Abstract

Any real network with infectious viruses must be immunised before it develops an uncontrollable outbreak. The strategy of fragmenting a network into small parts by immunising prioritised nodes has received attention. Although it is known as an NP-hard problem, which means it is impossible to solve exactly, many powerful methods have been proposed. This report aims to propose efficient immunising methods, based on testing and comparing the proposed methods on networks with a community structure.

Contents

1 Introduction

1.1 Motivation

Networks abound in the real world. From a simple example, relationships between people, there are many different kinds not only in Euclidean space, but also in abstract space. Examples are the World Wide Web, metabolic networks, food webs, the Internet, acquaintance networks, and many others. [1]

In 1960s, there was a series of important experiments known as the 'small-world' experiments of Milgram. [2] 296 volunteers in Nebraska and Boston were asked to pass a letter to a target person, 'A', a stockbroker in Boston. There are some rules which one is if they did not know 'A' directly, they were to pass the letter to someone else who might know A, and forward the letter to that person.

About 29 percent of the letters, 64 out of 217, reached A, and the average path length was around five and a half or six intermediate people. It means that any two people can be connected with 5 acquaintances. It is obviously not a property for all networks, but it is worth to know that the world can be connected in short ways. This has become known as the 'six degrees of separation' between any two people, although that phrase was not used in Milgram's paper. This experiment shows that the world is small, so we can easily imagine how diseases can be spread quickly in networks.

When infectious diseases spread across networks, understanding and studies of networks are necessary to prevent enormous epidemics. Examples of serious disease spreading are SARS, MERS, black death and small pox etc. Infectious diseases can exist in various ways across different networks. For instance, we can liken diseases to viruses on the Internet. Internet virus, called Code Red worm virus, was emerged and 359000 computers had infected for 14 hours on July 19th, 2001. [14] A rumour spreading between people also has the dynamics of an epidemic. [1] Developing of Information and Communication Technology accelerated the diffusion of spreading in a global level.

Among such problems, we focused on a way to mitigate the spread of transmissible disease by fast, low-cost methods. Some diseases have limited vaccines, or vaccines that are too expensive to provide to everyone. AIDS, the sexual disease, can be one of example. It is important to have good immunisation strategies. One practical method is to immunise a network before an epidemic outbreak occurs. This project aims to do it with mathematically equivalent to fragmenting a given network into small pieces, with the removal of a minimum number of nodes. In this case, the nodes are individuals in social networks.

The problem is known as an NP-hard problem. [3] It is impossible to solve exactly in polynomial time, so there are various proposed approaches. With many proposed methods available, we need to test and compare them for networks with community structures as they are ubiquitous in empirical network data.

1.2 Project Aims and Objectives

For a long time, a network was regarded as a random graph in which nodes move randomly. However, the discovery of the hub was one development showing that a

network has an order. [9] It led to many dynamic researches of networks, and in recent years, many powerful strategies for immunising networks have been proposed. This project tries to simulate and combine some algorithms to find effective low-cost methods for immunising a network with a community structure. The algorithms aim to prioritise the nodes to be removed, which results in the fragmentation of a network into small pieces. Some community-based algorithms get different results, depending on how nodes are assigned to communities. [8] The size and dynamics of epidemics are depending on the structure of contact networks. [8] Hence, the following section is about community detection methods. The next section is about immunising algorithms that segment a network into many small pieces. Finally, the algorithms and community detection will be tested on various models and collected data in the real world. This paper develops a K-balanced partitioning by using a bipartition algorithm, gbMTP, [5] and combines with a community based immunising algorithm, CbCI. [8] In addition, it is compared with other immunising networks and CbCI with another community detection algorithm, called infomap, in a model and empirical data.

2 Backgrounds

This section will address some backgrounds to understand complex networks. First, we will discuss some terms and properties of networks. As this project focuses on networks with community structures, types and structures are followed by them.

2.1 Definitions and terms

A networks is defined as a set composed by *vertices* or *nodes* and *edges*, connections between nodes. When an edge is directed in one direction, the graph is *directed* graph. *Undirected* graph has an edge directed in both directions between two nodes. The number of edges connected to a node is *degree*. If the edge is weighted, the network is called a *weighted* network, otherwise, is called a *binary* network. In addition, a networks can be classified by the number of types of nodes. A network is composed by one type of nodes, it is called as an *unipartite* network. A bipartite network is composed by two types of nodes, which must be connected with nodes which has another type.

