Comparison of network complexity meausres

Yipei Zhao
Student number: 170145594

A thesis presented for the degree of
Master of Science Data Analytics
Supervised by Jens Christian Cluassen
October 2021

Contents

1 Introduction

In modern world, there are many fields where a network can be investigated and studied. For example, electric cables in a region would form a large network.[1] Also, analysing social networks and use their unique properties to drop promotion precisely would help firms to increase profits.[2] In addition, food webs can be considered as networks and they have been studied for many years.[3] There are many aspects of a network, but it is vital to ask: what is the complexity of a network? How a network to be designed as complex/simple? The goal of this project is to summarise and compare different complexity measures, including implementations and applications on different real networks.

In this project, we want to be specific on transport networks. A transport network transfer either passengers or goods with the flow, from initial location to their destination. To apply complextiy of transport networks, the designers of the network have to think about this: how do I modify the complexity of the network and what change will it bring to the network? Transport networks is different to common networks in real world, but before investigate the uniqueness of transport network, it is vital to study network science as a whole.

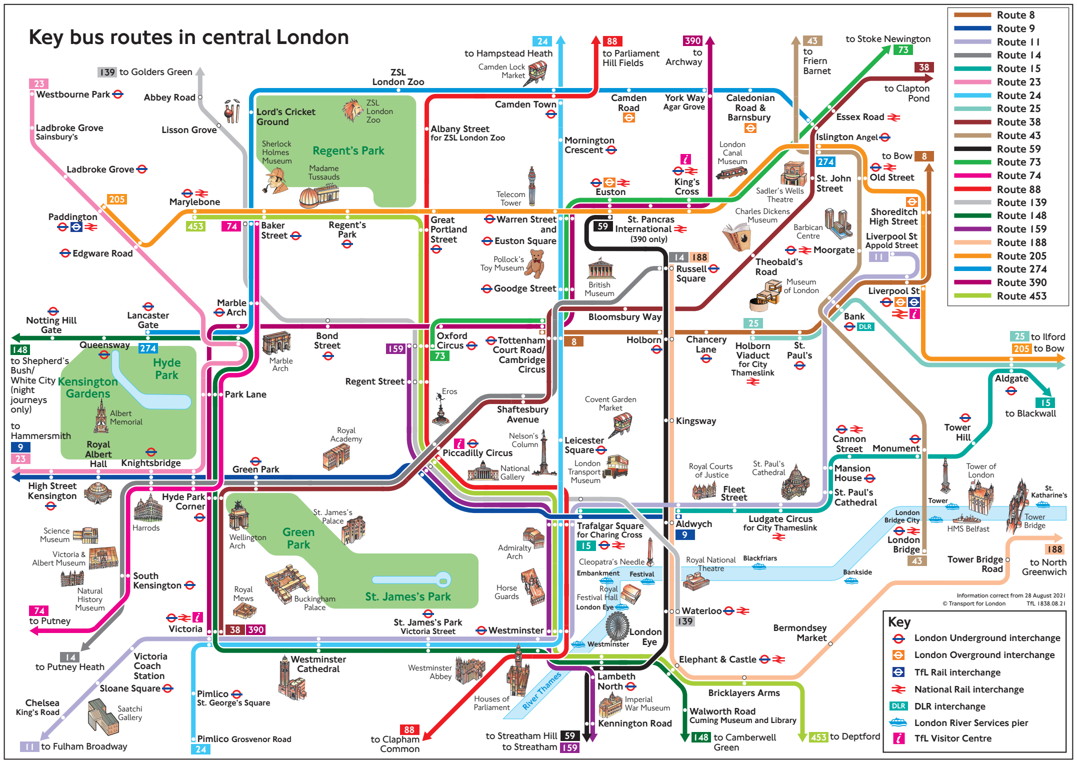

Figure 1: Key routes in central London.[4]

1.1 Graph theory

Firstly, what does network scientists study? A network contains n nodes/vertices and m edges/links, nodes can be connected using edges. Allowing construcion of assets. In a transport network, a node can be consdiered as a stop on the road and edges are the roads/path connecting stops.

> ”we view network science as the study of the collection, management, analysis, interpretation, and presentation of relational data” [5]

1.1.1 Graph

To stay simple, all networks in this project have the following properties:

- Undirected. Edges are not allowed to have direction. If the graph is originally directed, it will be turned into an undirected graph.
- Unweighted. All the edges are fairly recognized and no weights are assgined.
- No multi-edges. There will be at most one edge between two nodes.
- No self-links. Nodes are not allowed to connect to themselves.

Theorem 1.1 *(Graph/network) A graph or network G can be represented by three sets. A node set $V : \{V_1, V_2, V_3...V_n\}$ and an edge set $E : \{E_1, E_2, E_3...E_m\}$ and a function set ι such that $E \to V_\alpha \times V_\beta$ implies there is a connection between node V_α and node V_β.*

In non-mathematicial words, a network or a graph is a collection of nodes and edges, each node can connect to other nodes via edges.The border between the term network and graph is very ambiguous, and network scientists use them interchangeably. There are many different type of graphs. A path graph contains n nodes and all nodes are connected in a consequence, or a clique which all nodes are connected to others. Since a clique connects all its nodes, leads to a number of $n(n-1)/2$ edges in total.

Theorem 1.2 *(Subgraph) A subgraph G' is a slice of a graph G, such that $V' \epsilon V$, $E' \epsilon E$ and ι' matches two nodes belongs to V', i.e. $E' \to V'_\alpha \times V'_\beta$. All the edges must have their starting and ending points within the subgraph.*

Theorem 1.3 *(Neighbours) If node i is connected to node j, node i is a neighbour of node j, vice versa. If node i is connected to node j, k, l, the node set $\{j, k, l\}$ is said to be the neighbourhood of node i.*

Theorem 1.4 *(Distance) Distance $d_{i,j}$ in a undirected, unweighted graph is the minimum number of edges required to connect node i and j. Average distance $\langle d \rangle$ is defined as $\langle d \rangle = \frac{2}{n(n-1)} \sum_{i,j} d_{i,j}$.*

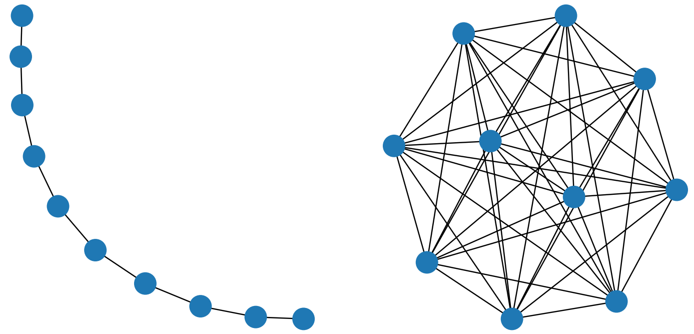

Figure 2: From left to right: A path with 10 nodes($n = 10$) and 9 edges($m = 9$); a clique with 10 nodes and 45 edges($m = 45$).

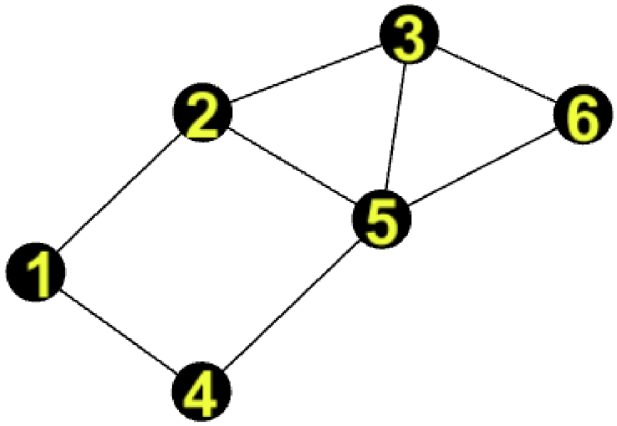

Figure 3: A simple network with 6 nodes and 8 edges.

Refereing to figure 3, distance between node 1 and 6 is 3, denoted by $d_{1,6} = d_{6,1} = 3$. Distance between node i and node j is equal to node j to node i: $d_{i,j} = d_{j,i}$. The average distance of the network is 1.53.

