

Robustness and Optimization of Complex Networks

Reconstructability, Algorithms and Modeling

Robustness and Optimization of Complex Networks

Reconstructability, Algorithms and Modeling

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 25 maart 2013 om 12:30 uur

door

Dajie Liu

Master Universitario di II livello, Politecnico di Torino, Italië
geboren te Suqian, Jiangsu provincie, China.

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. P.F.A. Van Mieghem

Samenstelling promotiecommissie:

Rector Magnificus,	Voorzitter
Prof.dr.ir. P.F.A. Van Mieghem,	Technische Universiteit Delft, promotor
Prof.dr.ir. R.E. Kooij,	Technische Universiteit Delft
Prof.dr.ir. M.P.C. Weijnen,	Technische Universiteit Delft
Prof.dr. C.J. Stam,	VU University Medical Center
Prof.dr.ir. P.M.A. Slood,	Universiteit van Amsterdam
Prof.dr. D. Stevanović,	University of Primorska

ISBN 978-90-79787-45-6

Published and distributed by:

Next Generation Infrastructures Foundation

P.O. Box 5015, 2600 GA Delft

The Netherlands

E: info@nextgenerationinfrastructures.eu

I: www.nextgenerationinfrastructures.eu

This research has been funded by the Next Generation Infrastructures Foundation.

Keywords: Complex Networks, Graph Spectra, Reconstructability, Line Graphs, Social Networks, Community Structure, Hypergraphs, Brain Networks, Randomness

Copyright © 2013 by D. Liu

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author or Delft University of Technology.

To my family

Contents

1	Introduction	1
1.1	Network reconstruction from spectra	2
1.2	Graph energy	3
1.3	Inverse line graph algorithms	5
1.4	Random line graphs	8
1.5	Social networks with community structure	9
1.6	Randomness of brain networks	11
I	Reconstructability and Energy of Networks	13
2	Spectral Perturbation and Reconstructability	15
2.1	Spectral Perturbation	15
2.1.1	Description and definition of reconstructability	15
2.1.2	Theory	17
2.2	The Reconstructability of random graphs	19
2.2.1	Weibullian probability distribution of $\theta_p(N)$	19
2.2.2	$E[\theta_p(N)]$ as a function of N and p	19
2.3	The Reconstructability of Other Networks	21
2.3.1	Scale-free networks	22
2.3.2	Small-world networks	23
2.3.3	Deterministic graphs	24
2.3.4	Summary	26
2.4	The Linear Scaling Law With Eigenvector Perturbation	26
2.5	Chapter conclusion	27
3	Graph Energy	29
3.1	Graph energy vs. m_0	29
3.1.1	Deterministic graphs	29
3.1.2	Random graphs	30
3.2	Graph energy vs. m_{-1}	30

3.3	Influence of assortativity on graph energy	33
3.3.1	Random graphs	33
3.3.2	Grid graph	35
3.3.3	Random trees	35
3.3.4	Small-world graphs	35
3.3.5	Scale-free graphs	37
3.4	Chapter conclusion	37
 II Line Graphs and Root Graphs		39
4	MARINLINGA	41
4.1	Link adjacency matrix (LAM) and line graph	41
4.2	Properties of the LAM	42
4.2.1	The basic forbidden link adjacency patterns in a LAM	44
4.3	MARINLINGA	46
4.3.1	Matrix relabeling	46
4.3.2	Construction algorithm	50
4.3.3	Worst case complexity of MARINLINGA	58
4.4	Chapter conclusion	59
5	ILIGRA	61
5.1	Notation	61
5.2	Concept	62
5.3	Theoretical preliminaries	63
5.4	Algorithm description	65
5.5	Complexity	69
5.6	An example	69
5.7	Evaluation	71
5.7.1	The link density of line graphs	74
5.8	Chapter conclusion	75
6	Random Line Graphs	77
6.1	Theoretical preliminaries	78
6.1.1	Formation of line graphs	78
6.1.2	The bandgaps of the number of links L in line graph $H(N, L)$	79
6.2	A random line graph model	84
6.3	The assortativity of (H, G) during the merging process	86
6.3.1	Random line graphs with cliques of the same size $s_j = S$ for $j = 1, 2, \dots, C$	86
6.3.2	Heterogeneous random line graphs with cliques of different sizes	93

<i>CONTENTS</i>	ix
6.4 Chapter conclusion	95
III Social Networks and Brain Networks	97
7 Social Network Modeling	99
7.1 The representation of social networks	100
7.1.1 Preliminaries	100
7.1.2 Hypergraph representation	101
7.1.3 An illustrative example	102
7.2 Properties of social networks	103
7.2.1 Topological properties	103
7.2.2 Spectral properties	105
7.3 Characterizing the real-world social networks	106
7.3.1 ArXiv coauthorship networks	106
7.3.2 IMDB actor collaboration network	107
7.3.3 The SourceForge software collaboration network	108
7.4 Modeling of social networks	109
7.4.1 Model description	109
7.4.2 Properties of the growing hypergraph model	112
7.5 Chapter conclusion	116
8 Randomness of Brain Networks	117
8.1 Spectral randomness metric	117
8.2 Metrics partially indicating randomness	119
8.3 Randomness of small-world graphs	120
8.3.1 Non-repetitive rewiring	120
8.3.2 Randomness of small-world graphs	120
8.4 Randomness of brain networks	121
8.4.1 Description of brain networks	121
8.4.2 The structure coefficient of brain networks	123
8.5 Chapter conclusion	125
9 Conclusions	129
9.1 Contribution summary	132
9.2 Future work	132
A Properties of the matrix E_k	135
B Quality of the bound of Ξ	137

C	The Initialization of MARINLINGA When $s_3 = 0$	139
C.1	When $s_1 = 1$	139
C.2	When $s_1 = 2$	139
C.3	When $s_1 = 3$	140
C.4	When $s_1 \geq 4$	143
	C.4.1 When $s_2 \geq 3$	143
	C.4.2 When $s_2 \leq 2$	143
D	Proof of the adapted semicircle law	147
E	Notation	149
	Bibliography	151
	Samenvatting (Summary in Dutch)	159
	Acknowledgements	161
	Curriculum Vitae	163
	Publications by the author	165
	NGInfra PhD Thesis Series	167

Thesis Summary

The infrastructure networks, including the Internet, telecommunication networks, electrical power grids, transportation networks (road, railway, waterway, and airway networks), gas networks and water networks, are becoming more and more complex. The complex infrastructure networks are crucial to our human society, and it has been a hot research field to make our complex infrastructure networks more robust and optimize the performance of them. Besides man-designed infrastructure networks, complex networks also cover many natural networks, such as social networks, ecological networks, and biological networks. In order to tackle some of the difficult social issues, ecological problems, and unsolved medical problems, we must learn how these natural complex networks organize, operate, and function.

Complex networks can be represented by graphs. A graph consists of a collection of nodes and a collection of links that connect the nodes. A graph is uniquely described by its adjacency matrix, of which the entry on row i and column j is one only if node i and node j in the graph is connected by a link, otherwise the entry is zero. Each adjacency matrix is associated to a unique set of eigenvalues and corresponding eigenvectors. The eigenvalues and corresponding eigenvectors of a graph, also called the spectrum of the graph, contains all the information of the graph, and the topological/physical meanings of some eigenvalues and eigenvectors are already known. The knowledge on the spectra of networks is of crucial importance to the many aspects of the researches on complex network, such as connectivity of networks and virus spreading in networks. The line graph $l(G)$ of a graph G has a set of nodes mapping the set of links in G , and two nodes in $l(G)$ are adjacent if and only if the corresponding links in G have a node in common. Some problems of graphs can be transformed to much easier ones in the domain of line graphs. For example, partitioning the nodes to find the overlapping communities in a graph can be done by partitioning the links in the line graph of the concerned graph. Moreover, the line graphs often share common features with real-world complex networks, like highly clustered and assortative mixing. Hence, the line graphs are considered by many to model real-world complex networks.

The robustness and optimization of complex network is a rather broad research field. We focus on the reconstruction of complex networks from the spectral domain and the line graph domain. This thesis is organized as follows. We first study the

reconstruction of networks from their eigenvalues and eigenvectors and the spectral properties of networks. In the second part of this thesis, we present two algorithms which reconstruct networks from the line graph domain, the properties of the line graphs, and a random line graph model. We at last give the research results on two types of real-world networks.

The adjacency matrix of a graph can be computed with its eigenvalues and eigenvectors. When some of the eigenvalues are set to zero, the adjacency matrix can still be correctly computed. We propose a measure, the reconstructability coefficient, defined as the maximum number of eigenvalues that can be removed. We find that the reconstructability coefficient is linear function of the size of the network for all networks that we have studied. We give some results on the spectral metric, the energy of a graph, which is defined by the sum of the absolute value of all the eigenvalues. We also explore the relations between graph energy and the topological metric, assortativity, for many different types of networks.