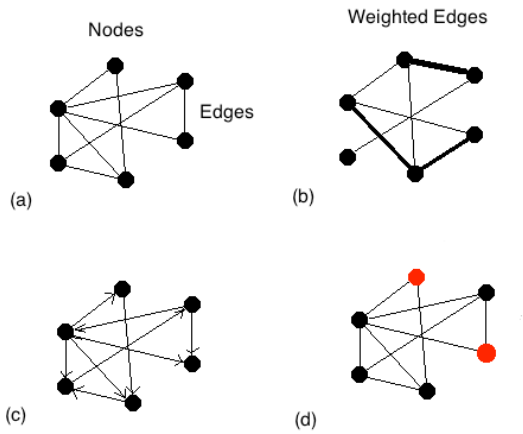

Figure1: Examples of types of networks (a) an undirected and unweighted network (b) a weighted network (c) a directed network (d) a bipartite network, and (a),(b) and (c) are unipartitie networks.

2.2 Properties of networks

2.2.1 Degree

A degree is an important property in networks. The degree of the i^{th} node is denoted with k_i. The number of nodes is denoted with N. Thus, in undirected networks, the number of edges, L, is the half of the sum of the node degrees.

$$L = \frac{1}{2}\sum_{i=1}^{N} k_i \tag{1}$$

In undirected networks, the average degree of nodes, is also important property.

$$\langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2L}{N} \tag{2}$$

We define p_k as the fraction of nodes which have degree k. The degree distribution for the network is expressed by making a histogram of the degrees of vertices. This degree distribution has different shapes depending on models of networks. In random graphs, studied by Erdős and Rényi the degree distribution is binomial or Poisson. However, networks in the real world have highly right-skewed degree distributions. It means that there are a few nodes with high degrees and many nodes with low degrees. Scale-Free Networks are likewise networks with power-law degree distributions.

2.2.2 Community structure

At global level, community structures are defined as structures with many vertexes with high density in a group and low density between groups in networks. The group is equivalent to a community. [1] Most social networks are community structure. For

example, citation networks have communities, classified by interests of each research field. As it is common structure, study of networks with community structure is needed. As the density for one the community is high then others, the dynamic spreading is fast .

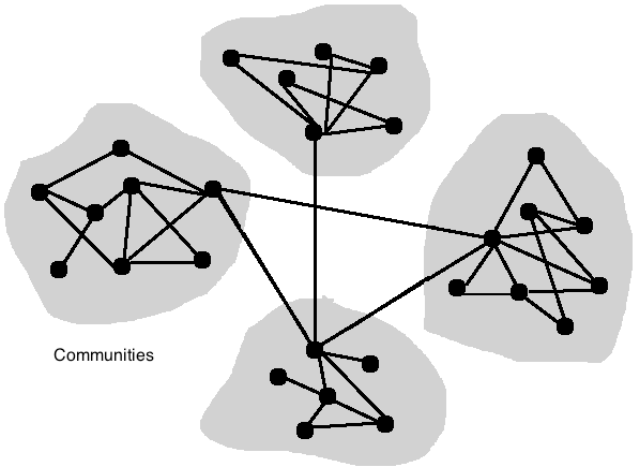

Figure 2: An example of a community structure in networks.

2.2.3 Giant Components

Connected components which are the set of nodes which are connected. It is different with communities or modules. It is completely separated with other components. In this project, it is important property, because the algorithms fragment a network into small pieces. The size of giant components can be a standard to see efficiency of the algorithm. It is a good algorithm which removes the small number of nodes with small size of the largest connected componenet.

3 Community detection algorithms

Some immunising algorithms start with given a network which nodes are assigned into communities. [8] Depending on how nodes are assigned into community, the importance of nodes can be differently measured. The detection aim to divide a network into communities with low density edges between communities.

3.1 K-balanced partitioning (KB)

K-balanced partitioning is to divide a network into equally partitioned K communities. Where the value of k is 2, the problem, called the minimum bisection problem, is already NP-hard. [4] Consequently, it is also NP-hard, so we propose a method that combines some proposed methods. Lim *et al*. proposed a balanced bipartition algorithm, gbMTP, which is based on MTP. [5] MTP is also a proposed method which discovers high quality partitions in graphs. MTP bipartition a network with high quality subgraphs. However, the subgraphs are conducted by removing some nodes from the original graph, and the sizes of subgraphs are not balanced. gbMTP attaches the removed nodes into two subgraphs and aims to balance the two graphs. KB is an algorithm which iterate gbMTP to get K balanced communities. The global balanced partitioning merits to get high conductance. In many real world graphs, hub

nodes are problematic, because when they assigned into one community, interdependency between communities will be high. KB partitioning distributes problematic nodes into different communities to reduce interdependency.