A node i has degree k_i if it has k edges connected to other nodes. For figure 3, we have $k_1 = 2, k_2 = 3, k_3 = 3, k_4 = 2, k_5 = 4, k_6 = 2$. Thus, the total number of edges L in a network can be calculated:

$$L = \frac{1}{2}\sum_{i=0}^{i} k_i \tag{1}$$

Using equation 1, we can calculate the total number of links for figure 3 is equal to:

$$L = \frac{1}{2}(2+3+3+2+4+2) = 8 \tag{2}$$

An important parameter of a network is generated using degrees, which is average degree of a network:

$$\bar{k} = \frac{1}{N}\sum_{i=0}^{i} k_i \tag{3}$$

or, simply:

$$\bar{k} = \frac{2L}{N} \tag{4}$$

Applying formula 3 to figure 3, the average degree is:

$$\bar{k} = \frac{1}{6}(2+3+3+2+4+2) = \frac{8}{3} \tag{5}$$

or equivalently, using formula 4:

$$\bar{k} = \frac{1}{6}(8 * 2) = \frac{8}{3} \tag{6}$$

Theorem 1.5 *(Clustering coefficient) Clustering coefficient C_i illustrates the proportion of edges exists between a node i's neighbours. For a given node i with degree k, the clustering coefficient is defined as:*
$C_i = \frac{2L_i}{k_i(k_i-1)}$
where L_i represents the number of links exists between node i's neighbours.

In figure 3, node 2 and node 4 are node 1's neighbours. There are no edges between node 2 and node 4, so $C_1 = 0$. On the otherhand, node 3 has neighbours 2,5 and 6. There are 2 edges between node 2,5 and 6, $C_i = \frac{2*2}{3*2} = \frac{2}{3}$. Average clustering tends to suggest the average connectivity of the network.

Theorem 1.6 *(Laplacian Matrix) A Laplacian matrix is defined as:*

$$L_{i,j} = \begin{cases} k_i & \textit{if i=j} \\ -1 & \textit{if an edge exists between node i and j, and i is not equal to j} \\ 0 & \textit{otherwise} \end{cases} \tag{7}$$

Diagonal elements $L_{i,i}$ suggests the degree of node i.A non-zero non-diagonal element $L_{i,j}$ implies a connected edge between node i and j. Laplacian matrix allows quick calculation of the graph properties and a visulisation of the network.

In order to evaluate different complexity measures, we need to introduce some graph models that are commonly used to model real world problems.

1.2 Random graphs

There are infinite number of real networks in real world, but defining or observing all of them is not feasible. For simulations and comparisons, network scientists introduced the idea of random networks. They are also known as Erdős-Rényi network in honour of two mathematicians: Paul Erdős and Alfréd Rényi. They have important contributions to understand the properties of a random network[6].

There are two definitions of a random network:

- $G(n, p)$ network. A network with n nodes will be initialised, there will be at most $(n)(n-1)/2$ edges. Each edge will be instantiated with probability p. This approach brings a randomness property to the graph; number of edges m. The expectation of m is equal to $p(n)(n-1)/2$.
- $G(n, m)$ or $G(n, L)$ network. A network with n nodes will be initialised, m or L edges will be connected from a random node to another random node. Due to the non-randomness of $G(n, m)$ networks, they are used to simulate the behaviour of a random network in this thesis. They will be also referred to random network/random graph in this thesis unless stated otherwise.

Previously, definition of clustering coefficient and average distance were introduced. For a given random graph, the clustering coefficient and average distance can be calculated using formulas.[7] The average clustering coefficient of a random graph is p, or $2m/((n)(n-1))$(number of instantiated edges divided by total number of possible edges). Clustering coefficient is used to illustrate the ratio between connected links and possible links between a node's neighbours. If there are k neighbours of a node, there can be at most $k(k-1)/2$ edges between the neighbours. In these $k(k-1)/2$ edges, only p of them will be instantiated. Thus, the ratio of connected edges and possible edges becomes $\frac{pk(k-1)/2}{k(k-1)/2} = p$. Additionally, average distance of a random graph $L_r \approx \frac{ln(n)}{ln(\bar{k})} \approx \frac{ln(n)}{ln(2m/n)}$. To be noticed, both parameters are expectations.

1.3 Rewiring

Except random graphs, network scientists desire more techniques to allow them to modifiy the properties and variables of a network. Network scientists would use a technique called rewiring to change the properties and parameters of a network, and monitor the change of parameters respect to the rate of rewiring.[8] In this project, two simple rewiring techniques are used: single link rewiring and pairwise rewiring.

- Starting with an edge (u, v); with starting node u and ending node v. Single link rewiring will look for a node w that hasn't yet been connected to node u. Once w is found, edge (u, v) will be removed and a new edge (u, w) will be added to the network.

- Starting with two edges (u, v) and (x, y). Pairwise rewiring will remove both edges, and two new edges will be added: (u, y) and (x, v).

Single link rewiring tends to give higher randomness to the network, and pairwise rewiring preserves the degree distribution. Both rewiring techniques require a parameter p, which is the probability of rewiring for each edge. If $p = 1$, using single link rewiring will cause the network to become a random network. However, since pairwise rewiring preserves the degree detribution, $p = 1$ will not cause the network to become completely random.

1.4 Small-world

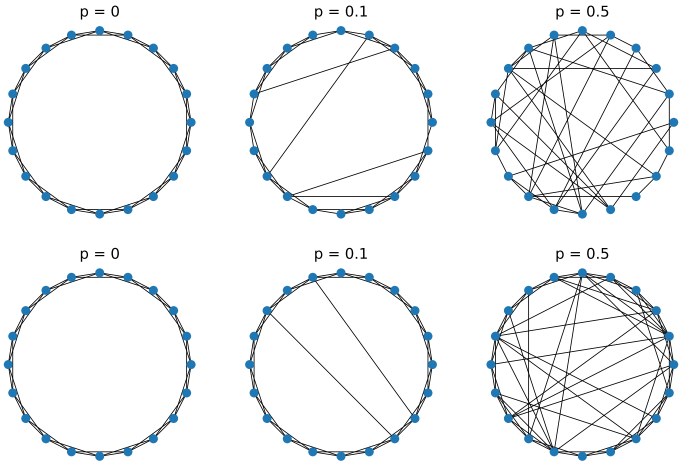

Figure 4: A demonstration of WS model(top) and NW model(bottom). The paramters are: $n = 20$, $k = 4$, $p = 0, 0.1, 0.5$.

About 50 years ago, a famous study was carried out by Standley Milgram[9] in the interest of this question: how many intermediates are needed to pass a message between two irrelevant or distanced person? This is known as the small-world problem. As counterintuitive as it may seem, the medium number of intermediates needed is only 5(an average of 6)[9]. This is not a fair and undoubtable experiment but it is almost impossible to determine the actual number of messengers needed in real world. Nevertheless, this number would be smaller than most peoples' expectation. Mathematically, the small world problem is the study of graphs with small average distance, since network scientists believe that in real world networks, the average distance is small with high average clustering coefficient. Previously, we introduced the formula to calculate the average distance L_r of a random graph. Thus, if a graph has $L/L_r < 1$

and $C/C_r > 1$, this graph can be considered to have the small-world property.

A small-world network can be generated using a Watts-Strogatz(WS) model[10] or a Newman-Watts(NW) model(a variant of the WS model)[11]. Both models require three parameters: number of nodes n, number of connected closest neighbours k and rewiring probability p. The key of both model is rewiring. The graph starts with n nodes, each node is connected to $k(k-1$ if k is odd) nearest neighbours; $nk/2$ edges will be created. For each edge, there is a proabbility p that this edge will be rewired to another ending node(single link rewiring). While rewiring, the WS model removes the edge (u, v) and add a new edge (u, w). Thus, the number of edges stays the same. However, the NW model maintains the edge (u, v) and adding the new edge (u, w), causing the expectation of number of edges after rewiring to be $nk/2 + pnk/2$. Rewiring will add short paths to the networks, and cause the average distance to be exceptionally smaller. Suggested by Barabási[7], to obtain both high clustering and low average distance(properties of a small-world network), p should be between 0.001 and 0.1.

1.5 Scale-free network

A controversial topic of network science is whether real networks are usually scale-free[12]. To understand what is a scale-free network, we need to scope into the degree distribution of graphs.