For the reconstruction of networks from the line graph domain, we propose two algorithms MARINLINGA and ILIGRA. While all previous algorithms rely on Whitney's theorem, MARINLINGA is based on the principle of link relabeling and endnode recognition. ILIGRA reconstructs the graphs from the line graph domain with the linear time complexity. This thesis extends the researches in the line graph domain. We find that the number of links in a line graph with a fixed number of nodes can not take some consecutive natural numbers, and these numbers are called a bandgap of the line graph. We present the exact expressions of the bands and bandgaps of the number of links in line graphs. In order to facilitate the researches in the line graph domain, we propose a model which randomly generates line graphs. The essence of our model is to merge step by step a pair of nodes in cliques, subjecting to some rules to ensure that the resulting graphs are line graphs. Thanks to the random line model, a method to generate a serial of graphs of which the assortativity increases linearly has been invented.

This thesis studies two types of real-world networks: social networks and human brain networks. We characterize the overlapping community structure of the social networks of ArXiv coauthorship, IMDB actors collaboration and SourceForge collaboration, and propose a growing hypergraph model, based on preferential attachment. The proposed hypergraph model captures the fundamental properties including the power-law distributions of group size, group degree, overlapping depth, individual degree and interest-sharing number of real-world affiliation networks, and reproduces the properties of high clustering, assortative mixing and short average path length of social networks. To study brain networks, we propose a spectral randomness metric to quantize the randomness of networks. Based on the randomness measuring method, we have found that the brain networks of Alzheimer's disease are statistically more random than the healthy brain networks.

Chapter 1

Introduction

Complex networks, as an interdisciplinary subject, lie at the intersection between graph theory and statistical mechanics. The research of complex networks originated from the pioneering works [1][2][3][4][5][6][7] on percolation and random graphs in the middle of last century. However, complex networks have not been a focus of attention until the end of 20th century. The last decade has witnessed a rapid advancement in the studies of complex networks which help to understand dynamic behavior of real systems coming from different fields such as biology [8][9][10] (food-web, nervous system, cellular metabolism, protein-protein interaction network, gene regulatory networks), social systems [11] (scientific collaboration, citation, linguistic networks, and technological systems), Internet [12], World Wide Web [13], power-grid [14] etc. These real-world complex networks appear to possess many features that random graph model fails to capture, such as highly clustering, assortative\disassortative mixing, power law degree distribution, and community structure. Among many fruitful results, Watts and Strogatz's investigation of small-world networks [15], Barabási and Albert's scale-free graph model [16], Newman's mixing patterns [17][18], and Girvan and Newman's community structure in networks [19][20] are mostly known to the research community. Investigations on complex networks help us understand and predict the behavior of real-world complex systems and infrastructures, also enable us to make complex systems more robust and heighten the performance of the complex networks of infrastructures.

In this chapter, we give the outline of the main course of this thesis. There are many hot aspects in the research field of robustness and optimization of complex networks. In the first part, we first study the reconstructability of networks from their eigenvalues and eigenvectors when partial information of the spectra is lost. We find that the reconstructability is linear with the sizes of networks. Then, we give some empirical results on the spectral metric, energy of networks. In the second part of this thesis, we first present two algorithms which reconstruct networks from the line graph domain. Then, we show some properties of the line graphs, and we particularly find that the number of links of line graphs cannot take some sets of consecutive natural numbers.

At last, we propose a random line graph model. In the third part, we first study the overlapping community structure of some real-world social network and propose a growing hypergraph model which reproduces the overlapping community structure. Then, we propose a spectral randomness measure, and based on the spectral randomness measure, we statistically differentiate the brain networks of Alzheimer’s disease and the healthy brain networks.

1.1 Network reconstruction from spectra

“Robustness” or “resilience” properties of complex networks are commonly analyzed by perturbing the network [21][22]. The simplest perturbations, that we call elementary changes, are the omission/addition of a link and/or a node or the rewiring of links. Any topological perturbation can be constructed as a sequence of elementary changes. The degree of degradation of the network performance, measured in terms of some sets of graph metrics, under a certain topological perturbation is commonly regarded as a measure of the robustness of that network. Service perturbations are usually much more complex to define and to analyse, because they consist of changes in traffic, latency, availability, and many other unknown processes. Spectral graph theory has been applied to understand properties of networks[23][24]. Meanwhile, the eigenvalue perturbation method has been well developed and applied in quantum mechanics [25][26]. Another spectral operation, the Karhunen Loéve transform (KLT), is widely used in the area of image data compression [27].

We explore the effect of perturbing the spectra of complex networks. The eigenvalues of most complex graphs are nondegenerate, and their corresponding eigenvectors are unique. As explained in Section 2.1.1 and provided all eigenvectors are known, we remove eigenvalues from the spectrum of a graph and check whether the graph can still be exactly reconstructed, by exploiting the zero-one nature of the elements of the adjacency matrix. The requirement of *exact* reconstruction is unique and in contrast to, for example, image compression, where some image information is lost. Removing eigenvalues can be regarded as a particular type of spectral perturbation. On the other hand, the eigenvectors can also be perturbed by, for example, adding noise to each component of the eigenvectors. One could as well consider a metric such as the connectivity, hop-count, diameter, clustering coefficient etc., and investigate its degradation under such spectral perturbation (eigenvalue removal and noise addition to eigenvectors).

A large number of topological measures have been derived (see e.g. [28]) from the adjacency matrix and Laplacian to capture different features of a network as well as to classify networks. Examples are the distribution of the nodal degree, which is the number of links incident to a node, and the average clustering coefficient, which describes the link density among the direct neighbors of a node. These measures can be highly correlated, pointing to a certain level of redundancy among them. It is highly

desirable to determine a specific small set of independent measures that is sufficient to characterize the network structure. However, the questions, “which measures possess more information regarding to the topology” and “what is the level of redundancy or correlation between measures”, is far from understood.

We examine the maximal number of eigenvalues that can be removed from the spectrum given the set of eigenvectors. The method and related theory are explained in Section 2.1. We define the reconstructability coefficient θ of a network as the maximum number of eigenvalues that can be set to zero, given that the adjacency matrix can be reconstructed exactly. Via extensive simulations on different classes of graph, presented in Section 2.2 and 2.3, we found the remarkable linear scaling law

$$E[\theta] = aN \quad (1.1)$$

where the real number $a \in [0, 1]$ depends on the graph G . Moreover, the variance $\text{Var}[\theta]$ was sufficiently smaller than the mean $E[\theta]$ such that $E[\theta]$ serves as an excellent estimate for θ . For sufficiently large N , the law (1.1) tells us that a portion a of the smallest eigenvalues (in absolute value) can be ignored or removed from the spectrum and that the adjacency matrix is still reconstructable (provided we have the exact eigenvectors). Section 2.4 investigates the sensitivity of the reconstructability coefficient θ under random perturbation of the eigenvectors in the class of Erdős-Rényi random graphs and concludes that the perturbation can be as high as 10% (the norm of an eigenvector is 1). The reconstructability coefficient θ (or the scaled one coefficient $a = \frac{E[\theta]}{N}$ in (1.1)) can be regarded as a spectral metrics of the graph, that expresses how many dimensions of the N -dimensional space are needed to represent or reconstruct the graph. Roughly, a high reconstructability coefficient θ reflects a “geometrically simple” graph that only needs a few orthogonal dimensions to be described. The precise physical or topological meaning of the reconstructability coefficient θ is not yet entirely clear.

1.2 Graph energy

The energy of a simple graph has been studied by various authors [29][30][31][32]. The energy E_G of a graph G , is defined as $E_G = \sum_{j=1}^N |\lambda_j|$, where λ_j is the j th eigenvalue of the adjacency matrix A of the graph G . The energy E_G of a graph $G(N, L)$ with N nodes and L links is bounded [33] by

$$\sqrt{2L + N(N-1)|\det A|^{2/N}} \leq E_G \leq \sqrt{2NL}$$

The multiplicity of zero eigenvalues of adjacency matrix is denoted by m_0 , and the multiplicity of -1 eigenvalues of adjacency matrix is denoted by m_{-1} . Assortativity is a graph metric which measures how strong the tendency is that the nodes connect to

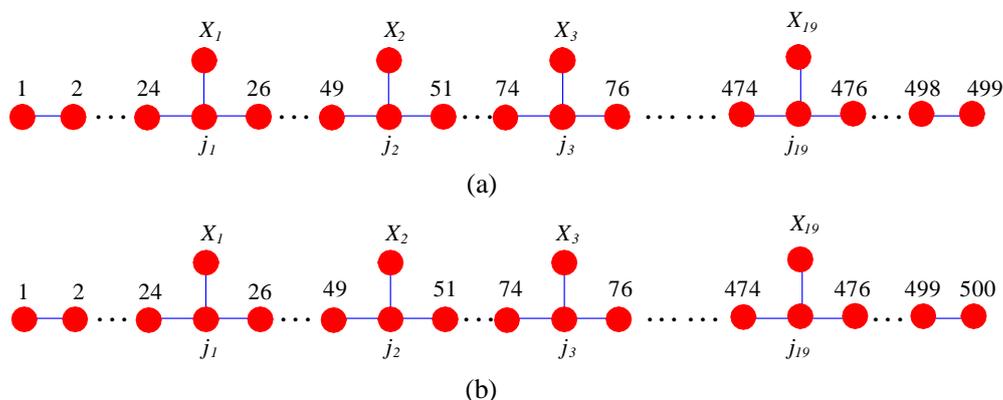


Figure 1.1: The molecular graphs introduced by Gutman et al. [29]. The graphs are constructed by adding 19 nodes X_1, X_2, \dots, X_{19} , to a long path graph and connecting by a link node X_k and node j_k , where node j_k is the $(25k)$ th node on the path, for example, $j_1 = 25$ and $j_2 = 50$. (a) There are 499 nodes on the path. By shifting the nodes $X_k, k = 1, 2, \dots, 19$ one step leftward or rightward, the graphs with $m_0 = 0, 2, 4, \dots, 20$ can be obtained. (b) There are 500 nodes on the path. By shifting the nodes $X_k, k = 1, 2, \dots, 19$ one step leftward or rightward, the graphs with $m_0 = 1, 3, 5, \dots, 19$ can be obtained.

other nodes with similar degree. The assortativity coefficient of a graph is denoted by ρ_D . When $\rho_D > 0$, the nodes of the graph connect preferentially to other nodes with similar degree, and the graph is said to be assortative; When $\rho_D < 0$, the pair of adjacent nodes tend to have different degrees, and the graph is said to be disassortative. This report studies the relations between the energy E_G of a graph and the just mentioned three metrics, m_0 , m_{-1} , and ρ_D , under the constraint that the number N of nodes and the number L of links are constant.

