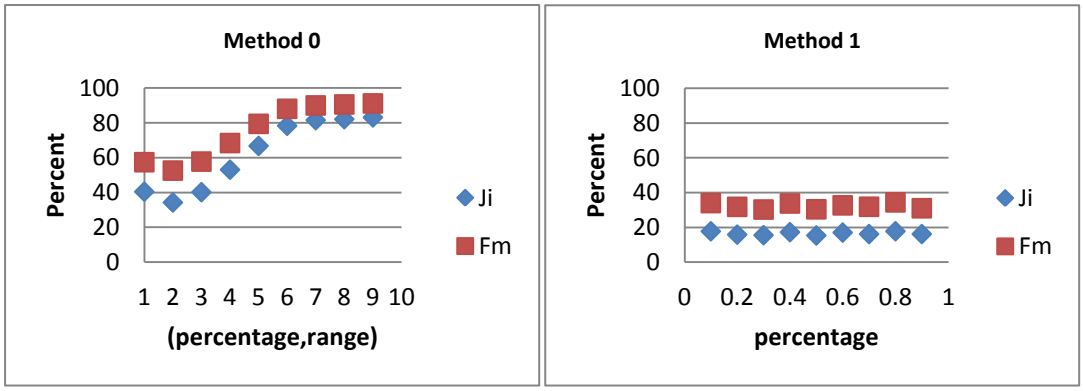
From figure 63, in method 0, the trend is just like WS model. Therefore, we speculated that it is influenced by the community detection algorithm of infomap. This has been proved in the next section.

Following is the compared table. The rank of performance: method 2 > method 0 > method 3> modified method 2 > method 1> modified method 3.

Method	(Percentage, Range)	(Ji, Fm)	size
0	(1, 0.1)	(41.35%, 59.08%)	2.73
1	(0.1)	(18.94%, 36.52%)	20.65
2	(0.1)	(48.50%, 65.67%)	1
3	(0.1)	(47.46%, 63.99%)	6
Modified method 2		(36.94%, 54.83%)	10
Modified method 3		(14.2%, 37.56%)	500