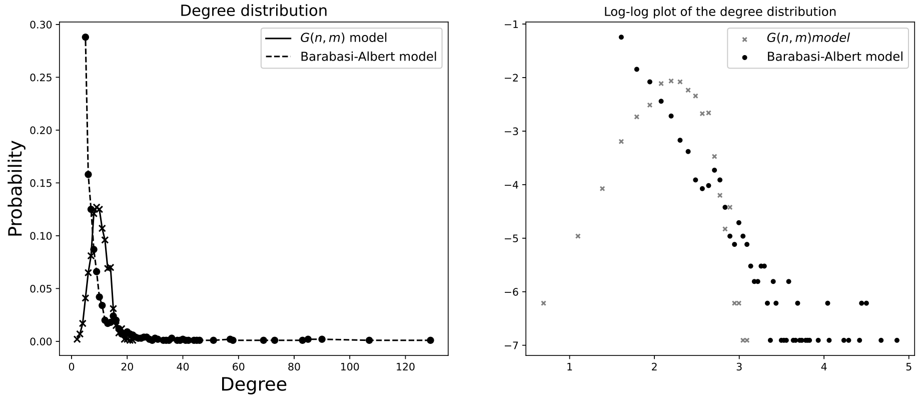

Figure 5: Degree distribution of a $G(n, m)$ graph with $n = 1000, m = 5000$ and a graph generated using Barabási-Albert model with $n = 1000, m = 5$.

Suggested by Barabási[7], the degree distribution of a random network is expected to follow a binomial distribution. However,it might not be the actual distribution of real networks. A controversial idea that hasn't yet been proven in network science community is: are real networks' degree distribution follows a power-law distribution?

A power-law distribution follows: $P(k) \sim k^{-\gamma}$, the parameter γ is typically in the range $2 < \gamma < 3$. If the degree distrution of a network follows a power-law distribution, the network is said to be a scale-free network. Even though there are counter-examples[13], many network scientists still believe that real networks are scale-free[14]. In order to further simulate the behaviour of real networks, Barabási introduced the Barabási-Albert(BA) model to create scale-free networks.[7].

The BA model requires two parameters: n and m. Initally, only one node is created and $n-1$ nodes will be added to the network consequently. Whenever a node is added into the network, it will connect to m nodes. The logic of connection is the key of this model. Nodes are more likely to connect to nodes with more edges than nodes with less edges. For instance, a node has been added to the network, it is more likely to connect to a node with degree $k = 7$ than a node with degree $k = 3$. This logic of connection is called preferential attachment. Essentially, like in real world, nodes are more likely to connect to nodes that have more impact on the network.[15] The BA model ensures most of the nodes have low degree, whereas only a few nodes have exceptionally high degree, as shown in figure 5. An ideal way to fit the power-law distribution is using a linear regression to fit the data in log-log scale.

1.6 Generating graphs

All graph models are utilized as below(unless specified):

- $G(n, m)$ random graphs/networks or random graphs/networks. With given n, m will be randomly selected between $n-1$ to $n(n-1)/2$ to create various networks with different number of m.

- For WS and NW graphs, three parameters are required. Given parameter n, k will be randomly selected between 1 and $(n-1)$ to give a larger range of m. p is also randomize, within the range 0.01 and 0.1 to simulate small-world network as mentioned in section 1.4. By randomize two parameters, more diversity samples will be generated.

- Two parameters are needed for BA graphs, given n, m will be randomize between 1 and $(n-1)$, so we can generate samples with different number of edges.

2 Methods

2.1 Implemeted methods

In this project, 8 methods will be introduced, 7 of them are based on paper pubilished[16][17], and a new measure MA_{RI} will be introduced. MA_{RI} is based on the idea of MA_g, and it also performs similarly.

To compare complexities, all complexities should be normalised within range $0 < complexity < 1$. Hence, finding the upper bound and lower bound of each complexity is essential, even though the lower bound for most measures is 0.

- Subgraph measures:
  - $C_{1e,st}$
  - $C_{1e,spec}$
  - $C_{2e,spec}$
- Product measures:
  - MA_g
  - MA_{RI}
  - Cr
  - Ce
- OdC (Entropy measure)

2.2 Different subgraph measures

Definition of subgraph was introduced in section 1.1.1, so a intuitive idea of measure is to count the number of different subgraphs in a given graph. The question is: how to count subgraphs and how to determine whether they are truly identical or not? Different subgraphs can be created using edge deletion. There are two approaches in this project: removing one edge or remove two edges. Removing more edges is not optimal considering the complexity of calculation. Deleting one edge from a graph with m edges will result in m subgraphs, or $\binom{m}{2}$ using two edges deletion. After obtaining m or $\binom{m}{2}$ subgraphs, the next step is to find number of unique subgraphs. Two identical subgraphs are said to be isomorphic. To determine isomophicness, there are two possible methods.

The first approach is to find out the number of spanning tree of a subgraph. A spanning tree contains all the nodes in a subgraph with minimum number of edges. This can be calculated using the Kirchkoff's theorem[18]: number of spanning tree = cofactor of Laplacian matrix L. By counting different number of spanning trees, isomorphicness can be determined: two subgraphs wtih same m ,n and same number of spanning trees are isomorphic. This is a computationally cheaper approach with possible errors. A more complex but precise way to determine isomorphicness is to compare the spectrum of Laplacian matrix of subgraphs. Calculating and comparing both Laplacian matrix and signless Laplacian matrix(all entries are positive or 0) will offer a better result. If two subgraphs with same spectrum on both Laplacian matrix and signless Laplacian matrix, they are isomorphic. As mentioned, the complexity need to be normalised to allow comparison. A good upper bound for both edge deletion method is $m_{cu} = n^{1.68} - 10$.[16].Three measures can be defined:

- $N_{1e,st}$, the number of different subgraphs with different number of spanning trees, after cutting one edge. The normalised complexity $C_{1e,st} = (N_{1e,st} - 1)/(m_{cu} - 1)$.
- $N_{1e,spec}$, the number of different subgraphs with different spectrums, after cutting one edge. The normalised complexity $C_{1e,spec} = (N_{1e,spec} - 1)/(m_{cu} - 1)$.
- $N_{2e,spec}$, the number of different subgraphs with different spectrums, after cutting two edges. The normailised complexity $C_{2e,spec} = (N_{2e,spec} - 1)/(\binom{m_{cu}}{2} - 1)$.

2.3 Potential problems and solutions of different subgraph measures

During the implementation of different subgraph measures, problems were found, possible solutions are also given for future implementation.
Different subgraph measures are principally simple, but they are complex to compute, within at least $O(n^2)$ time[16]. This is not the only problem. An upper bound of the complexity $m_{cu} = n^{1.68} - 10$ was assumed by Kim and Wilhelm[16] to normalise the complexity. However, from the simulation, we found that this may not be the actual upper-bound of the different subgraph measures.

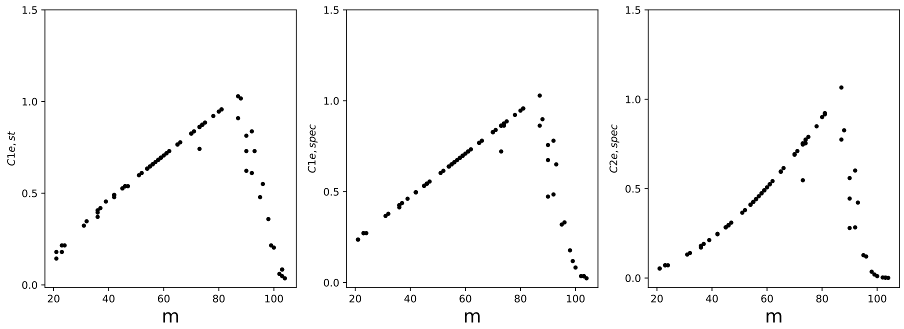

Figure 6: Different subgraph measure of 1000 $G(n, m)$ random graphs, with $n = 15$.

The complexity is abnormal for graphs with around 90 edges and 15 nodes as shown in figure 6. This could imply that the upper bound assumption m_{cu} is not correct, but there is another possible reason, which is the problem of floating point arithmetic.

On most machines today, numbers are represented in binary system[19]. For example, 0.2 is recorded as 0.00110011001100110011... in a binary system. This series is infinite, represented by $1 * 2^{-3} + 1 * 2^{-4} + 1 * 2^{-7} + 1 * 2{-}8....$ For obvious reasons, computer scientists don't want to work with infinite series, therefore, the series is approximated. On a modern compueter, the series is usually approximated to 63 digits with 1 digit represents the sign of the number. After approximation, the error could cause the equal operation to fail. A well known example is that for modern programming language or machine that operates this numbering system, 0.2+0.1 does not equal to 0.15+0.15. As a result, the inaccurate comparison may cause more number of different subgraphs than actual.