As shown in Figure 1.1, Gutman et al. [29] constructed two groups of molecular graphs: (a) The first group of graphs $G_k, k = 1, 2, \dots, 11$, have 518 nodes and 497 links, and the graphs were constructed in such a way that $m_0(G_k) = 2(k - 1)$; (b) The second group of graphs $G_k, k = 1, 2, \dots, 10$, have 519 nodes and 498 links and satisfy that $m_0(G_k) = 2k - 1$. It was shown by computations that the graph energy decreases almost linearly in k for both groups of graphs.

1.3 Inverse line graph algorithms

A simple graph¹ with N nodes and L links is denoted by $G(N, L)$. The line graph $H = l(G)$ of a graph G is a graph [34] in which every node in H corresponds to a link in G and two nodes are adjacent if and only if their corresponding links have a common node in G . The graph G is called the *root graph* of H . The complete graph with three nodes K_3 is a line graph, which has two different root graphs, K_3 and $K_{1,3}$. Except for K_3 , Whitney's theorem [35] states that, all line graphs have only one root graph (isomorphic graphs are considered as the same graph). Clearly, the number N_H of nodes in H equals the number L of links in G . The theoretical basis for the line graph was given by Whitney's theorem on the link isomorphism of two graphs [35][36]: if connected graphs G_1 and G_2 have isomorphic line graphs, G_1 and G_2 are isomorphic unless one is K_3 and the other is $K_{1,3}$. Cvetković *et al.* [37] reviewed the state-of-the-art knowledge about line graphs.

There exist plenty of real-world networks that can be modeled by line graphs [38][39]. Given M clubs and N students at an university, every student joins two clubs. Each student has different choices (we assume that there are enough clubs). We define two networks G_1 and G_2 . The M clubs are the nodes of G_1 and two nodes are adjacent if two clubs have the same student as their member. The N students are the nodes of G_2 and two nodes are adjacent if two students belong to the same club. Clearly, G_2 is the line graph of G_1 . Such pairs (G_1, G_2) are common in on-line social networks like Facebook, Twitter and etc., where users join the special groups where they share the same interest with others. A graph is assortative if its low-degree nodes tend to be adjacent with other low-degree nodes and its high-degree nodes tend to be adjacent with other high-degree nodes. Line graphs are assortative and clustered [40][41][38][42]. Instead of partitioning the links of the root graph for the overlapping communities detection, it has been suggested to equivalently detect the communities in the line graph of the concerned network [43][44][45]. Computing the line graph of a graph and constructing the root graph of a line graph also play an important role in bond percolation threshold predictions [46], and it enables us to compare the properties of a random line graph and its root graph.

The following formula [34] can be used to compute the adjacency matrix of the line graph $l(G)$ of a graph G ,

$$A_{l(G)} = (R^T R)_{L \times L} - 2I \quad (1.2)$$

where R is the incidence matrix of the undirected graph G . If link j is incident to node i , the entry r_{ij} of R is 1, otherwise 0. In each column there are exactly two 1-entries. To facilitate the applications of line graphs, the construction of a line graph H from a root graph G and the inverse construction from the line graph $H = l(G)$ to the root graph

¹A simple graph is an unweighted, undirected graph containing no self-loops (links starting and ending at the same node) nor multiple links between the same pair of nodes.

G are necessary. Constructing the root graph is far more complex than computing the line graph. Before constructing the root graph from a given graph, it is important to know whether the graph is a line graph. Up till now, the following criteria for a graph to be a line graph exist in the literature:

- A graph is a line graph if and only if it is possible to find a collection of cliques in the graph, partitioning all the links, such that each node belongs to at most two of the cliques (some of the cliques can be a single node) and two cliques share at most one node [36]. If the graph is not K_3 , there can be only one partition of this type.
- A graph is a line graph if and only if it does not have the complete bipartite graph $K_{1,3}$ as an induced subgraph, and if two odd triangles² have a common link, the subgraph induced by their nodes is the complete graph K_4 [47].
- A graph is a line graph if and only if none of the nine forbidden subgraphs (see Figure 1.2) is an induced subgraph of it [48].
- A graph is not a line graph [34] if the smallest eigenvalue of the adjacency matrix (1.2) is smaller than -2 .

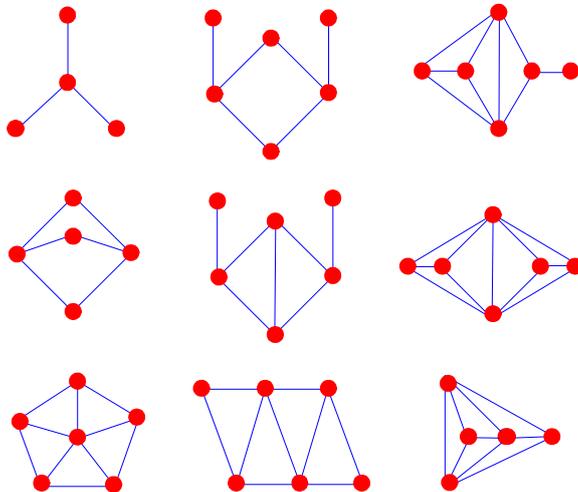


Figure 1.2: The nine forbidden subgraphs for line graphs [48].

Based on the above criteria and Whitney's theorem, several algorithms for constructing the root graph have been proposed. Two algorithms for line-to-root graph

²If every node is adjacent to two or zero nodes of a triangle, it is an even triangle.

construction were proposed simultaneously by Roussopoulos [49] and Lehot [50]. The algorithm of Roussopoulos is based on the theorem of Krausz [51]: A graph is a line graph if and only if it is possible to find a collection of cliques in the graph, partitioning all the links, such that each node belongs to at most two of the cliques (some of the cliques can be a single node) and two cliques share at most one node. Roussopoulos' algorithm starts with choosing an arbitrary link in the input graph and calculating the number of triangles containing this link. Depending on this value the starting cell is determined. The starting cell is a complete graph K_m ; if $m = 2$ it is a link; if $m = 3$ a triangle that contains the starting link. Having a starting cell of the input graph, the algorithm of Roussopoulos continues to find a clique, which is deleted. In addition, in each step the vertices of the clique are labeled by a group number. One node in a line graph cannot be assigned to more than two groups (otherwise it is not a line graph). The nodes of the root graph are those partitions and all nodes are assigned to exactly one partition. In the constructed graph there is a link between two nodes, if the nodes are assigned to partitions that have a non-empty intersection. The approach of Roussopoulos is based on finding the largest cliques and sequentially the number of triangles that contain this link. Theoretically finding the largest connected component is, however, an *NP*-complete problem [52]. Lehot's algorithm employs the principles of Van Rooij and Wilf [47]: A graph is a line graph if and only if it does not have the complete bipartite graph $K_{1,3}$ as an induced sub-graph, and if two odd triangles³ have a common link, the sub-graph induced by their nodes is the complete graph K_4 . Lehot's algorithm first constructs a root graph G from the given graph H , and then compares $l(G)$ and H to determine whether the given H is a line graph, unlike Roussopoulos' algorithm, which determines whether the given graph H is a line graph during the construction of the root graph G . Naor and Novick [53] proposed a parallel algorithm for line-to-root graph construction based on a divide-and-conquer scheme. Motivated by eigenvectors, Simić [54] proposed an algorithm for recognizing generalized line graphs. Simić's algorithm searches for the maximum degree node in each loop. Degiorgi and Simon [55] proposed a constructive algorithm that bases on the Ore's proof [56] of Whitney's theorem [35], which states that two connected edge isomorphic graphs with more than four nodes are also node isomorphic and there exists exactly one node isomorphism which generates the given edge isomorphism. The root graph construction examines 2-coloring classes in the input graph components. They showed that their algorithm is more time efficient than algorithms of Roussopoulos and Lehot for sparse line graphs and non-line graphs.