A. MTP (Minus Top-k Partition) [5]

The first step is to find the top-k high degree nodes in a given network. The value of k is set as 0.05N where N is the number of nodes. After removing them from the network, we find the giant connected component form the remaining network. Finally, we divide the GCC into two subgraphs A and B. This algorithm output a set of (A, B) subgraphs. When this algorithm divides the giant connected component in two subgroups, it uses a partitioning algorithm, called METIS. [13]

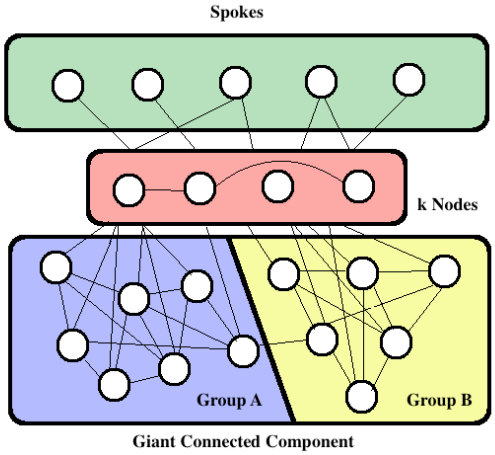

Figure 3: A simple network for explaining MTP algorithm. The red group is composed of top-k nodes with high degree, and the green group is composed of spokes nodes. Spokes are nodes which are only connected with the red group.

B. gbMTP (global balanced Minus Top-k Partition) [5]

gbMTP is an algorithm, extended by MTP. It is a global balanced bipartition algorithm, and has two steps, attaching step and balancing step. Firstly, the output subgraph sets (A, B) does not equal to the original graph, because some nodes are removed.

In attaching step, it starts with ordering the removed nodes to attach into A or B. It attaches every removed node greedily with respect to conductance change. The order of attaching nodes is determined by three rules. The first rule is that The hub nodes are considered before spokes which are nodes only connected with hub nodes. Hub nodes are more significantly affect to the cut edges rather than spokes. The next rule is that high degree nodes are considered before lower degree nodes. It follows decreasing order. Finally, the third rule is that spokes are considered consecutively, when the spokes are connected in the same component. It means that there is no order between the spokes which are connected in the same components. After attaching the removed nodes in the subgraph A or B respect to the conductance change, the output graphs are not guaranteed to be balanced which means that two graphs are not same size graphs.

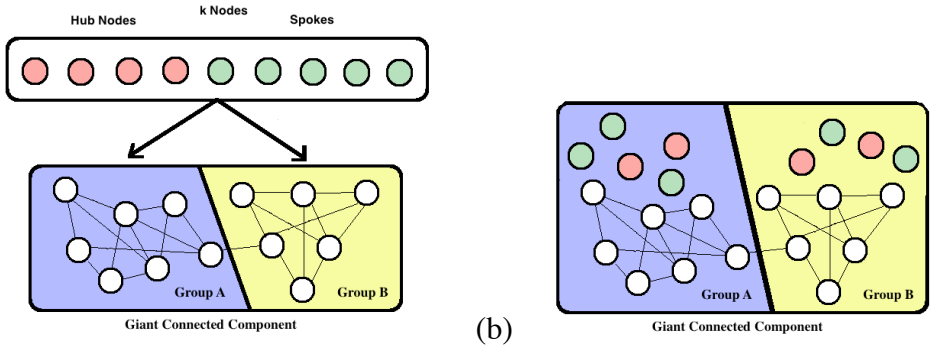

Figure 4: (a) It makes a rank of hub nodes and spokes to reinsert in the giant connected component. (b) The k nodes are reinserted based on changes of conductance when the node is inserted. However, the size is not guaranteed to be blanced.

In balancing step, we assume that B is bigger than A. If so, B has the number of $|B| - |A|$ more nodes rather than A. We select $\frac{[|B|-|A|]}{2}$ number of nodes to move them to A. We greedily find selecting nodes whose result the smallest conductance of movement, and move them from B to A. Finally, this algorithm output two balanced subgraphs A and B.

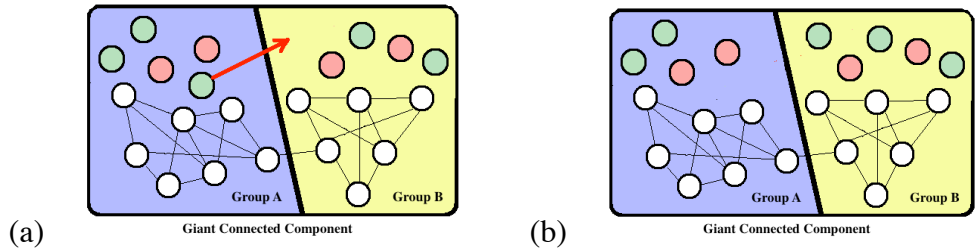

Figure 5: (a) The group A has 12 nodes, but the group B has 10 nodes. One node from the group A should be moved into the group B. The moving node is also choosen based on the conductance change. (b) The network is divided into two groups which have balanced size.