The core of different subgraph measure is to compare the cofactor($C_{1e,st}$) or spectrum($C_{1e,spec}$ and $C_{2e,spec}$) of a subgraph. Given the fact that the proabbility of a decimal number to appear in the sprectrums is high and the cofactor might exceed the upper bound of numbering system[20]. The comparisons will be inaccurate with a high probability. There are three possible solutions:

- As suggested, errors will be made when approxiamted by the machine. An error threshold can be used when comparing spectrums and cofactors. For example, two numbers with relative error less than 1% can also be considered as equal numbers. One disadvantage is the increase of complexity, taking more time and effort to compare the spectrums/number of spanning trees.

- Similarly, numbers can be rounded before comparison to avoid error. This is used in the implementation of different subgraph measures, all cofactors and spectrums are rounded to first 10 significant figures. This solution requires less computation time than first solution and reduce the number of misclassification. The drawback is that similar graphs can be considered as isomorphic graphs, this also applies to the first provided solution, but with higher accuracy for large graphs. This may still gives complexities larger than 1, but it is the best solution considering the effort spent. Thus, this solution iwll be used in this project.

- Instead of using m_{cu} as the upper bound, m or $n(n-1)/2$ can be used for one-edge-deleted subgraph complexity and $\binom{m}{2}$ for two-edges-deleted subgraph complexity. This gurantees the normalisation and avoid the mistake that caused by the first two solutions, but on the other hand, causing the complexity to be different.

A unique problem with $C_{2e,spec}$ is the value is not properly scaled for small graphs. The upper-bound of $C_{2e,spec}$ is 0.5 while $n \leq 7$. To have an upper-bound at 1, the complexity values have to be scaled by 2. However, scale by 2 will cause the complexity to exceed 1 for larger graphs. Thus, we sticked to the original normalisation and scaling, $C_{2e,spec}$ will have an upperbound at 0.5 for $n = 7$, and less for smaller ns.

2.4 Product measure

The core idea of product measure is to assign low complexity at both extremes(path and cliques) and higher complexity to graphs with medium number of edges.

2.4.1 Medium articulation for graphs(MA_g)

This measure is based on redundancy and mutual information of a graph.[21] The unnormalized complexity can be defined as the product of redundancy and mutual in-

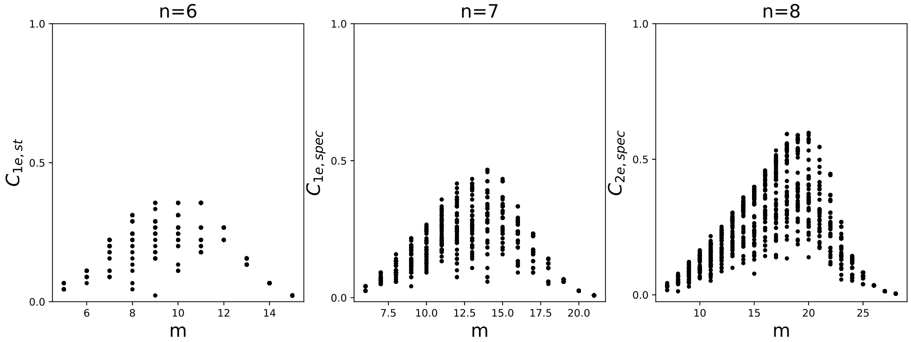

Figure 7: $C_{2e,spec}$ complexities of $G(n,m)$ random graphs for n = 6,7,8 respectively. For each m, 50 samples are generated

formation $C = R * I$, with redundacy $R = \frac{1}{m}\sum_{i,j>i} ln(d_i d_j)$ and mutual information $I = \frac{1}{m}\sum_{i,j>i} ln(\frac{2m}{d_i d_j})$. The core of this measure is the variable $d_i d_j$. It is product of degree of two nodes. This variable will be maximised when the graph is fully connected(clique) and minimized when the graph is least sparse and highly ordered(path).

- Highest redundancy: $R_{clique} = 2ln(n-1)$
- Lowest redundancy: $R_{path} = 2(\frac{n-2}{n-1})ln(2)$
- Highest mutual information: $I_{path} = ln(n-1) - (\frac{n-3}{n-1})ln2$
- Lowest mutual information: $I_{clique} = ln(\frac{n}{n-1})$

Since the maximum and the minimum of redundancy and mutual information can be found, normalization technique can be applied to ensure the complexity MA_g is bounded by 0 and 1.

$$MA_g = MA_R * MA_I, \tag{8}$$

with:

$$MA_R = 4(\frac{R - R_{path}}{R_{clique} - R_{path}})(1 - \frac{R - R_{path}}{R_{clique} - R_{path}}) \tag{9}$$

$$MA_I = 4(\frac{I - I_{clique}}{I_{path} - I_{clique}})(1 - \frac{I - I_{clique}}{I_{path} - I_{clique}}) \tag{10}$$

2.4.2 Efficentcy complexity(Ce)

Efficiency complexity is based on the efficiency of adding shorter path in a network.[22] As suggested in section 1.4, adding short paths will significantly shorter the average shortest distance in a network. However, adding more paths will cost energy. The ratio between shorten distances and cost of energy is defined as the efficiency. A complex graph should maintain short average distance and low energy used as possible. The efficiency is calculated as $E = \frac{1}{n(n-1)/2}\sum_{i,j}\frac{1}{d_i,j}$. To normalise the complexity, identify the lowest graph with lowest efficiency, a path, is: $E_{path} = 2/(n(n-1))\sum_{i=1}^{n-1}(n-i)/i$. Consequently, the normalised complexity is $C_e = 4(\frac{E-E_{path}}{1-E_{path}})(1-\frac{E-E_{path}}{1-E_{path}})$

2.4.3 Graph index complexity(Cr)

In section 1.1.1, Laplacian matrix was introduced. Cr is depending on another graph matrix which is called adjacency matrix. The adjacency matrix is essentailly Laplacian matrix with all diagonal elements being 0. This records all the edges within the network. Using previous studies, all eigenvalues of the adjacency matrix are real.[23] The largest eigenvalue is also knows as the index r. To normalise the complexity, r is found out that $2cos(\pi/(n+1) \leq r \leq n-1)$.[23] Therfore, a normalised complexity can be constructed, which is given by: $Cr = 4c_r(1-c_r)$ with $c_r = \frac{r-2cos(\pi/(n+1))}{n-1-2cos(\pi/(n+1))}$.

2.5 Offdiagonal complexity(OdC)

A node-node link correlation matrix need to be constructed to calculate OdC.[17] The matrix element $c_{i,j}$ records the number of neighbours with degree $j \geq i$ of all nodes with degree i. OdC assigns higher complexity to graphs that their nodes have no preferences on their neighbours degree. A vector a_n is calcualted by summing up all the rows in the matrix. The normalised OdC is calculated as:
$OdC = -(\sum_{n=0}^{k_{max}-1} A_n ln(A_n))/ln(n-1)$ with $A = a_n/\sum_{n=0}^{k_{max}-1} a_n$(proability of each a_n).

2.6 MA_{RI}

MAg measure is a product measure, which assigns higher complexity to graphs with medium number of edges and lower complexity at both extremes(path and clique). Using the product of redundancy R and mutual information I, with normalisation, MAg is defined as[16]:

$$R = \frac{1}{m}\sum_{i,j>i} ln(d_i d_j)$$
$$I = \frac{1}{m}\sum_{i,j>i} ln(\frac{2m}{d_i d_j}) \tag{11}$$

$$MA_R = 4(\frac{R-R_{path}}{R_{clique}-R_{path}})(1-\frac{R-R_{path}}{R_{clique}-R_{path}})$$
$$MA_I = 4(\frac{I-I_{clique}}{I_{path}-I_{clique}})(1-\frac{I-I_{clique}}{I_{path}-I_{clique}}) \tag{12}$$
$$MA_g = MA_R * MA_I$$

I can be written as:

$$I = \frac{1}{m}\sum_{i,j>i} ln(\frac{2m}{d_i d_j})$$
$$I = \frac{1}{m}(\sum_{i,j>i} ln(2m) - \sum_{i,j>i} ln(d_i d_j)) \tag{13}$$
$$I = \frac{1}{m}\sum_{i,j>i} ln(2m) - \frac{1}{m}\sum_{i,j>i} ln(d_i d_j)$$

$$I = ln(2m) - R \tag{14}$$

$R_{path}, R_{clique}, I_{path}$ and I_{clique} represent the lowest redundacy, highest redundancy, highest mutual information and lowest mutual information of graphs with fixed n respectively. The equations can be found in section 2.4.1. Kim and Wilhelm suggested that network scientists may use $C = (R - R_{path})(I - I_{clique})$ as a complexity meassure, however, the upper bound cannot be found to normliase the complexity [16]. From our study, an upper bound of C can be calculated analytically.