We propose two algorithms, MARINLINGA and ILIGRA. The first algorithm, MARINLINGA, is based on relabeling the adjacency matrix of the line graphs. MARINLINGA does not explicitly rely on Whitney's theorem, as all previous companion algorithms, but uses link relabeling and endnode recognition. The second algorithm, ILIGRA checks

³If every node is adjacent to two or zero nodes of a triangle then it is an even triangle.

the connectivity locally. The root graph G is constructed based on the correspondence between a node in the line graph H and a link in its root graph G . Due to the choice of an arbitrary node in the line graph H and checking the connectivity of its neighbors during the algorithm's execution. Besides being conceptually simple, ILIGRA has been shown through simulations to be the most efficient inverse line graph algorithm so far proposed.

1.4 Random line graphs

A simple graph ⁴ with N nodes and L links is denoted by $G(N, L)$. The line graph H of a simple graph G is a graph in which every node corresponds to a link in G and two nodes in H are adjacent if and only if their corresponding links in G share a node. The graph G is called the root graph or the root graph of H . The number N_H of nodes in H equals the number L of links in G . Whitney's Theorem [35][36] states that, if connected graphs G_1 and G_2 have isomorphic line graphs, G_1 and G_2 themselves must be isomorphic unless one is K_3 and the other is $K_{1,3}$. Cvetković *et al.* [37] surveyed the literature on line graphs.

Line graphs can model many real-world networks. For instance, a network of tennis players is formed when we connect two players who have played in the same game and a network of tennis games is a graph where two games are linked if the same competitors have played in both of them. The network of tennis games is the line graph of the network of tennis players [40]. In metabolisms, the chemical reaction network in which the nodes are the reactions and two nodes are linked if they have the same chemical compound, is the line graph of the chemical compound network in which the nodes are the compounds and two nodes are linked if they are involved in the same chemical reaction [38][39]. Line graphs can also model social networks as they are highly clustered and assortative [40][41][38][42]. Moreover, line graphs have been used in detecting and modeling the overlapping community structure in social networks [43][44].

Despite of the significance of line graphs in the field of graph theory and complex networks, a model to generate random line graphs is still lacking. In Chapter 6, we propose a model to randomly generate line graphs with prescribed number of nodes and number of links. Before introducing the model, we discuss some preliminaries and various properties of random line graphs. Especially, we show that, given the fixed number of nodes, the number L of links in line graphs possesses forbidden gaps in the set \mathbb{N} of integers. Without generating the root graphs first, our model is capable of generating line graphs with specific link density and assortativity. Our model also enables us to generate a group of root graphs whose assortativity coefficient strictly follows a linear law. Our model constructs line graphs by merging step by step a pair

⁴A simple graph is an unweighted, undirected graph containing no self-loops nor multiple links between the same pair of nodes.

of nodes in a group of separate cliques. The nodal merging at each step must be implemented following certain rules which ensure that the constructed graphs are line graphs. Two nodes, which are merged at each step, are randomly chosen. Given the cliques of the same size, the assortativity [17][18] of the line graphs in each step is close to 0, and the assortativity of the corresponding root graphs has a linear relationship with the steps of the merging process. If a relatively smaller number of cliques of different size are added to the majority cliques of the same size, the characteristics of the assortativity of the line graphs become largely different. The line graphs are also constructed with the cliques whose sizes follow a binomial distribution. The corresponding root graphs appear equivalent to Erdős-Rényi random graphs with binomial degree distributions, zero assortativity and semicircle eigenvalue distributions.

1.5 Social networks with community structure

Social networks are currently widely studied [14][57][19]. Social networks are often defined as networks where nodes are individuals and links are relations between individuals, reflecting acquaintances, friendships, sexual relations, collaboration, common affiliation, etc. Most social networks possess common properties of the real-world networks, such as high clustering coefficient, short characteristic path length and power law degree distribution [16][15]. Particularly, they possess some special properties like assortative mixture, community and hierarchical structure [19][43][18][58]. The communities are the subnets, which exhibit relatively higher levels of internal connections. Community structures feature important topological properties that have catalyzed researches on community detection algorithms and on modularity analysis [59][20][58]. The communities overlap with each other when nodes belong to multiple communities. The overlap of different communities exists widely in real-world complex networks, particularly in social and biological networks [60][44][61]. Human beings have multiple roles in the society, and these roles make people members of multiple communities at the same time, such as companies, universities, families or relationships, hobby clubs, etc.

In the movie actor network, where nodes are the actors and two actors are connected if they have been casted together in one or more movie, we could regard the set of actors in one movie as a community. According to the definition of movie actor network, the communities of all the movies are cliques. These communities overlap with each other if they have actors in common. The similar networks are the science coauthorship networks (nodes represent the scientists and two nodes are connected if they have coauthored one or more articles and the articles are communities), the journal editor networks (nodes as the editors and two editors are adjacent if they serve on the same editorial boards of journals) and sports player networks (nodes as players and two players who played in the same games are connected).

These types of social networks are known as affiliation networks. The affiliation networks, an important and large type of social networks, are the focus of this article. The communities in affiliation networks are called groups. In the rest of this chapter, the terms “community” and “group” will be interchangeably used. Affiliation networks naturally contain many fully connected subnetworks which are called cliques or complete subgraphs in the language of graph theory, since the nodes of the same group, such as a movie cast, are all connected with each other. The clique structure of social networks increases largely the percentage of triangles among the three hops walks, consequently resulting in high clustering coefficient. Besides the statistics of individuals such as clustering coefficient, characteristic path length and nodal degree, we are also interested to answer the following questions: the number of groups, the number of individuals each group has, the groups each individual belongs to, the number of individuals every pair of groups have in common, the number of groups every pair of individuals join together, and the number of groups each group is adjacent to (two groups are adjacent if they have individuals in common).

Palla et al. [60] defined four metrics to describe how the communities of networks overlap with each other: the membership number of an individual, the overlapping depth of two communities, the community degree and the community size. Palla et al. [60] showed that the communities of real-world networks overlap with each other significantly. They reported that the membership number of an individual and the overlapping depth of two communities and the community size all follow a power law distribution, except that the community degree features a peculiar distribution that consists of two distinct parts: an exponential distribution in the beginning and a power law tail. Poller et al. [62] proposed a toy model of which both the community size and the community degree follow a power law distribution, by applying preferential attachment to community growth. There have been many efforts devoted to the modeling of social networks [11][63][64]. The growing networking model proposed by Toivonen et al. [64] succeeds in reproducing the common characteristics of social networks: community structure, high clustering coefficient and positive assortativity. The degree distribution of this model is somewhat deviating from a power law distribution despite being heavy-tailed.

In Chapter 7, we propose a complete set of metrics which can fully characterize the overlapping community structure of networks. The social networks are represented by hypergraphs. The hypergraph representation of networks facilitates the computations of the characterizing metrics. We establish a hypergraph-based social network model which exhibits innate tunable overlapping community structure. By comparing simulation results of our model with results of real-world networks, we show that our hypergraph model exhibits the common properties of large social networks: the community (group) size, the community (group) degree and the community (group) overlapping depth all follow a power law distribution, and our model possesses high clustering coefficient, positive assortativity, short average path length. By tuning the

input individual membership number to follow a power law distribution, the individual degree and the interest-sharing number also follow a power law distribution.

1.6 Randomness of brain networks

The interconnection pattern of a network can be represented [23] by a graph $G(N, L)$ with N nodes and L links. Various model networks have been introduced to study the behavior of complex systems having underlying network structures. These model networks are based on simple principles, still they capture essential features of the underlying systems. In the random graph model of Erdős and Rényi (ER) any two nodes are randomly connected with probability [5][6][7][65]. This model assumes that interactions between nodes are random. Recently, with the availability of large maps of real world networks, it has been observed that the random graph model is not appropriate for studying the behavior of real world networks. Hence, many new models have been introduced. Chung and Lu [66][67] proposed a random graph with given degree sequences, which has been later named configuration model. Configuration model has deterministic degree sequence, hence, is less random. Watts and Strogatz [15] proposed a model, popularly known as the “small-world network,” which has properties of small diameter and high clustering. Moreover, this model network is very sparse: a network with a very few number of edges, another property shown by many real-world networks. In addition to the above mentioned properties, Barabási and Albert [16] show that degree distributions of many real-world networks have a power law. This implies that some nodes are much more connected than the others. Barabási-Albert’s scale-free model and Watts-Strogatz’s small-world model have contributed immensely in understanding evolution and behavior of the real systems having network structures. These two models led to an outbreak in the field of networks, in which of randomness and regularity coexist [14][68].

Real-world networks have several universal features, like small diameter, large clustering coefficient, scale-free degree distribution, assortative or disassortative mixing of the nodes, module structures [14], etc. Irrespective of real-world networks having one or more of the above-mentioned features, one thing common to all of them, is the existence of some amount of randomness or disorder in the link structure. According to many recent studies [69][70], randomness in links is one of the most important and desirable ingredient for the proper functionality or the efficient performance of complex networks. For instance, information processing in brain is considered to be highly influenced by random links among different modular structures [71]. The question arises whether one can identify or characterize the level of randomness in the complex networks. Some authors have proposed methods to assess randomness of complex networks qualitatively [72][73].

De Haan et al. [74] has shown that the large-scale functional brain network organi-

zation in Alzheimer patients deviates from the optimal "small-world" network structure towards a more random type. This studies in Chapter 8 aims to present network spectra based methods which measure randomness of networks in a quantitative way. With the proposed randomness measure we differentiate the brains of Alzheimer patients and brains of non-Alzheimer patients.