Table8. The comparison between hiding methods for ER model in infomap

C. BA model



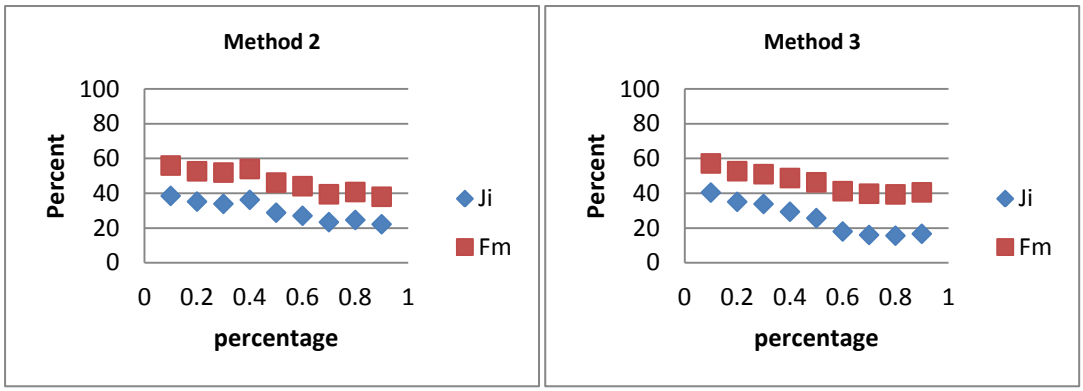


Figure65. The hiding results of each method in BA model in infomap

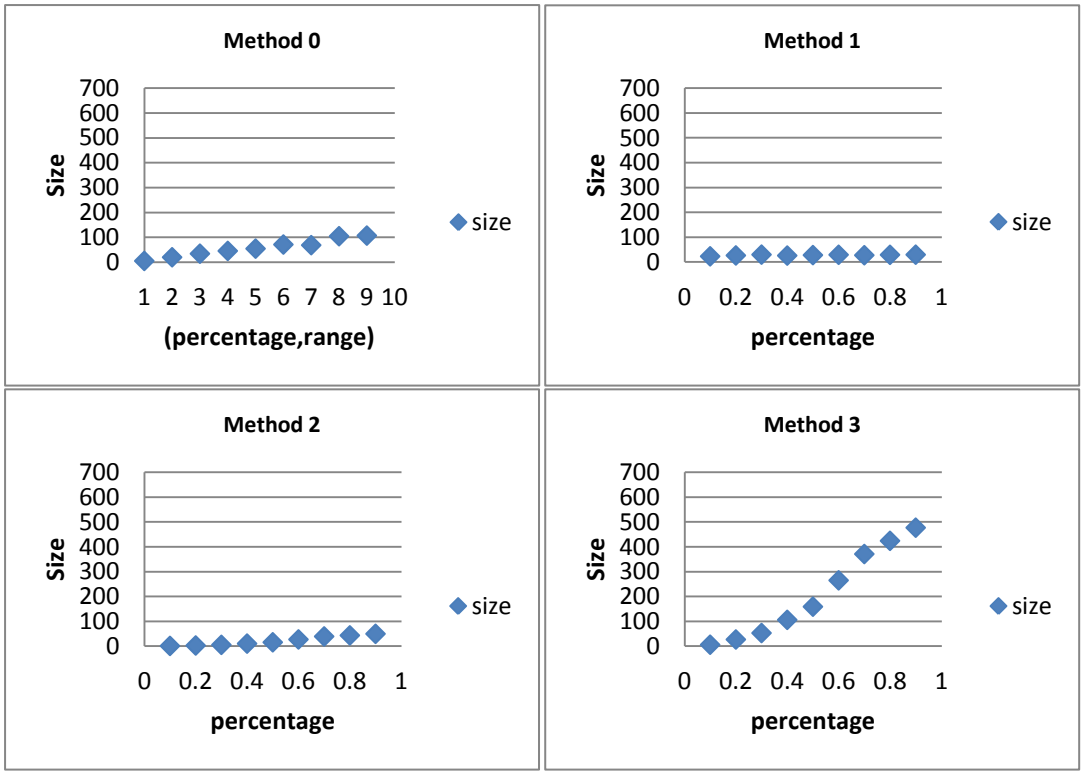


Figure66. The size of each method in BA model in infomap

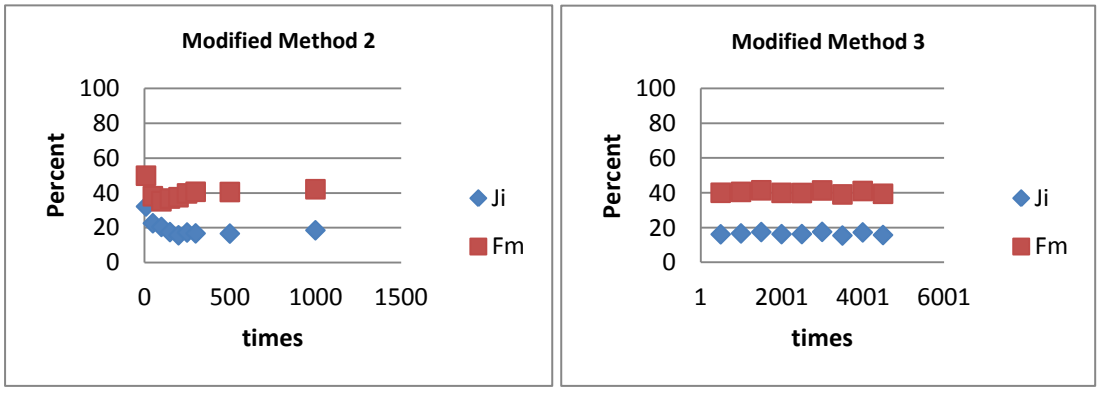


Figure67. The hiding results of modified methods in BA model in infomap

Through observing three random graphs, we think this trend is caused by infomap, since this did not occur in real networks.

Following is the table about the performance, and the rank of these hiding methods in

BA model is: method 2 > method 0 > method 3 > modified method 2 > method 1 > modified method 3

Method	(Percentage, Range)	(Ji, Fm)	size
0	(1, 0.1)	(40.43%, 57.377%)	5
1	(0.1)	(17.73%, 33.95%)	23.6
2	(0.1)	(38.53%, 55.97%)	1.13
3	(0.1)	(40.43%, 57.17%)	6
Modified method 2		(32.14%, 49.82%)	10
Modified method 3		(16.22%, 40.09%)	500

Table9. The comparison between hiding methods for ER model in infomap

The purpose that we take comparison in the end is that there are slight differences between these three models in infomap. Actually, in OSLOM, the results of these methods are almost the same because of the homeless nodes. As we described in section II.I1.B., the different formation of each random graph might have an effect on hiding results. BA model seems to be more effective than ER and WS model in infomap in general. We infer the reason is that BA model is a scale-free network. Most of the degrees of nodes are few. Thus, if a node with high degree, it means that the node is really important in networks. Hence, if our hiding method changes the edges around the node, the network will be affected drastically. This is the fragile point in this network.

Method 0, method 3, modified methods seem to be more efficient in infomap than OSLOM since they were nearly no use in OSLOM except for method 3. The methods that are most affected are method 2, method 3 and modified methods. These methods can easily achieve the target by adding few edges in infomap while they cannot even if the total edges they added is more than thousands times in OSLOM. Moreover, as the guess we mentioned earlier that we infer the community structure in real networks in infomap is more complete and well-defined, it is contrary in random graphs because hiding methods is more useful in infomap. One of the reasons for these different results might be that the community structure of different community detection algorithms are affected by the different types of networks. The other might be that hiding approaches are related to different types of networks.