Assuming the upper bound C_{max} can be found, $0 < C/C_{max} < 1$. As suggested in equation 14, $I = ln(2m) - R$, we can rewrite the complexity equation:

$$C = (R - R_{path})(ln(2m) - R - I_{clique}) \tag{15}$$

$$C = -R^2 + (ln(2m) - I_{clique} + R_{path})R + (-R_{path}ln(2m) + R_{path}I_{clique}) \tag{16}$$

Refereing to equation 16, the complexity function is a quadratic function with only variable R, which means, there is one and only one extreme. Considering the nature of product measure, it is safe to assume that the extreme is a maxima. To find the maxima, differentiates the function respect to $R(R_{max})$ where the function's slope is 0:

$$\frac{dC}{dR} = -2R_{max} + ln(2m) - I_{clique} + R_{path} = 0 \tag{17}$$

$$R_{max} = \frac{ln(2m) - I_{clique} + R_{path}}{2} \tag{18}$$

Even without assumption, $d^2C/dR^2 = -2$ implies the extreme is a maxima. As C_{max} is found, substitutes equation 18 into equation 15:

$$C_{max} = (R_{max} - R_{path})(ln(2m) - R_{max} - I_{clique})$$
$$C_{max} = (\frac{ln(2m) - I_{clique} + R_{path}}{2} - R_{path})(ln(2m) - \frac{ln(2m) - I_{clique} + R_{path}}{2} - I_{clique})$$
$$C_{max} = (\frac{ln(2m) - I_{clique} - R_{path}}{2})(\frac{ln(2m) - R_{path} - I_{clique}}{2}) \tag{19}$$

$$C_{max} = \frac{(ln(2m) - I_{clique} - R_{path})^2}{4} \tag{20}$$

Thus, using equation 20, a new normalized measure MA_{RI} can be defined using C/C_{max}:

$$MA_{RI} = \frac{4(R - R_{path})(I - I_{clique})}{(ln(2m) - I_{clique} - R_{path})^2} \tag{21}$$

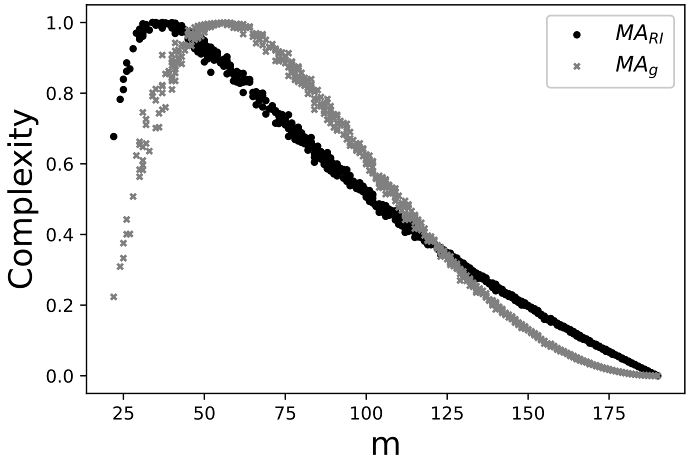

Figure 8: MA_g and MA_{RI} complexity of $G(n, m)$ networks, all networks have 20 nodes and 1000 samples with random m have been generated.

As shown in figure 8, MA_{RI} gives higher coplexity to sparser graphs but less complexity when approaching to medium number of links. Additionally, it decreases more linear and smoothly compare to MA_g.

3 Result

3.1 Complexity measures on small random graphs

To validate implemented measures, all measures are applied on $G(n, m)$ random graphs, with $n = 7$ and 50 sampels are generated for each m. As shown in figure 9, we reproduced results as Kim and Wilhelm did in [16]. Except $C_{2e,spec}$, as mentioned in section 2.3, there is a scaling problem. Most of the methods reaches its maximum briefly before medium number of edges. Low complexity are given to highly connected graphs and sparse graphs as expected.

Different subgraph measures perform similarly when the graph is neither sparse or strongly connected, there is a big difference between the maximum and minimum with same m. Thus, it is very difficult to predict the complexity of a graph with given m and n. The highest complexity is reached at $m = 15$ for $C_{1e,st}$ and $C_{1e,spec}$ and $m = 14$ for $C_{2e,spec}$. There is a miss in the plot: $C_{1e,spec}$ and $C_{2e,spec}$ plot does not contain a data point at (6,0). There is a very small proability for $G(n, m)$ model to generate a star graph(n-1 nodes are connected to 1 node, in total of n-1 edges), which will result in 0 complexity using $C_{1e,spec}$ and $C_{2e,spec}$ measure. Uses different subgraph measures for small graphs is optimal as it is relatively independent of m, but not for large graphs due to its complexity.

OdC is based on the node-node link correlation matrix of a graph.[17] Thus, it spreads across the space and has little relationship with m. OdC assigns a lot of graphs with 6 edges high complexity than desired. OdC is "hierarchy sensitive", it may not create big difference between graphs when the graphs are considerably small.

All 4 product measures are similar, gives higher complexity value at medium number of edges and less at both extremes. There is a very small difference between graphs with same number of edges. Ce and Cr tends to give highest complexity to graphs with exactly $n(n-1)/4$ edges, and MA_{RI} and MA_g reach their maximum before medium number of edges as expected. Product measure are highly depending on m, such that complexity of a graph can be approximated solely based on m and n. Network scientists may use machine learning techniques to approximate complexity of a graph using m and n in a small amount of time. On the other hand, product measure may not be optimal because a complexity measure should not solely based on m and n, but the overall structure of a network.

3.2 BA,WS and NW model

As informed in section 2.3, different subgraph measures have normalisation problem and the complextity would exceed 1.

Surprisingly, diffrent subgraph measures and product measures are struggling to seperate random graphs, WS graphs and NW graphs. Only OdC seperates random graphs and WS,NW model by giving random graphs higher complexity than WS and MW model with fixed m. This is because OdC awards graphs with complicated degree correlation.

More interesting results are obtained from BA graphs. Different subgraph measures assign lower complexity to BA graphs compare to random graphs. This can be caused by the preferential attachment. Preferential attachment ensures most nodes have low degree and builds hubs(nodes with high degree) in the graph. After cutting an edge/two edges between hubs and nodes with small degree, there is a high chance an isomorphic subgraph can be found, thus lower the complexity of the graph. In another word, subgraphs resulted by cutting the edge between hubs and node with small degree are very similar and occassionally isormophic.