C. KB

From the result of gbMTP, it outputs two balanced graphs. We propose KB algorithm to divide a network into K balanced communities. To avoid confusion of denoting k, in this method, K is the number of communities. The main idea is to iterate gbMTP. If we input the number of iterating, m, the number of output communities are 2^m. For example, we put A and B again into gbMTP, then we will get four balanced partitioning. The final output is a community index list for each node.

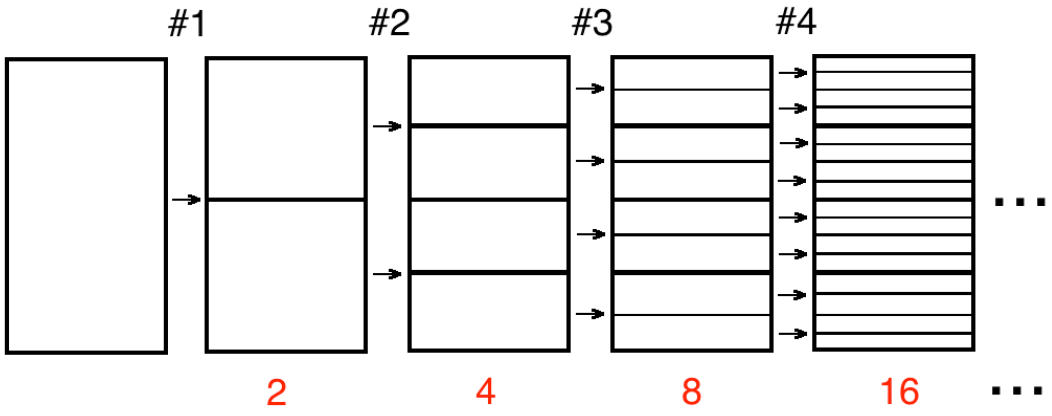

Figure 6: The KB algorithm which each step divides groups into two balanced groups.

3.2 Infomap [6]

It is a community detection method, developed by Rosvall and Bergstrom. It is outperformed with CbCI, one of immunising methods what we used in the next section. [8] The basic idea is to identify the communities by finding a good coarse-grained description of information flows on the network. It is same as finding important structures by describing the trajectory of a random walk on the network.

The simple way is that set a unique name on each node in the whole network. From two-level description of the random walk such that clusters have unique name, but nodes within the network can be reused, leads more short path description. Expression of a random walk can be described by the names, and it gives efficient coarse graining of networks.

4. Immunising algorithms

The basic idea of immunising strategies is to rank nodes in order of great importance then remove the nodes in the order until outbreak of the network. The following strategies have different ways to measure importance of nodes.

4.1 Random

This random method is to remove a node randomly and iterate it. It has no any importance in removed nodes. If we think this strategy in a real disease network, it is same as giving vaccines to a random person.

4.2 Degree

Many proposed strategies are based on the degree of a node. Many real networks have fat-tailed degree distributions. A few hubs play an important key to spread diseases. Degree is a way to find out which nodes are hubs. In this method, we firstly get every degree of every node and attack the node which has the highest degree in the network. It is repeated until the network is fragmented into many small pieces.

4.3 Eigenvector Centrality (EC) [11]

The node rank is the corresponding eigenvector which also corresponding to the largest eigenvalue of the adjacency matrix. For example, the element of the adjacency matrix of an unweighted network is 1 when two nodes are connected, and 0 when they are not connected. Thus, in an unweighted and undirected network with N nodes, the adjacency matrix is a symmetric matrix which elements are only 0 and 1. The first step is to calculate eigenvalues and eigenvectors. Next, the ranks are normalized such that the sum of all centrality scores is 1. The node with the highest value is removed, and we repeat the process with the removed adjacency matrix. If there are disconnected components, the algorithm works individually for disconnected components. The score of the disconnected components is 1/ N. However, this algorithm gives too big score to a few hub nodes, so it does not a good algorithm. [7]

4.4 Closeness Centrality (CC)

This method is based on distances of nodes. Closeness Centrality at node i is the inverse of the sum of the distance from the node to all other nodes in the network.

$$c(i) = \left(\frac{A_i}{N-1}\right)^2 \frac{1}{C_i} \tag{3}$$

A_i is the number of reachable nodes from node i. C_i is the sum of distances from node i to all other nodes.