Part I

**Reconstructability and Energy of
Networks**

Chapter 2

Spectral Perturbation and Reconstructability

In recent years, many network perturbation techniques, such as topological perturbations and service perturbations, were employed to study and improve the robustness of complex networks. However, there is no general way to evaluate the network robustness. In this chapter, we propose a new global measure for a network, the reconstructability coefficient θ , defined as the maximum number of eigenvalues that can be removed, subject to the condition that the adjacency matrix can be reconstructed exactly. Our main finding is that a linear scaling law, $E[\theta] = aN$, seems universal, in that it holds for all networks that we have studied.

2.1 Spectral Perturbation

2.1.1 Description and definition of reconstructability

The topology of a network G consisting of N nodes and L links can be described by the adjacency matrix A , a $N \times N$ zero-one matrix, where the element $a_{ij} = 1$ if there is a link between node i and node j , else $a_{ij} = 0$. Assuming that the graph is undirected, the adjacency matrix A is symmetric. All eigenvalues are real and A possesses an eigenvalue decomposition [23, art. 9, p. 443]

$$A = X\Lambda X^T$$

where $X = [x_1 \ x_2 \ \cdots \ x_N]$ is an orthogonal matrix (such that $X^T X = X X^T = I$) with as columns the real and normalized eigenvectors x_1, x_2, \dots, x_N of A , corresponding to the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N-1} \geq \lambda_N$ in descending order and the diagonal matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{N-1}, \lambda_N)$. We are interested to know how many

eigenvalues of A are needed to be able to reconstruct A exactly, given the set of eigenvectors x_1, x_2, \dots, x_N . In other words, we perturb the spectrum, the set of eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_{N-1}, \lambda_N$, of the adjacency matrix A by omitting the j smallest eigenvalues in absolute value. Since $\sum_{j=0}^N \lambda_j = 0$, on average half of the eigenvalues of the adjacency matrix A are negative. Therefore, we reorder the eigenvalues as $|\lambda_{(1)}| \leq |\lambda_{(2)}| \leq \dots \leq |\lambda_{(n)}|$ such that $\lambda_{(j)}$ is the j -th smallest (in absolute value) eigenvalue corresponding to the eigenvector $x_{(j)}$. Let us define the $N \times N$ matrices

$$\Lambda_{(j)} = \text{diag}(0, \dots, 0, \lambda_{(j+1)}, \lambda_{(j+2)}, \dots, \lambda_{(N)})$$

and

$$A_{(j)} = \tilde{X} \Lambda_{(j)} \tilde{X}^T$$

where $\tilde{X} = [x_{(1)} \ x_{(2)} \ \dots \ x_{(N)}]$ is the reordered version of X corresponding to the eigenvalues ranked in absolute value. Thus, $\Lambda_{(j)}$ is the diagonal matrix where the j smallest (in absolute value) eigenvalues are put equal to zero, or equivalently, they are removed from the spectrum of A . The spectral perturbation here considered consists of consecutively removing more eigenvalues from the spectrum until we can no longer reconstruct the adjacency matrix A . Clearly, when $j = 0$, we have that $A_{(0)} = A$ and that, for any other $j > 0$, $A_{(j)} \neq A$. Moreover, when $j > 0$, $A_{(j)}$ is not a zero-one matrix. In Figure 2.1, we show the histograms of the entries of $A_{(5)}$, $A_{(10)}$, $A_{(15)}$ and $A_{(20)}$ for an Erdős-Rényi random graph with $N = 36$ nodes and link density of $p = 0.5$. The removal of a part of the eigenvalues impacts the distribution of entries around 1 and the distribution of entries around 0 similarly, as shown in Figure. 2.1. This means that the deviation of entries around 1 and the deviation of entries around 0 are almost the same, and that the distribution of values around 1 and 0 will reach 1/2 roughly simultaneously, when the number of removed eigenvalues increases gradually.

Using Heavyside's step function $h(x)$,

$$h(x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

we truncate the elements of $A_{(j)}$ as $h\left((A_{(j)})_{ij} - \frac{1}{2}\right)$. If we now define the operator \mathcal{H} applied to a matrix $A_{(j)}$ that replaces each element of $A_{(j)}$ by $h\left((A_{(j)})_{ij} - \frac{1}{2}\right)$, then

$$\tilde{A}_j = \mathcal{H}(A_{(j)})$$

is a zero-one matrix, with the possible exception of elements $\frac{1}{2}$. The interesting observation from extensive simulation is that there seems to exist a maximal number θ , such that

$$\begin{aligned} \tilde{A}_j &= A, \text{ if } j \leq \theta \\ \tilde{A}_j &\neq A, \text{ if } j > \theta \end{aligned}$$

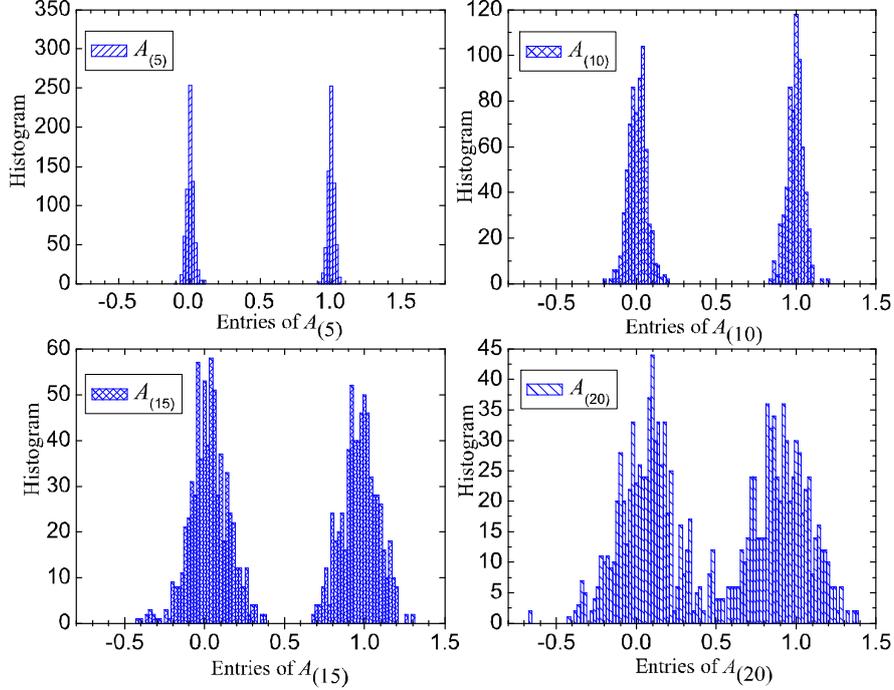


Figure 2.1: The histograms of the entries of $A_{(5)}$, $A_{(10)}$, $A_{(15)}$ and $A_{(20)}$. The matrix A ($A = A_{(0)}$) is the adjacency matrix of an Erdős-Rényi random graph with 36 nodes and link density of 0.5.

In other words, θ is the maximum number of eigenvalues that can be removed from the spectrum of the graph such that the graph can still be reconstructed precisely, given the matrix X . We therefore call θ the *Reconstructability Coefficient*.

2.1.2 Theory

The eigenvalue decomposition $A = X\Lambda X^T$ of a symmetric matrix can be rewritten in vector notation as,

$$A = \sum_{k=1}^N \lambda_k x_k x_k^T = \sum_{k=1}^N \lambda_k E_k \quad (2.1)$$

where the matrix $E_k = x_k x_k^T$ is the outer product of x_k by itself. Any element of A can be written, with the above relabelling of the eigenvectors according to a ranking in

absolute values of the eigenvalues $|\lambda_{(1)}| \leq |\lambda_{(2)}| \leq \dots \leq |\lambda_{(N)}|$ as

$$a_{ij} = \sum_{k=1}^m \lambda_{(k)} (E_{(k)})_{ij} + \sum_{k=m+1}^N \lambda_{(k)} (E_{(k)})_{ij} \quad (2.2)$$

where $m \in [1, N]$ is, for the time being, an integer. As shown in Appendix A, the 2-norm of E_k is not larger than 1, so that $|(E_{(k)})_{ij}| \leq 1$ for any $1 \leq k \leq N$, which implies that $-1 \leq (E_{(k)})_{ij} \leq 1$. Relation (2.1) also explains why an ordering in absolute value is most appropriate for our spectral perturbation: the usual ordering $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{N-1} \geq \lambda_N$ in algebraic graph theory would first remove $\lambda_N < 0$, then λ_{N-1} and so on. However, $|\lambda_N|$ can be large and its omission from the spectrum is likely to cause too big an impact.

The reconstructability of a graph as described in Section 2.1.1 is now reformulated as follows. Since a_{ij} is either zero or 1, it follows from (2.2) that, if

$$\left| a_{ij} - \sum_{k=m+1}^N \lambda_{(k)} (E_{(k)})_{ij} \right| < \frac{1}{2} \quad (2.3)$$

we can reconstruct the element a_{ij} as

$$a_{ij} = \begin{cases} 1 & \text{if } \sum_{k=m+1}^N \lambda_{(k)} (E_{(k)})_{ij} > \frac{1}{2} \\ 0 & \text{if } \sum_{k=m+1}^N \lambda_{(k)} (E_{(k)})_{ij} < \frac{1}{2} \end{cases}$$

The reconstructability requirement (2.3) determines the values of m that satisfy the inequality. The largest value of m obeying (2.3) is denoted by θ , called the reconstructability coefficient of a graph.