There is no hiding method that is the most appropriate in random graphs in infomap. But we can ensure that method 2 and method 0 often have better results than other methods in this size and types of random graph in infomap.

In comparison with real network in infomap, since the size of American College football is the most similar one to the size of these random graphs modes, we take American College football as a criterion to compare. In method 0, it is no use in American College football even if we added 66 edges. On the other hand, the results

could be reached by adding 3 edges in WS model. In method 1, the hiding results in WS model are better than American College football. In method 2 and method 3, the percentage of Ji and Fm decreased gradually if we added more number of edges in American College football. Compared to them, adding edges in method 2 and method 3 in WS model cannot contribute to decrease the percentage of Ji and Fm. In modified methods, WS model is better than American College football at the beginning. To the end of such methods, their results are the same. In brief, we can speculate that applying hiding approaches in the random graphs might be more helpful and efficient than in real networks.

V. Conclusion

In this project, we devised four efficient hiding methods and two modified methods to hide community in undirected network. We have demonstrated that different types of networks have an influence on the hiding results. We used random graph and real networks as the testing data and then applied to OSLOM and infomap community detection algorithms. In OSLOM, all the hiding results of real network in each method met the target while none of hiding results of random networks in each method did. In addition, different types of random graphs seem to have an effect on hiding methods. Besides, there is no method that is the best in each size of networks since method 2 is the most suitable in Zachary's karate club while it is no use in American College football. Moreover, the hiding results are affected by the size of network. If the network size become bigger, the hiding results will get worse. Furthermore, the different community detection algorithms impact drastically on the hiding results. The hiding results of OSLOM are better than that of infomap in general. Finally, implying different hiding results definitely influence the hiding methods. Some methods are useless while other methods are quite useful.

In the project, the dataset is not enough to test the relationship between the different size of networks and hiding methods. Therefore, the results we applied in the small size of networks cannot be the truth in all small size of networks. Besides, we did not test the limitation of our hiding methods. These methods should have a range of the size of networks so that we can use these methods in the range to reach the best performance. Moreover, if we can find a new parameter that defines the number of adding edges in the method 2 and method 3, it will be useful to improve the results. Finally, we hope these problems can be solved in the future work.

VI. Bibliography

Ahn, Y. Y., Bagrow, J. P., and Lehmann, S., 2010. Link communities reveal multiscale complexity in networks. *Nature* [e-journal] 466, pp.761–764 Available through: http://www.nature.com/nature/journal/v466/n7307/abs/nature09182.html

Albert, R., Jeong, H., and Barabási, A., 2000. Error and attack tolerance of complex networks. *Nature* [e-journal] 406, pp.378-382 Available through: http://www.nature.com/nature/journal/v406/n6794/full/406378a0.html

Altman, R. B., Dunker, A. K., Hunter, L., Lauderdale, K., and Klein, T. E., 2001. A stability based method for discovering structure in clustered data. In: the Pacific Symposium, *Biocomputing 2002* .pp.6-17

Barabasi, A. L., Albert, R., and Jeong, H., 1999. Mean-field theory for scale-free random networks. *Physica A* [e-journal] 272(1-2), pp.173-187 Available through: http://dx.doi.org/10.1016/S0378-4371(99)00291-5

Bae, E., Bailey, J., and Dong, G., 2006. Clustering similarity comparison using density profiles. In: AI'06, *the 19th Australian joint conference on Artificial Intelligence: advances in Artificial Intelligence*. Berlin, Germany 2006.

Beygelzimer, A., Grinstein, G., Linsker, R., and Rish, I., 2005. Improving network robustness by edge modification. *Physica A: Statistical Mechanics and its Applications* [e-journal] 357(3-4), pp.593-612 Available through: www.sicencedirect.com

Callaway, Duncan S., Newman, M. E. J., Strogatz, Steven H.,and Watts, Duncan J. 2000. Network Robustness and Fragility: Percolation on Random Graphs. *Physical Review Letters* [e-journal] 85(25), pp.5468-5471 Available through: arXiv:cond-mat/0007300v2 [cond-mat.stat-mech]

Crucitti, P., Latora, V., and Marchiori, M., 2004. Model for cascading failures in complex networks. *Phys. Rev. E.* [e-journal] 69(4) Available through: http://link.aps.org/doi/10.1103/PhysRevE.69.045104

Crucitti, P., Latora, V., Marchiori, M., and Rapisarda, A., 2004. Error and Attack Tolerance of Complex Networks. *Physica A: Statistical Mechanics and its Applications* [e-journal] 340(1-3), pp.388-394 Available through: www.sicencedirect.com