The high value of the centrality is guaranteed when node i has short average distance between node i and other nodes. The algorithm removes the node with the highest value and repeat the process with the new graph. The value gives high score to nodes which are near the centre of communities. However, it has high computational cost in large networks. [7]

4.5 Betweenness Centrality (BC)

Betweenness centrality of node i is denoted by b_i,

$$b_i = \sum_{\substack{j,k \in N \\ j \neq k}} \frac{n_{jk}(i)}{n_{jk}} \tag{4}$$

n_{jk} is the number of the shortest paths between node j and k. $n_{jk}(i)$ is the number of paths which pass through the node i. The b_i value is a measurement how often each

node appears on a path between node j and k. It is powerful but has high computational cost in large networks. O(N^2M) time are needed for this algorithm. [8]

4.6 PageRank (PK) [12]

This algorithm results from a random walk of the network. It is a algorithms for ranking webpages. It starts with computing probabilities corresponding to the random walk for each pages. The high chance of someone clicks the page in random walks yields the high PK scores. It also performed with removing the node with the PK highest value, and repeat the process.

4.7 Collective Influence (CI) [7]

The value of collective influence is a way to measure influences of nodes on networks.

$$\mathrm{CI}_l(i) = z_i \sum_{j \in \partial Ball(i,l)} z_j \tag{5}$$

where

$$z_i \equiv k_i - 1 \tag{6}$$

It is the CI value of node i, and k_i is the degree of node i. $Ball(i,l)$ is the set of nodes, far from node i within distance l.

Firstly, we calculate this value for all nodes in a network. The descending order of the calculated CI values is the rank to be removed. We remove nodes for the first step of which the number is given as an input of the algorithm. Then, we repeat the calculating and the process. The O($N^2 logN$) time is needed for the CI algorithm.

To reduce costs of the algorithm, we use the reinserting method. It starts with the fragmented network with small pieces such that the fraction of the largest connected component is equal or less than 0.01. Then, we count the number of connected components when the node i is reinserted in the network. The node which has the smallest number of connected components is reinserted into the network for the first time. We repeat the process until the initial network is appeared.

4.8 Community based Collective Influence (CbCI) [8]

It is proposed method by Kobayashi and Masuda. They extended the CI algorithm especially for networks with community structure. It considers a community in the network as a supernode, and the communities performs as one coarse-grained network. To do this, we need to do community detection algorithm. From the algorithm, the coarse-grained network is a weighted network which has N_c nodes as the number of communities.

As same as a CI algorithm, the CI value of the coarse-grained network is defined by

(7)

$$\mathrm{CI'}_l(I) = z'_I \sum_{J \in \partial Ball(I,l)} z''_J$$

where

$$z'_I \equiv \sum_{I'=1}^{N_c} \tilde{A}_{II'} - \min_{I'} \tilde{A}_{II'} \tag{8}$$

$$z''_J \equiv \sum_{J'=1}^{N_c} \tilde{A}_{JJ'} - \tilde{A}_{JJ^-} \ \ (l \geq 1) \tag{9}$$

I and J indicate communities, and $\mathrm{CI'}_l(i)$ value is the collective influence value of community I. J^- denotes the community which is at distance $l-1$ from I.

$$\mathrm{CbCI}(i) = z'_I \sum_{I' \in \tilde{\partial} Ball(I,1)} \frac{\sum_{i' \in I'} A_{ii'}}{\tilde{A}_{II'}} \sum_{\substack{J \in \tilde{\partial} Ball(I,l) \\ I^+ = I'}} z''_J \tag{10}$$

where node *i* belongs to community I.

The algorithm is basically same with CI. We calculate the CbCI and rank them in descending order. The first ranked node is removed, and the algorithm repeats the process until the largest connected component's size is equal or less than 0.01N. Likewise CI algorithm, the removed order is used in the community network. The order of reinserted node is based on the number of communities when the node is reinserted.
The $O(N^2 logN)$ time is needed for the CbCI algorithm without a community detection algorithm.

5 Results

In this section, we compare the algorithms in a network model and empirical data sets. The value q is the faction of removed nodes, and the size of the largest connected component is denoted by G(q).

5.1 Scale-free network model (Barabási-Albert model) [9]

Scale-free network is a model which has a fat- tailed degree distributions. One of property of it is that it has hub nodes. However, it does not have a community structure, so we used this model to see how community based algorithms are outperformed in in a network with community structures, not the scale-free model. It does not show efficiency of the community based algorithms. I used the BA model with N=5000 and $\langle k \rangle \approx 12$, m = 6.