Using (2.2), the reconstructability requirement (2.3) is equivalent to

$$\left| \sum_{k=1}^{\theta} \lambda_{(k)} (E_{(k)})_{ij} \right| < \frac{1}{2}$$

A further analysis is difficult due to the appearance of the matrix elements $(E_{(k)})_{ij}$, of which, in general, is not much known. Since $|(E_{(k)})_{ij}| \leq 1$, we can bound the sum Ξ

$$\Xi = \left| \sum_{k=1}^{\theta} \lambda_{(k)} (E_{(k)})_{ij} \right| \leq \sum_{k=1}^{\theta} |\lambda_{(k)}| |(E_{(k)})_{ij}| \leq \sum_{k=1}^{\theta} |\lambda_{(k)}| \quad (2.4)$$

In many cases, this bound is conservative because, on average, half of the eigenvalues of the adjacency matrix A is negative. Moreover, the matrix element $(E_{(k)})_{ij}$ can also be negative. We show in Appendix B for the class of Erdős-Rényi random graph $G_p(N)$ that the bound (2.4) is, indeed, too conservative and that only extensive simulations seem appropriate to determine the reconstructability coefficient θ .

2.2 The Reconstructability of random graphs

Traditionally, complex networks have been modeled as Erdős-Rényi (ER) random graphs $G_p(N)$, which can be generated from a set of N nodes by randomly assigning a link with probability p to each pair of nodes. Besides their analytic tractability [65], the ER random graphs have also served as idealized structures for peer-to-peer networks, ad-hoc networks, gene networks, ecosystems and the spread of diseases or computer viruses. If a graph problem cannot be solved analytically for $G_p(N)$, experience teaches that chances are high that the problem is analytically intractable for all graphs with at least one parameter that can be changed (such as N). For this reason, ample attention is devoted to analyse the behavior of the reconstructability coefficient $\theta_p(N)$ for the random graph $G_p(N)$.

According to the Wigner's Semicircle Law [75][76][77] and the fact that the complement of $(G_p(N))^c = G_{1-p}(N)$, for sufficiently large N , the spectrum of $G_p(N)$ is symmetric around $p = \frac{1}{2}$. Therefore, we expect that $\theta_p(N) = \theta_{1-p}(N)$ for sufficiently large N , which is confirmed by simulations below.

2.2.1 Weibullian probability distribution of $\theta_p(N)$

We explore three classes of ER random graphs: $G_{0.2}(200)$, $G_{0.5}(200)$ and $G_{0.8}(200)$. For each class, 5000 random graphs are generated and we compute the reconstructability coefficient $\theta_p(N)$ of each graph. Figure 2.2 illustrates the probability distribution of the reconstructability coefficient $\theta_p(N)$ of ER random graphs, fitted by a Weibull distribution. The probability density function of Weibull distribution is,

$$f_{\text{Weibull}}(x; \phi, \varphi) = \begin{cases} \frac{\varphi}{\phi} \left(\frac{x}{\phi}\right)^{\varphi-1} e^{-\left(\frac{x}{\phi}\right)^\varphi} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

where ϕ is the scale parameter and φ is the shape parameter [78].

2.2.2 $E[\theta_p(N)]$ as a function of N and p

Here, we investigate the average reconstructability coefficient $E[\theta_p(N)]$ as a function of N and p , for ER random graphs. First, we consider the ER random graphs $G_{0.5}(N)$, where the graph size N is increased from $N = 50$ to $N = 2500$ with a step of 50. Within each class $G_{0.5}(N)$, the realizations of θ for 10000 graphs are computed. The mean $E[\theta]$ as well as the standard deviation of θ are depicted in Figure 2.3. $E[\theta]$ appears to be linear with N , as fitted in Figure 2.3. The linear fitting function of $E[\theta]$ as a function of N , for $G_{0.5}(N)$, is

$$E[\theta_{0.5}(N)] = 0.36N$$

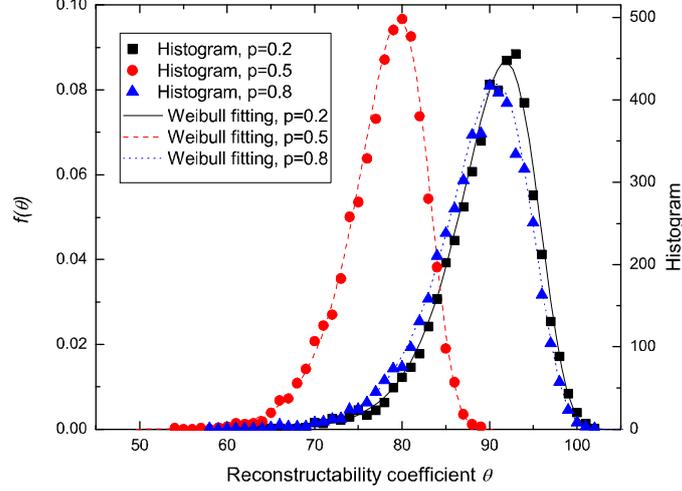


Figure 2.2: The probability density function $f(\theta)$ for $N = 200$, and $p = 0.2, 0.5, 0.8$. For each p , 5000 samples of θ are computed, and the histograms are fitted by Weibull distribution.

Table 2.1: The parameter a of the general scaling law for ER random graphs.

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
a	0.50	0.46	0.42	0.40	0.39	0.40	0.41	0.45	0.47

Furthermore, Figure 2.4 (a) shows the scaling law of $E[\theta_p(N)]$ as a function of N for random graphs with link density p from 0.1 to 0.9. These extensive simulations suggest the linear scaling law (1.1). The magnitude of a for the ER random graphs is surprisingly large. In $G_{0.5}(N)$, about 40% of the spectrum is redundant from the point of view of reconstructability defined above. For $G_p(N)$ with $p = 0.1, 0.2, \dots, 0.9$, the parameter a of the general linear scaling law (1.1) are shown in Table 2.1. Thus, we further examine the slope $a(p)$ as a function of the link density p in Figure 2.4 (b). From Table 2.1 we observe the approximate symmetry of the slope $a(p)$. The slope $a(p)$ as a function of p is fitted with a parabola

$$a(p) = 0.39(p - 0.5)^2 \quad (2.5)$$

as shown in Figure 2.4 (b).

Let $E(\theta(p, N))$ be the function $E[\theta]$ as a function of p when there are N number of nodes. We consider the ER random graphs $G_p(200)$ with $N = 200$ nodes and p ranging

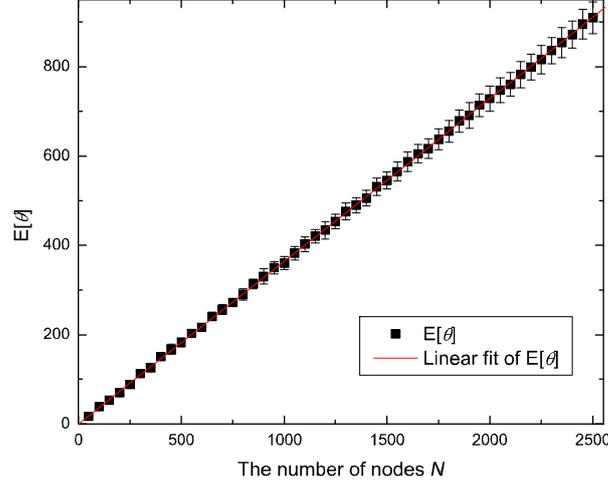


Figure 2.3: The mean of reconstructability coefficient $E[\theta]$ of ER random graphs $G_{0.5}(N)$ with standard deviation σ_θ shown as Y-error bar. $E[\theta]$, which is fitted linearly in the figure, is computed from 10000 samples for each N .

from $p = 0.01$ to 1 with a step of 0.01. Similarly, 10000 graphs are generated for each class $G_p(200)$. As depicted in Figure 2.5, the $E[\theta]$ as a function of the link density p follows a similar parabola

$$E[\theta(p, 200)] = 77(p - 0.5)^2 \quad (2.6)$$

$E[\theta]$ appears to be symmetric around $p = 0.5$ within the range $p \in [0.09, 0.91]$. It seems that, when $N \rightarrow +\infty$, $E[\theta(p, 200)] = 77(p - 0.5)^2$, holds generally for the ER random graphs. The parabola fitting only works well in the region $p \in [0.09, 0.91]$, where the graph is well connected, and so is its complementary graph, which means that the graph is not too sparse and not too dense. For very sparse/dense graphs, the analysis of the reconstructability coefficient is complicated because many degenerate eigenvalues appear.

2.3 The Reconstructability of Other Networks

In this section, we will examine whether the linear scaling law (1.1) between the average reconstructability coefficient $E[\theta]$ and the size N of a network is generally true for other types of networks.