MA_g and MA_{RI} perform similarly by assigning BA graphs lower value towards

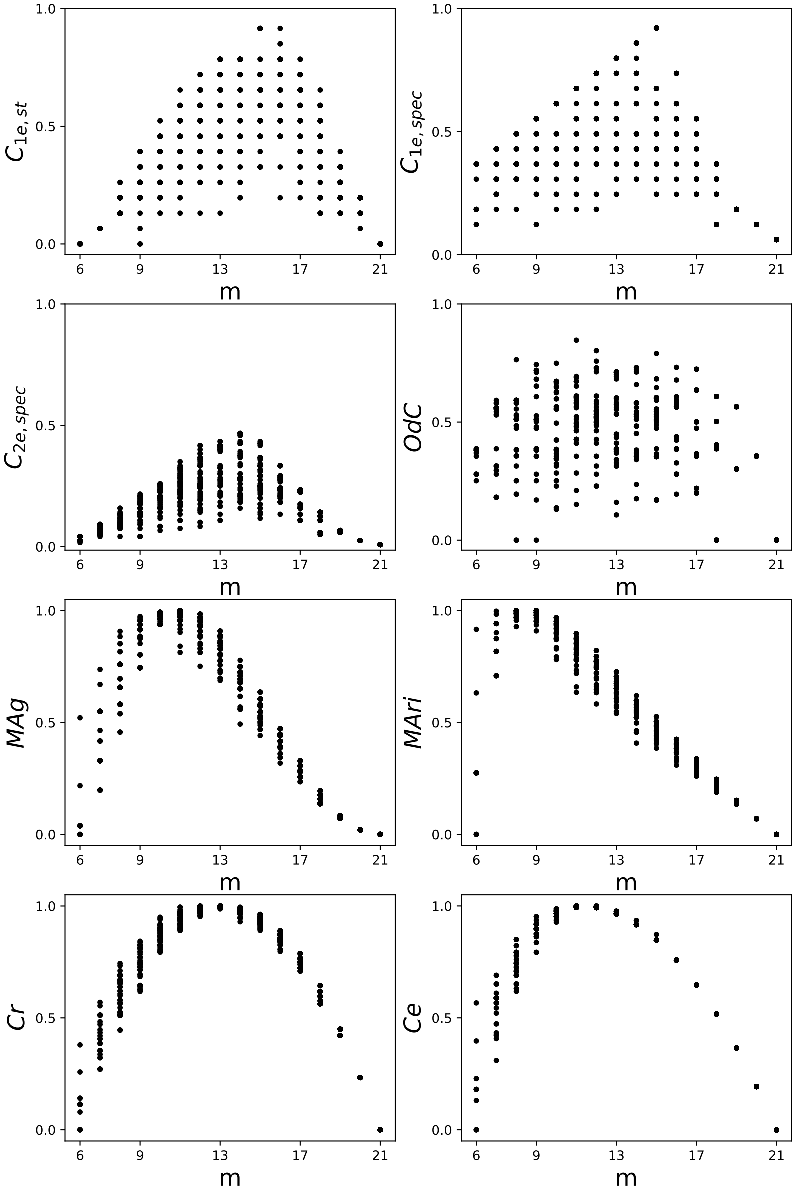

Figure 9: Complexity of graphs generated using $G(7, m)$ model, 50 samples are generated for each m. Methods from top-left to bottom-right are: $C_{1e,st}$, $C_{1e,spec}$, $C_{2e,spec}$, OdC, MAg, Cr, Cr and $MAri$.

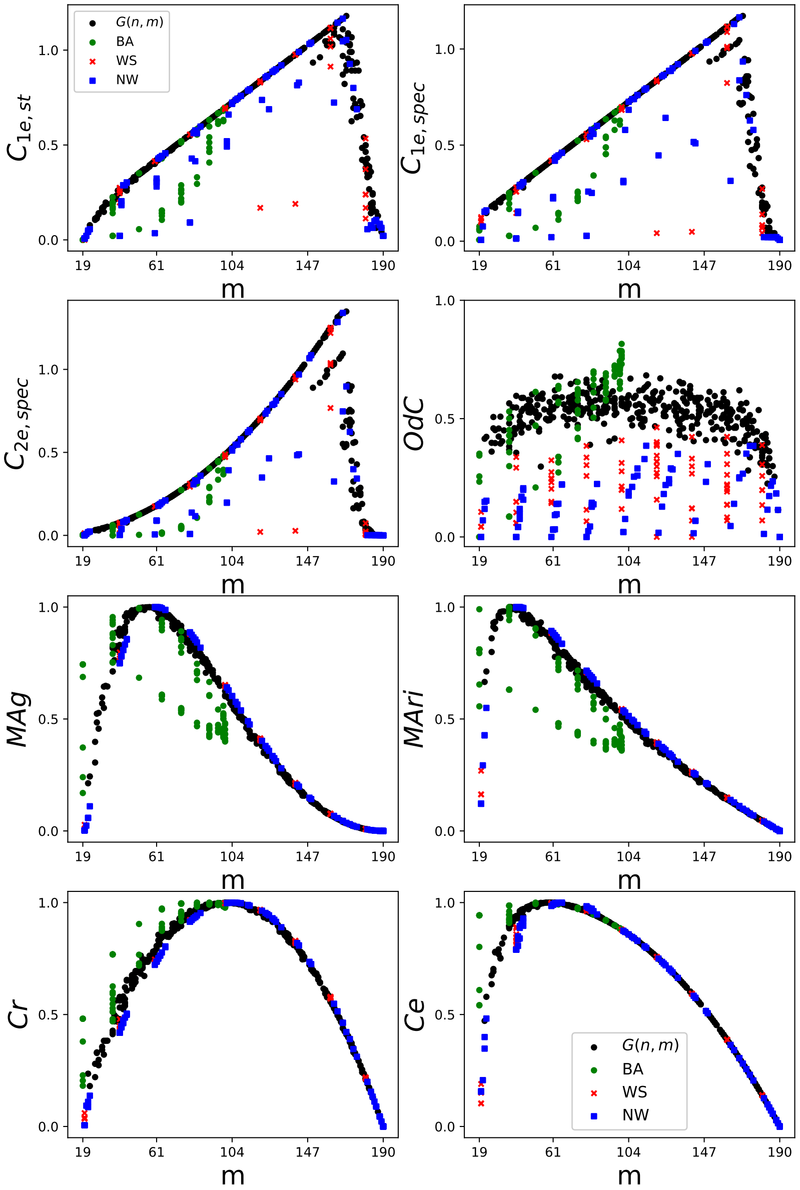

Figure 10: Complexity of 500 $G(n, m)$ graphs, 100 BA graphs, 100 WS graphs and 100 NW graphs, with $n = 20$. Graphs are generated according to section 1.6.

medium number of links. Both measures are depening on the variable $\sum_{i,j>i} d_i d_j$. BA graphs ususally have less $\sum_{i,j>i} d_i d_j$ because they are highly structure, causing less sum than a random graph. Contrarily, Ce and Cr cannot distinguish BA graphs and random graphs. Ce is based on the efficiency of a graph, a highly complex graph should have small average distance with not too much edges simultaneously. BA graph does not perform different than random graphs in Ce measure.

3.2.1 Configuration model

As introduced in section 1.4, a network is said to be scale-free if its degree distribution follows a power law distribution with $2 < \gamma < 3$. To observe how the change of γ would affect the complexity of a network, we need a model that can generate the graph with given γ. A configuration model [24] is able to turn a given degree distribution into a graph, which has the exact degree distribution as the given degree series.

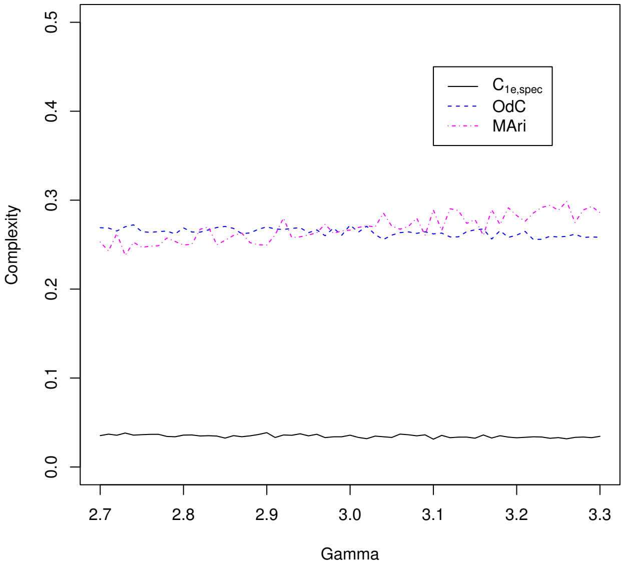

Figure 11: Complexities of graphs generated using configuration model, $n = 50$. Results are average of 50 simulations.

OdC and $C_{1e,spec}$ didn't change much by varying γ. MA_{RI} increases slightly as it is very sensitive to the change of degrees of nodes. Overall, varying γ does not impact the complexity by a lot. Due to limitation of our implementation, we can only generate a degree distribution within the range $2.7 < \gamma < 3.3$, by varying n and γ in a larger range, and adding a new variable n in the model, it can be more suggestful.

3.3 Complexity correlation

Different type of measures focus on different properties/parameters of a network, monitor the correlation between measures enable further understanding of the properties/parameters each measure is highlighting on. Additioanlly, network scientists may use more than one complexity measure on a network to determine whether they are truly "complex" or not. Combining complexity measures on networks will allow network scientists to comment on the network complexity from different aspect. For these reasons, we choosed three measures($C_{1e,st}$,OdC and MA_{RI}) and monitored their behaviours on random and BA graphs. The reason BA graphs are added is to investigate whether distinguish BA graphs and random graphs is possible or not.