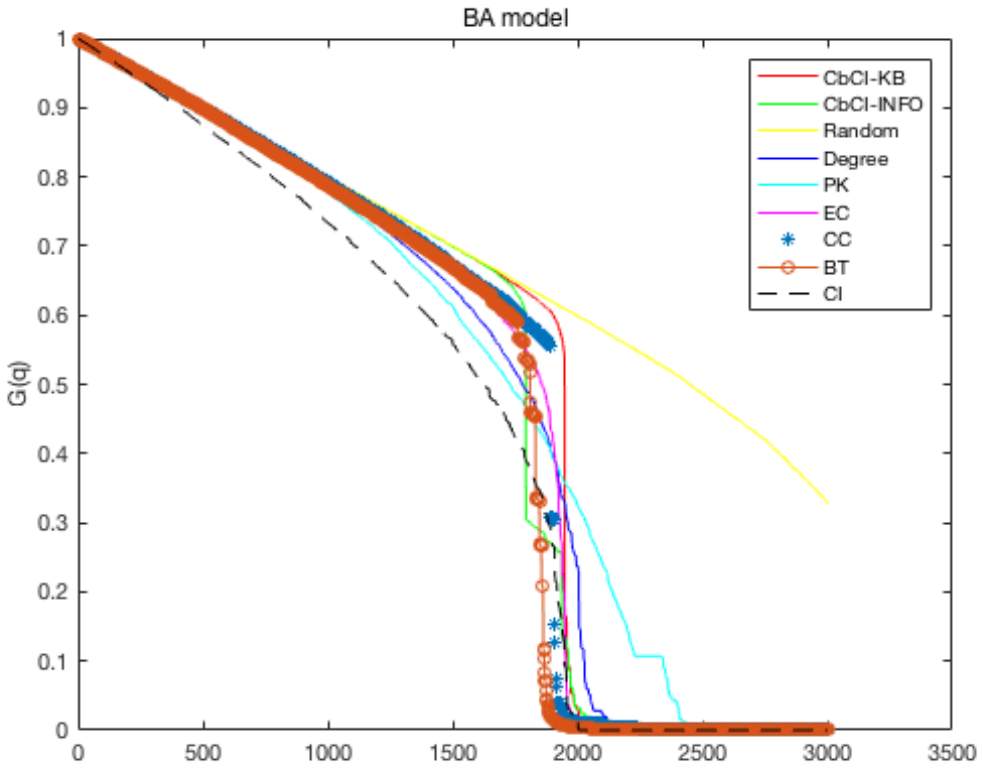

Figure 7: Relationship between removed nodes and G(q). Many algorithms are computed on the BA model.

In Figure 7, CbCI algorithms are not outperformed rather than other algorithms. The scale-free network does not have a community structure, the results show no big difference between other algorithms. 2000 nodes are about 40% of total nodes. It needed to such many nodes to be immunised or fragmented. As expected, the algorithm which removes node randomly show a bad result even it removed 60% of nodes.

5.2 Real world network data

We used three different empirical data sets. [10]

NETWORK	N	M	<K>	N_C	Q[8]	REF
BA	5000	29979	11.99	251	0.174	[9]
EMAIL-ENRON	36692	183831	10.02	2416	0.544	[10]
CA-GRQC	4158	13428	6.45	323	0.785	[10]
FACEBOOK	4039	88234	43.69	77		[10]

Table 1: The number of nodes and edges and an average degree for each network.
All three data sets are undirected and unweighted networks.