Dekker, Anthony H. and Colbert, Bernard D., 2004. Network robustness and graph topology. *Australasian Computer Science Conference* [e-journal] 26, pp.359-368 Available through: http://dl.acm.org/citation.cfm?id=979922.979965

Freeman, L. 1979. Centrality in social networks conceptual clarification. *Social Networks* [e-journal] 1(3), pp.215-239 Available through: doi:10.1016/0378-8733(78)90021-7

Fortunato, S., 2010. Community detection in graphs. *Physics Reports* [e-journal] 486, pp.75-174 Available through: ScienceDirect.com

Fortunato, S. and Castellano, C., 2009. Community structure in graphs. In: R.A. Meyers (Ed.), *Encyclopedia of Complexity and Systems Science.* Berlin, Germany 2009.

Girvan, M. and Newman, M. E. J., 2002. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* [e-journal] 99(12), pp.7821-7826 Available through: http://www.pnas.org

Holme, P., Kim, Beom J., Yoon, Chang N., and Han Seung K., 2002. Attack vulnerability of complex networks. *Phys. Rev. E.* [e-journal] 65(5) Available through: http://link.aps.org/doi/10.1103/PhysRevE.65.056109

Karrer, B., Levina, E., Newman, M. E. J., 2008. Robustness of community structure in networks. *Phys. Rev. E.* [e-journal] 77(4) Available through: http://link.aps.org/doi/10.1103/PhysRevE.77.046119

Lancichinetti, A., Fortunato, S., 2010. Community detection algorithms: a comparative analysis. *Phys. Rev. E.* [e-journal] 80(5) Available through: arXiv:0908.1062v2 [physics.soc-ph]

Lancichinetti, A., Fortunato, S., 2011. Limits of modularity maximization in community detection. *Phys. Rev. E.* [e-journal] 84(6) Available through: arXiv:1107.1155v2 [physics.soc-ph]

Lancichinetti, A., Fortunato, S., and Kertész, J., 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* [e-journal] 11(3) Available through: http://iopscience.iop.org/1367-2630/11/3/033015/

Lancichinetti, A., Fortunato, S., and Radicchi, F., 2008. Benchmark graphs for testing community detection algorithms. *Physical Review* [e-journal] 78(4) Available through: arXiv:0805.4770 [physics.soc-ph]

Lemmouchi, S., Haddad, M., and Kheddouci H., 2011. Study of robustness of community emerged from exchanges in networks communication. In: *MEDES '11 Proceedings of the International Conference on Management of Emergent Digital EcoSystems.*

Lancichinetti, A., Radicchi, F., Ramasco, J. J., Fortunato, S. 2011. Finding Statistically Significant Communities in Networks. *PLoS ONE* 6(4): e18961. Available through: doi:10.1371/journal.pone.0018961

Liu, W. C., 2012. networks [pdf] Taiwan: sinica. Available at: http://newsletter.sinica.edu.tw/file/file/18/1851.pdf

Newman, M. E. J. and Girvan, M., 2004. Finding and evaluating community structure in networks. *Phys. Rev. E.* [e-journal] 69(2) Available through: http://link.aps.org/doi/10.1103/PhysRevE.69.026113

Rosvall, M. and Bergstrom, C. T., 2007. Maps of random walks on complex networks reveal community structure. *PNAS* [e-journal] 105(4), pp.1118-1123 Available through: arXiv:0707.0609 [physics.soc-ph]

Scellato, S. et al., 2011. Understanding robustness of mobile networks through temporal network measures. In: INFOCOM 2011, *30th IEEE International Conference on Computer Communications*. Shanghai, China 10-15 April 2011

Singer, Y., 2006. Dynamic measure of network robustness, In: *Electrical and Electronics Engineers in Israel, 2006 IEEE 24th Convention.*

Steve, G., 2008. A fast algorithm to find overlapping communities in networks. In: PKDD 2008, *the 12th European Conference on Principles and Practice of Knowledge Discovery in Databases - Part I*, pp.408–423, September 2008

Traud, A. L., Kelsic, E. D., Mucha, P. J., and Porter, M. A., 2008. Comparing Community Structure to Characteristics in Online Collegiate Social Networks. *SIAM Review* [e-journal] 53, pp.426-543 Available through: arXiv:0809.0690v3 [physics.soc-ph

Tu, Y. 2000. How robust is the Internet? *Nature* [e-journal] 406, pp.353-354 Available through: http://www.nature.com/nature/journal/v406/n6794/full/406353a0.html

Yang, Q. and Lonardi S., 2005. A Parallel Algorithm for Clustering Protein-Protein Interaction Networks. In: *CSBW '05 Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference – Workshops.*