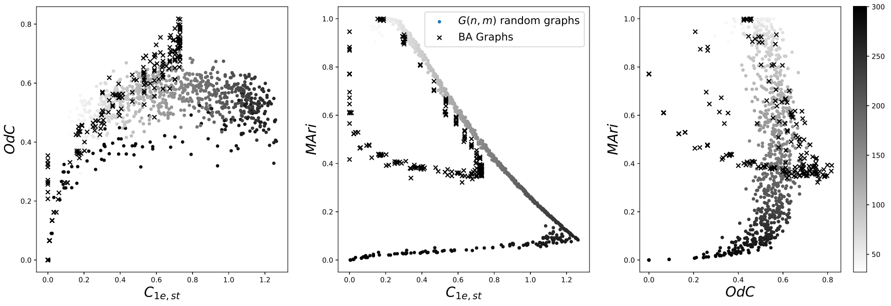

Figure 12: Correation of complexity measures on 1000 random graphs and 200 BA graphs with $n = 25$. Generated according to the rules stated in section 1.6. Colorbar represents change of m: higher the m, darker the datapoints.

As figure 12 suggests, the correlation between complexity measures are difficult to observe, since they are trying to address different things. As $C_{1e,st}$ value increases, OdC also increases. We indicated in section 3.1, m is not an important factor for OdC. On the other hand, $C_{1e,st}$ is highly based on m, the complexity increases linearly with m, and then complexity drops quickly after reaching the maximum. By constructing a correlation between $C_{1e,st}$ and OdC, seperate BA graphs and random graphs is not possible.

Since both complexity measure are heavily depend on m, thus we can observe a linear decreasing trend. We can detect a very unique distribution of the BA graphs' complexity: a shape between ellipse and half-moon. This is because both MA_{RI} and $C_{1e,st}$ assign lower complexity values to some of the BA graphs, compare to other BA graphs with same number of edges. Therefore, distinguish Ba graphs and random graphs is difficult.

The correlation between OdC and MA_{RI} of random graph seems simple: as m decrease, MA_{RI} decrease speedly but OdC change unnoticeably until the graph is highly connected. On the other hand, the correlation between OdC and MA_{RI} on BA graphs perform slightly different. As suggested, MA_{RI} distributes lower complexity to some BA graphs, and OdC is very consistent. In section 3.2.1, we did not observe a big change of complexity by varying γ. To observe a better result, using larger graph is optimal(for instance $n = 1000$), but due to technical issues(time complexity), they are not used here.

3.4 Complement graphs

Definition of a complement graph is fairly simple: the complement graph contains all the edges that are not in the original graph. To ensure complexity measures can be succesfully applied, graphs that will cause the complement graphs to be disconnected will be removed. Thus, in theory, the original graphs and the complement graphs are not exmtreme graphs. Analysing the complexity correlation between the original graph and the complement graph give us more inspirations of the measure. We have choose three different measures with different types: OdC, MA_{RI} and $C_{1e,spec}$. OdC seems to be very symmetric with respect to m. Since taking the completment graph is reversing the degree distribution, the consistency of node-node link relation is preserved. Causing the complexity value for original graph equal to the complement graph. As mentioned MA_{RI} is highly based on the number of edges, so observing a linear function is not a surprise. This alo applies to BA grapps. Similar to MA_{RI}, $C1_{1e,st}$ is more keen to m, causing another negative correlation between the original graph and the complement graph.

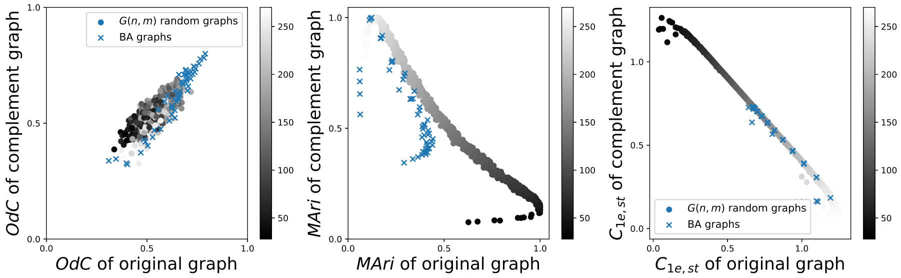

Figure 13: Complexities of the original graphs and complement graphs with $n = 20$. There are two types of graphs are generated: 500 $G(n, m)$ random graphs and 100 BA graphs.

3.5 Applying MA_{RI} on real networks

To test the new measure MA_{RI}, 6 real networks are collected. To ensure comprehensive evaluation, various types of networks are used. To be applicable, bus networks in 6 cities are collected and converted to networks from raw. Bus networks are different to other real networks. Bus networks contain extraordinary amount of nodes with degree 2. For simplification and general interest, bus networks are modified. In modified bus networks, all nodes with $k = 2$ are removed, whereas the edges are preserved. For instance, node b is only conneted to a and c. Node b will be removed from the network and a new edge (a, c) will be added to the network. The modification will be iterated multiple times until bus networks are free from node with degree $k = 2$. Modification will significantly decrease the distance of bus networks. Moreover, generated graphs are also added to be evaluated and compared.

To be mentioned, we will refer these non-bus networks as real networks, for convinience.

Label	Name	Type	n	m	L	L_r	MA_{RI}	OdC
Real networks								
1	Dolphins[25]	Animal interaction	62	159	3.357	2.524	0.999	0.517
2	PDZBase [26]	Protein interaction	161	209	5.326	5.326	0.824	0.310
3	Hamsterster[27]	Online social network	874	4003	3.217	3.058	0.963	0.532
4	Roget's Thesaurus [28]	Synonym network	994	3640	4.075	3.466	0.960	0.392
5	Flight[29]	Flight network	3397	19230	4.103	3.350	0.948	0.525
6	UK train [30]	UK train network	2490	4377	10.384	6.220	0.664	0.233
Bus networks								
7	London[30]		8653	12285	32.338	8.687	0.38	0.127
8	Paris[31]		10644	12309	47.631	11.059	0.173	0.065
9	Berlin[31]		4316	5869	33.284	8.366	0.358	0.134
10	Sydney[31]		22659	26720	36.131	11.688	0.173	0.064
11	Detroit[31]		5683	5946	70.513	11.708	0.062	0.020
12	Beijing[32]		9249	14058	27.891	8.214	0.441	0.167
Modified Bus								
13	London		3417	6018	18.308	6.462	0.553	0.128
14	Paris		2762	4301	15.386	6.975	0.468	0.106
15	Berlin		1662	2941	18.36	5.867	0.586	0.149
16	Sydney		4834	8358	17.665	6.838	0.49	0.089
17	Detroit		295	483	6.341	4.794	0.643	0.117
18	Beijing		4072	8325	14.864	5.902	0.64	0.195
Generated networks								
19	G(n,m) random network($n = 875, m = 4000$)		875	4000	3.313	3.061	0.970	0.330
20	WS network($n = 875, k = 10, p = 0.05$)		875	4375	5.000	2.942	0.974	0.123
21	NW network($n = 875, k = 10, p = 0.05$)		875	4587	5.075	2.883	0.980	0.089
22	BA random network($n = 875, m = 5$)		875	4350	2.922	2.950	0.999	0.358

Table 1: Label,description and parameters of used networks

After careful consideration, average distance ratio L/L_r is chosen to be the variable showin on x-axis in figure 14. Average distance is a more important parameter to consider about because it can identify how ”small-world” the network is. As suggested, all the bus networks have significantly higher average distance ratio than real

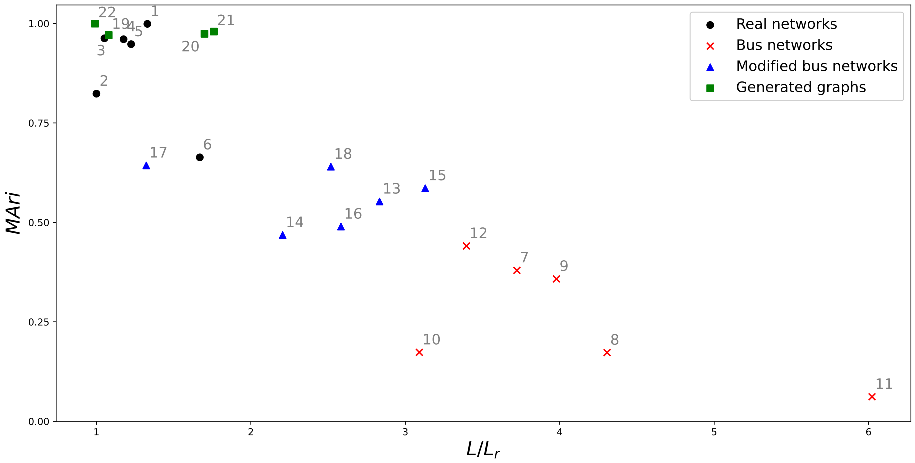

Figure 14: MA_{RI} complexity of real networks, bus networks, modified bus networks and graphs generated by graph models, labelling and description can be found in table 1.