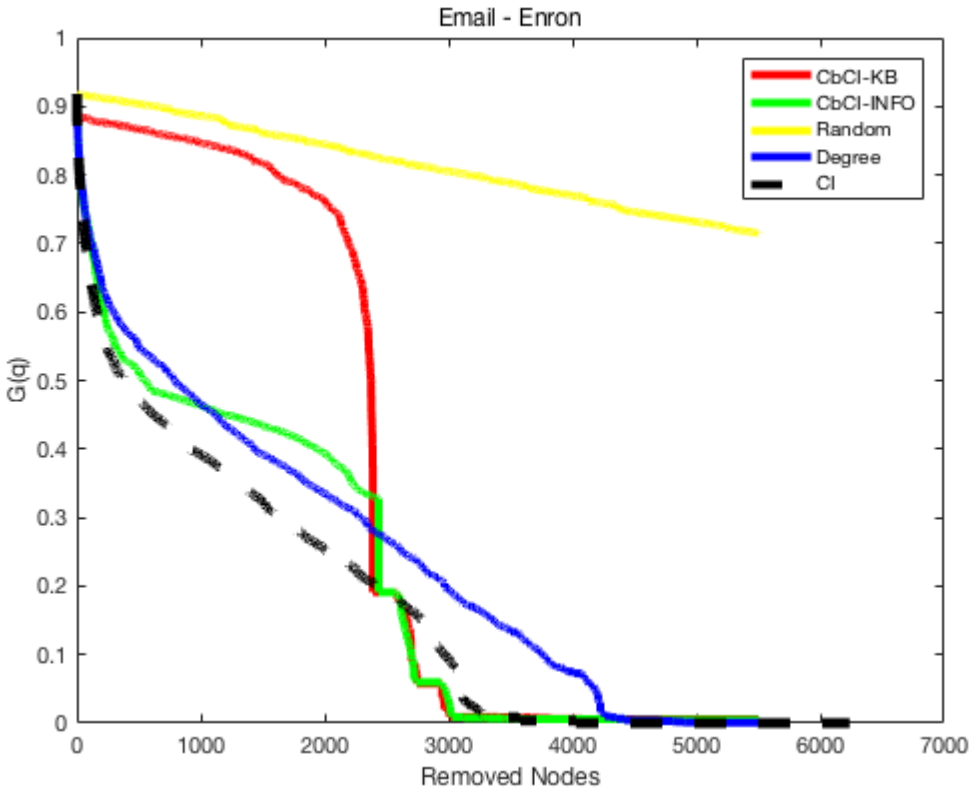

Figure 8: Relationship between removed nodes and G(q). Many algorithms are computed on the email-Enron network.

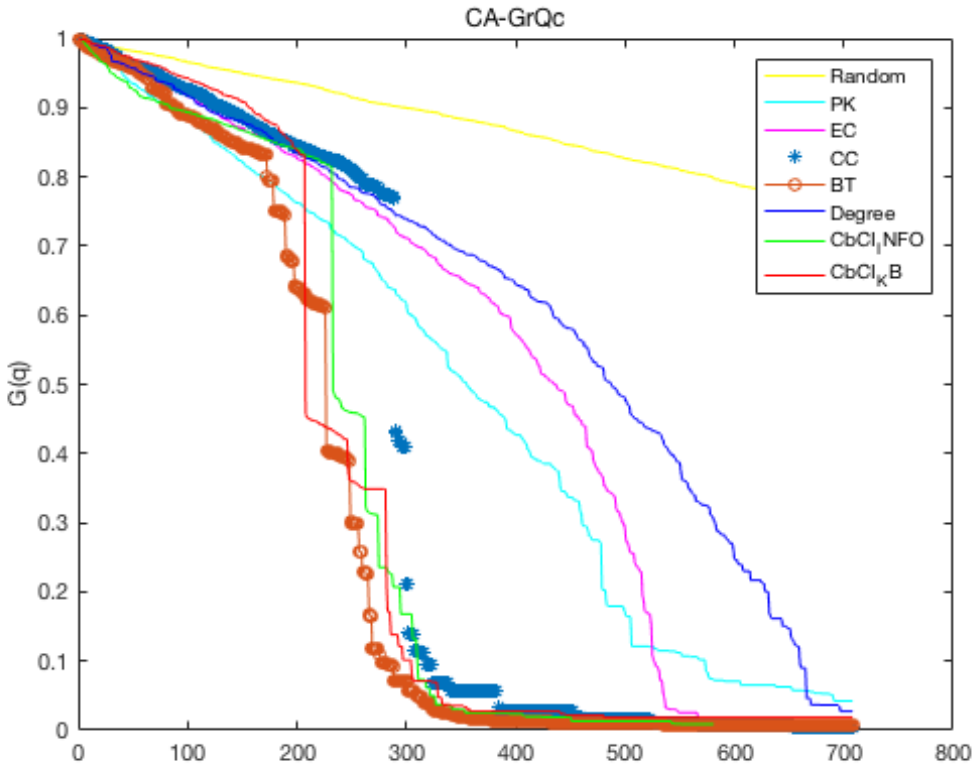

Figure 9: Relationship between removed nodes and G(q). Many algorithms are computed on the CA-GcQc network.