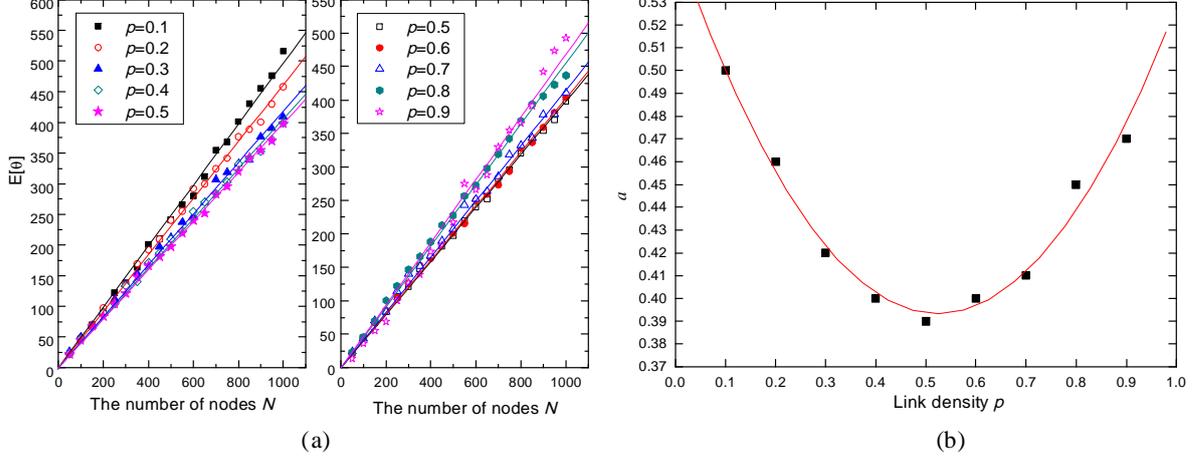


Figure 2.4: (a) The scaling law for random graphs with $p = 0.1, 0.2, \dots, 0.9$. We compute 10000 samples of θ to get the mean $E[\theta]$. (b) The curve fitting of the slope $a(p)$.

Table 2.2: The parameter a of the general scaling law for scale-free networks.

m	3	5	7
a	0.39	0.42	0.46

2.3.1 Scale-free networks

Power law graphs are random graphs specified by a power law degree distribution $\Pr[D = k] = L(k) k^{-\tau}$, where $L(k)$ is a slowly varying function of k [79]. The power law degree distribution is followed by many natural and artificial networks such as the scientific collaborations, the world-wide web and the Internet. Specifically, we investigate the Barabási-Albert power law graph [16][80][14], which starts with m_0 nodes. At every time step, we add a new node with m links that connect the new node to m different nodes already present in the graph. The probability that a new node will be connected to node i in step t is proportional to the degree $d_i(t)$ of that node.

We consider the Barabási-Albert graphs with N ranging from 500 to 2500 with a step of 100 and $m = 3, 5, 7$. Large network sizes N are selected because a power law degree distribution can be observed only when the network size is large. Within each class of the BA graphs with a specific N and m , 10000 graphs are generated.

In Figure 2.6 (a), we observe that the linear scaling law (1.1) seem to hold for

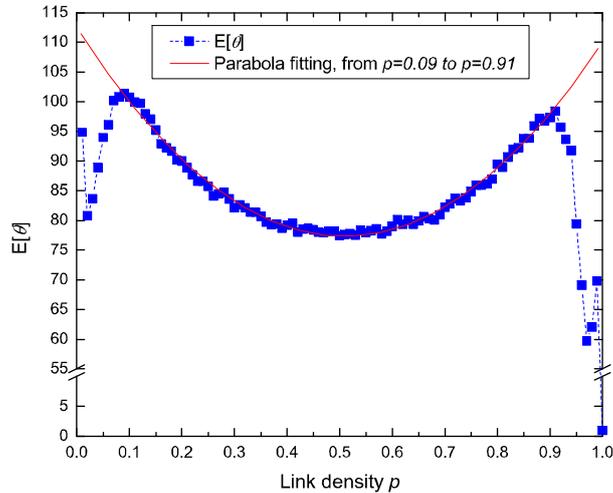


Figure 2.5: $E[\theta]$ of Erdős-Rényi random graphs $G_p(200)$ for p . $E[\theta]$ is computed by 10000 samples of θ . The function $E[\theta]$ of p , where $p \in [0.09, 0.91]$, is fitted by a parabola.

Barabási-Albert networks as well. The slope a of the corresponding linear fitting are shown in Table 2.2.

2.3.2 Small-world networks

The small-world model proposed by Watts and Strogatz [15] encompasses the following two structural features as observed in real-world networks. Any two nodes can be reached within a small number of links despite the large size of networks. Nodes are well clustered in the sense that two direct neighbors of a node are more likely to be connected compared to those in random graphs. The small-world model starts by building a ring with N nodes and by joining each node with k nearest neighbors ($k/2$ on either side of the ring). Upon the resulted ring lattice, each link connected to a clockwise neighbor is rewired to a randomly chosen node with a probability p_r , and is preserved with a probability $1 - p_r$. The small-world graph interpolates between a ring or lattice ($p_r = 0$) and a random graph with the constraint that each node has the minimal degree $k/2$ ($p_r = 1$).

The linear scaling law (1.1) of $E[\theta]$ as a function of N is also observed in Watts-Strogatz networks with different k and p_r , as shown in Figure 2.6 (b). Table 2.3 shows the parameters of the linear scaling law obtained from the curve fitting in Figure 2.6 (b). The slope a , or the proportion of eigenvalues that can be removed while the adja-

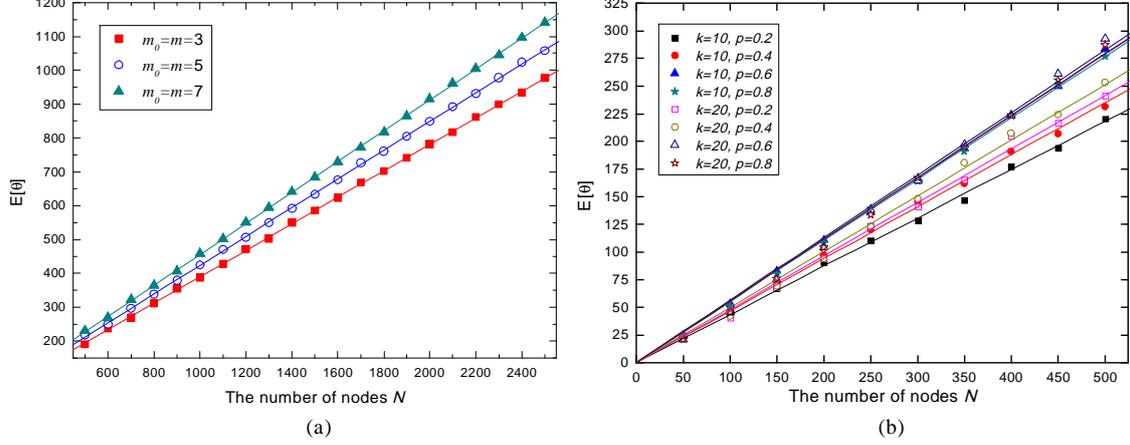


Figure 2.6: (a) The linear scaling law for $E[\theta]$ of Barabási-Albert networks. $E[\theta]$ is obtained by 10000 samples. m_0 is the number of nodes of the initial graph, and for each time step, the coming node can add m links to the current graph. (b) The linear scaling law for $E[\theta]$ of Watts-Strogatz networks. $E[\theta]$ is obtained by 10000 samples. For the initial ring lattice graph, every node has links with its k nearest nodes, and p_r , defined as the randomness, is the probability with which each link is rewired.

Table 2.3: The parameter a of the general scaling law for small-world networks.

k	10	10	10	10	20	20	20	20
p	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
a	0.44	0.47	0.56	0.55	0.48	0.50	0.57	0.56

gency matrix is still reconstructable, depends on the average degree k and the rewiring probability p_r .

2.3.3 Deterministic graphs

Finally, we explore the average reconstructability coefficient $E[\theta]$ as a function of N in a set of deterministic graphs: a) a path; b) a ring where each node on a circle is connected to its previous and subsequent neighbor on the ring; c) a wheel where a node locates in the wheel center while the other nodes are on a circle around the wheel center and the wheel center is connected to every node on the outer circle while each node on the outer circle connects to its previous and subsequent neighbor. d) D-lattice or D-dimensional

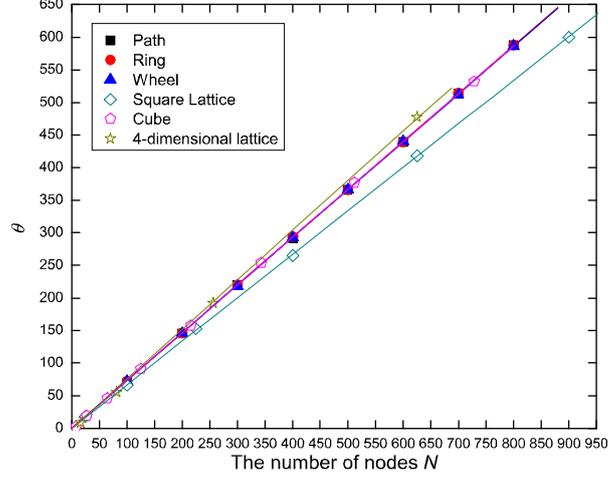


Figure 2.7: The linear scaling law for $E[\theta]$ of six special deterministic types of graphs. For path graph, ring graph and wheel graphs, $N = 100, 200, 300, \dots, 800$. We take $N = 25, 100, 225, 400, 625, 900$, $N = 8, 27, 64, 125, 216, 343, 512, 729$, $N = 16, 81, 256, 625$ for grid graphs, cubic graphs, and 4-dimensional lattice graphs, respectively.