networks. Even after modified, the average distance ratios are still relatively high. But higher average distance ratio brings less MA_{RI} complexity. We can suggest a negative correlation between the average distance ratio and the MA_{RI} complexity. There is an exception of real networks, which is the UK train network. Cosidering the similarity between bus networks and train networks, it is not surprising. Generated graphs also behaves similar to standard real networks; low average distance ratio with high complexity, as they are intended to simulate the behaviour of standard real networks. In theory, ”Flight” is also a public transport, so it should behave similar to bus networks or the train network. However, the ”Flight” network performs unlike transport networks. Flight networks are designed to be robust[33] to be functional under severe whether/accident. Unlike trains or buses, flight are less limited by physical path in the sky.

To discuss the cause of increase of MA_{RI} complexity after modified, we need to consider about the parameter MA_{RI} focusing on. $\sum_{i,j>i} d_i d_j$ is the only variable that affects the complexity of a graph. The essence of the measure is calculating the average $d_i d_j$. By removing nodes with degree 2, average $d_i d_j$ will be increased, and leads to an increase of the MA_{RI} complexity.

3.6 Rewiring on real networks

As recommened in section 1.3, rewiring is an important technique to monitor the behaviour of a network. Hence, we rewired real networks and modified bus networks to record their behaviour. Both single link rewiring and pairwise rewiring will be used.

The reason we are going to use OdC instead of MA_{RI} is that pairwise rewiring will keep the degree relationship, and MA_{RI} will stay unchanged.

How we rewire graphs:

- The rewiring is done gradually. Initially, graph G will be rewired with probability $p = 0.05$. After recording the results, we rewire it with probability $p = 0.05$ again. This step will be repeated 20 times.
- Both rewiring have been simulated on all graphs more than 10 times(”dolphins” and ”PDZBase” have been rewired more than 100 times, as they are relatively small), to generate more consistent results.

From figure 15, we can concludes that most real networks behaves similar using rewiring. As mentioned in section 1.3, single link rewiring results in higher randomness, because it destroys the degree distribution. In theory, when $p = 1$, the graph becomes a random network. OdC decreases significantly with the increase of rewiring probability for real networks, whereas the change of complexity is not large for pairwise rewiring. This is because OdC is highly sensitive to degree relationship, but pairwise rewiring does not change the degree relationship. Also, the error bar for ”dolphins” and ”PDZBase” is large. This is simply because they are small networks, rewiring will make larger impact than larger graphs.

There is an expcetion in real networks; the UK train network performs similar to bus networks rather than real networks. As public transportation networks, single link rewiring will cause the complexity to increase. Generally, the increase of complexity becomes small after $p = 0.4$; almost half of the links have been rewired. As shown in table 1, OdC complexities are very low for all the bus networks. As introduced by Claussen[17], OdC assign high complexity to graphs that have no preference for the degree of their neighbours. After destroying the degree correlation using single link rewiring, the OdC complexity will increase.

In contrast, pairwise rewiring will cause the complexity to decrease briefly like real networks, with similar reason.

4 Conclusion and further study

In this theis, we investigated and compared difference of complexity measures. Different complexity measure focuses on different properties and parameters of a graph. For example, different subgraph measures focusing on the general structure of subgraphs, leads to high complexity and not feasible to apply difference subgraph measures on large networks. In addition, the floating point algorithmetic is imperfect when dealing with large numbers and decimals. On the other hand, product measures are faily simple in terms of calculation time. Espectiall Cr, the easist complexity measure, which can be calculated in $O(n)$ time[16]. A drawback of product measures is that they are highly based on m, creating smaller difference than other complexity measures with fix m and n. OdC measure can distinguish random graphs and graphs generated using BA,WS and NW model, and within relatively small amount of time. However, the complexity value are spreaded. We suggest to use the measure that is most suitable, depending on which specific parameters or properties are most important in the problem.

Additionally, we constructed a new measure MA_{RI} based on the idea of MA_g. MA_{RI} assigns higher complexity to sparser graph than MA_g. To be noticed, to calculate MA_{RI}, R_{clique} and I_{path} are not required, but m is involved in the calcualtion.

We also compare the difference between real networks and bus networks, and we investigated the unique property of bus networks: high average distance with low complexity, which is the opposite of a small-world network. This causes averagely lower complexities than other real networks.

There are more complexity measures [34][35] for further studies and researches. We encourage study and application of the measures on different type of graphs and monitor the behaviour of different complexity measures. Further studies on complexity measures could help the network science community to build a comprehensive, robust and applicable complexity meausre which everyone can agree on.

An unexpected fact was observed: degree based measures(OdC,MA_g,MA_{RI}) generally assign relatively high complexity to very sparse graphs, except Ce. We are not sure whether this is a coincidence or not, but this can be a topic to further investigate.

Overall, we hope our result can stimulate more studies on network complexity meausre. The definition of complexity is already difficult to be determined. A good complexity measure should be able to distinguish: random networks, small-world networks, scale-free networks and real networks. In addition, highest complexity to graphs with number of edges slightly less than the medium. Our work provides useful aspect and observation to build a optimal complexity measure in the future.

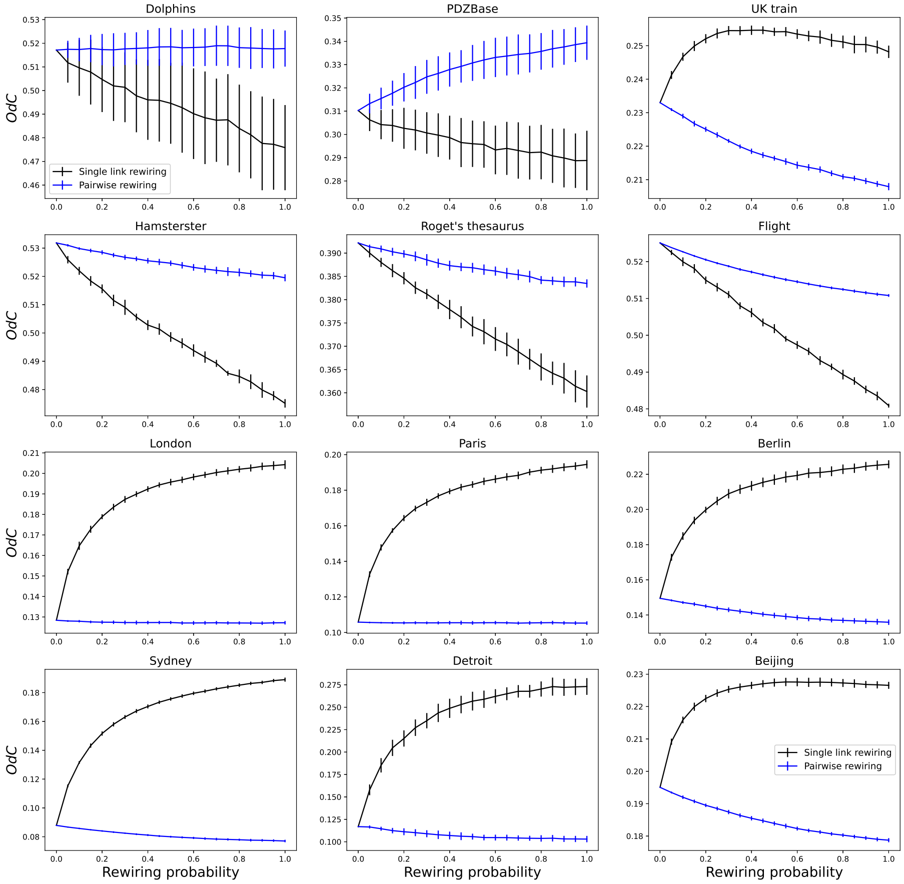

Figure 15: Change of OdC complexities respect to change of rewiring probaility with an error bar(one standard deviation).