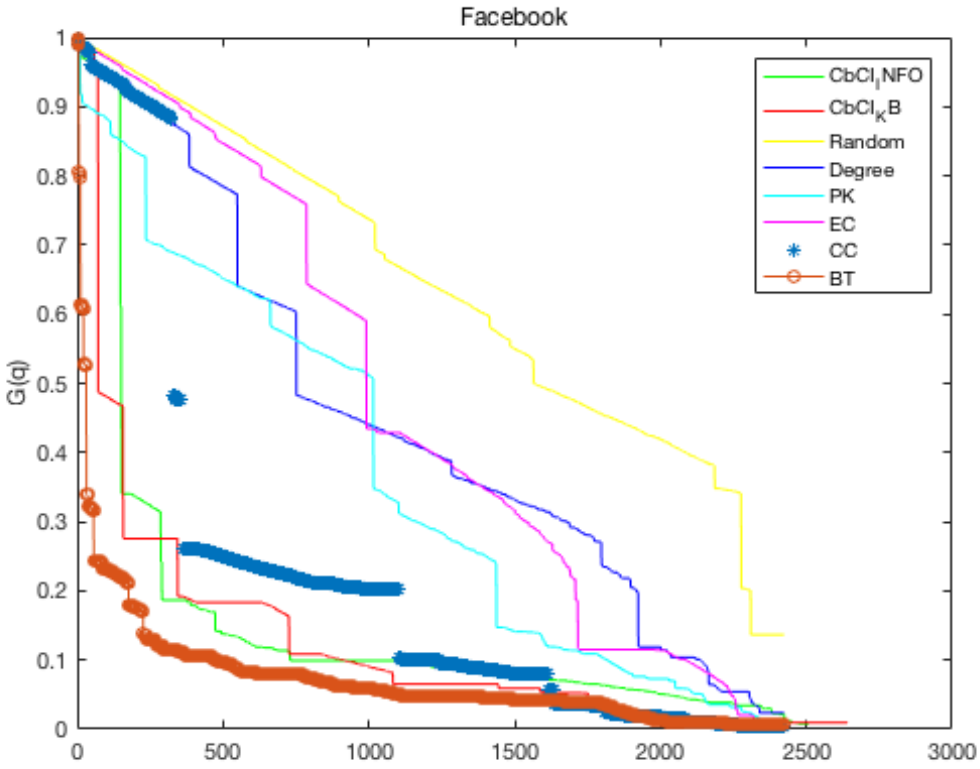

Figure 10: Relationship between removed nodes and G(q). Many algorithms are computed on the Facebook network.

(i) (Figure.8) Email network is the biggest network what we have used. It is a communication network in Enron Corporation. Nodes are connected when one email user send an email to other user such that it is undirected network. Some algorithms are not computed because of the high computational costs. However, community based immunising algorithms are not outperformed in this network. The reason is that Q value, the modularity from the Infomap community detection algorithm, is low in the network. (Table 1) CbCI with KB detection algorithm is not good as the CbCI with Infomap at the front removals. However, the final part is similar.

(ii) (Figure.9) CA-GrQc network is a collaboration network in General Relativity and Quantum Cosmology. [10] If two nodes collaborated at least once, two nodes are connected. It shows the outperforming of community based algorithms. The BT algorithm is also good, but it takes too many times, so the CbCI can be one of substitutions. Comparing between community detection algorithms, they perform similar, but KB is a little better.

(iii) (Figure.10) Facebook network has the small number of modules over the whole size of network. All algorithms are simulated with removing 60% of nodes, so the fraction is quite high rather than other network. However, it has a high modularity, so the community based algorithms are outperformed. Likewise, CA-GrQc network, there are not big differences between KB and INFOMAP, but KB works a little better. BT yields the best results despite of high computing costs.

6 Discussion

We compared many immunising algorithms and made a KB community detection algorithm with CbCI for networks with community structures. Many proposed algorithms are worked well, and combining between algorithms are also a good way to make other algorithms. If we know the network's property such as a modularity or a structure of it, it is good that applying an algorithm which can outperform in the network depending on the properties. Some other immunising algorithms with KB community detection, there are some better strategies can be made.

References

[1] M. E. J. Newman, The Structure and Function of Complex Networks, *SIAM Review* 45 (2003), 167-256.

[2] T. Jeffrey, and S. Milgram. An Experimental Study of the Small World Problem, *Sociometry*, 32 (4) (1969), pp. 425-443.

[3] S. Fortunato, Community detection in graphs, *Phys. Rep*. 486 (2010), 75-174.

[4] R. Krauthgamer, J. S. Naor, and R. Schwartz, Partitioning graphs into balanced components, *In 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 09 (2009), pp. 942-949.

[5] Y. Lim, W. Lee, H. Choi, and U. Kang, MTP: Discovering High Quality Partitions in Real World Graphs, *WWW,* 20 (3) (2017), pp. 491-514.

[6] M. Rosvall and C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, *Proc. Natl. Acad. Sci. USA*, 105 1118 (2008).

[7] F. Morone, H. A. Makse: Influence maximization in complex networks through optimal percolation, *Nature* (2015).

[8] T. Kobayashi and N. Masuda, Immunizing networks by targeting collective influencers at a mesoscopic level, arXiv:1605:03694, (2016).

[9] A.-L. Barabási and R. Albert, Emergence of scaling in random networks, *Science,* 286 (1999):509-12.

[10] http://snap.stanford.edu/.

[11] P. D. Straffin, Linear algebra in geography: eigenvectors of networks, *Mathematics Magazaine,* **53**(1980), 269-276.

[12] C. Moler, Experiments with MATLAB. Chapter 7: Google PageRank, *MathWorks, Inc.(*2011)

[13] G. Karypis and V. Kumar, Multilevel k-way partitioning scheme for irregular graphs. J. Parallel Distrib. Comput., **48** (1998), pp.96-129.

[14] D. Moore and C. Shannon, The Spread of the Code-Red Worm(CRv2), *CAIDA Analysis*, (2006).