Table 2.4: The parameter a of the general scaling law for deterministic graphs.

Graph Type	Path	Cycle	Wheel	Grid	Cube	4-dimensional lattice
a	0.73	0.73	0.73	0.67	0.73	0.76

lattices where all interior nodes have the same degree $2D$ and D is the dimension. For the first three types of graphs, we increase the number of nodes N from 100 to 800 with a step of 100. Here, we confine ourselves to the hyper-cube D -lattices in which each edge is of equal size. In this case, a 2D-lattice becomes a grid and a 3D-lattice equals a cubic lattice. For grid graphs, we take $N = (5k)^2$, $k = 1, 2, \dots, 6$, for cubic graphs, $N = k^3$, $k = 2, 3, \dots, 9$. The 4D-lattices with $N = k^4$ where $k = 2, 3, 4, 5$ are considered. Figure 2.7 shows the reconstructability coefficient θ of each graph, which seems always a linear function of the network size N . The corresponding linear curve fittings are described in Table 2.4. The paths, ring graphs and wheel graphs follows a similar linear relation between θ and N , because of similarity among their topologies. As the dimension D of a lattice increases, the slope a is larger. In other words, more eigenvalues can be removed without influencing the reconstruction of the adjacency matrix.

The complete bipartite graph $K_{m,n}$ consists of two sets \mathcal{M} and \mathcal{N} with $m = |\mathcal{M}|$ and $n = |\mathcal{N}|$ nodes respectively, where each node of one set is connected to all other nodes of the other set. There are no connection between nodes of a same set. The eigenvalues of the adjacency matrix $A_{K_{nm}}$ are $\lambda_1 = -\lambda_N = \sqrt{mn} = \sqrt{m(N-m)}$ and all others are zero. Hence, only two eigenvalues contribute in (2.1) such that $\theta \geq N - 2$. For large N , the general observed law (1.1) then shows that $a_{K_{nm}} = \frac{E[\theta_{K_{nm}}]}{N} \rightarrow 1$. In other words, the maximum possible limit value of a can be attained in very large complete bipartite graphs. However, this example shows that some graphs may have intrinsic zero eigenvalues, that should be distinguished from zero eigenvalues introduced by perturbations. Fortunately, for large complex networks, the probability that intrinsic zero eigenvalues occur tends to zero, such that the reconstructability coefficient θ mostly measures a resilient property of a large complex networks against a spectral perturbation.

2.3.4 Summary

Surprisingly, a linear scaling law between the average reconstructability $E[\theta]$ and the network size N have been observed in ER random graphs, power law graphs, small-world graphs and various deterministic graphs. This suggests that the linear relation $E[\theta] = aN$ may be a generic feature possessed by various complex networks.

2.4 The Linear Scaling Law With Eigenvector Perturbation

Besides the perturbation on eigenvalues, the eigenvector matrix can be perturbed. Recall that $X = [x_1 \ x_2 \ \cdots \ x_N]$ denotes the eigenvector matrix of the graph with N nodes. We generate a perturbation matrix R , which is an N by N matrix with random entries, chosen independently from a normal distribution with mean of 0 and variance of 1.

The Euclidean norm of a square matrix is identical to its largest singular value. Since $\|X\|_2 = 1$ for all the graph size N , we normalize the perturbation matrix as $\bar{R} = \frac{R}{\sqrt{N}}$ such that the norm of the new perturbation matrix $\|\bar{R}\|_2 \approx 1$ for all the graph size N . The perturbed eigenvector matrix is defined as

$$X' = X + \varepsilon \bar{R} \quad (2.7)$$

where the constant ε is called the perturbation factor. Under such perturbation on the eigenvector matrix, we re-examine the reconstructability coefficient.

In this section, we investigate the linear scaling law with normalized eigenvector perturbation defined by (2.7), confined to the ER random graphs. Intuitively, with

Table 2.5: The parameter a of the general scaling law for ER random graphs under eigenvector perturbation.

ε	0	0.01	0.05	0.1	0.2	0.3	0.4	0.5
a	0.38	0.38	0.37	0.36	0.31	0.13	0	0

strong perturbation in the eigenvector matrix (large ε), the reconstructability coefficient is expected to be small. Figure 2.8 shows the linear scaling law for Erdős-Rényi random graphs. With normalized eigenvector perturbation, the linear scaling law (1.1) still holds for Erdős-Rényi random graphs, when $\varepsilon \leq 0.2$.

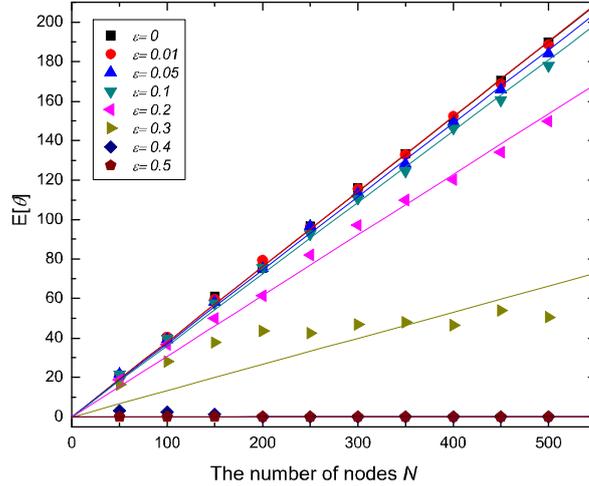


Figure 2.8: The linear scaling law with normalized eigenvector perturbation for Erdős-Rényi random graphs. $E[\theta]$ is the mean of 10000 samples of θ . The scaling factor ε are shown in the legend. The link density $p = 0.5$.

2.5 Chapter conclusion

In this chapter, we have studied the spectral reconstructability of complex networks, and defined a new metric of networks: the reconstructability coefficient θ . Through extensive simulations, we investigated the properties of the reconstructability coefficient θ for several important types of complex networks, such as ER random graphs, scale-

Table 2.6: The summary of the parameter a for different types of graphs/networks.

	a	
	min	max
ER random graphs	0.39	0.5
Barabási-Albert power law graphs	0.39	0.46
Watts-Strogatz small-world graphs	0.44	0.57
Deterministic graphs	0.67	0.76

free networks and small-world networks, and also other special deterministic types of graphs. A general linear scaling law (1.1), $E[\theta] = aN$, was found. For sufficiently large N , a portion a of the smallest eigenvalues (in absolute value) can be removed from the spectrum and the adjacency matrix is still reconstructable with its original eigenvectors. The magnitude of a for different types of complex networks with different parameters, varies from 39% to 76%, as shown in Table 2.6.

The properties of the mean of the reconstructability coefficient $E[\theta]$ was also studied under eigenvector perturbation for ER random graphs. The normalized Gaussian distributed noise matrix, scaled by the perturbation factor ε , was added to the eigenvector matrix X . Simulations show that the linear scaling law $E[\theta] = aN$ still holds for ER random graphs until the perturbation factor ε exceeds 0.2.

The basic eigenvalue relation (2.1) shows that the set of orthogonal eigenvectors are weighted by their corresponding eigenvalues. Any eigenvector specifies an orthogonal direction in the N -dimensional space. The eigenvector with an eigenvalue in absolute value close to 0 contains redundant information about the topology of the graph, in the sense that after the removal of this eigenvalue the network can still be reconstructed from the remaining spectrum. We observe that when the graphs have more constraints to be generated, the parameter a is higher. Those deterministic graphs, like path, ring and grid graphs, have more constraints to be generated, than ER random graphs, power law graphs and small-world graphs. In the spectral domain, the more generating constraints the graphs have, the less that N -dimensional space is “sampled”, or in other words, the less spectral bases (eigenvectors) we need to reconstruct the graphs. One may also say that the embedding of the graph structure in the N -dimensional space does not need those orthogonal dimensions (that act similarly as a kernel of a linear transformation).

Chapter 3

Graph Energy

This chapter reveals that graph energy has a linear relation with the multiplicity of zero adjacency eigenvalues, not only for the molecular graphs coined by Ivan Gutman, but also for graphs with random structure. We further show that the energy of a graph decreases linearly with the multiplicity of -1 adjacency eigenvalues for the graphs constructed by combining Erdős-Rényi random graphs and triangles. Last, we use the degree-preserving rewiring to increase/decrease the assortativity of an Erdős-Rényi random graph step by step to $1/-1$, and show that the energy decreases roughly linearly with the rewiring steps, and the energy is at its maximum when the graph is non-assortative, and the energy decreases when the graph becomes assortative/disassortative.

3.1 Graph energy vs. m_0

In this section, we study the relation between the energy E_G and the multiplicity m_0 of zero eigenvalues for graph $G(N, L)$. By adjusting the structure of the graph $G(N, L)$, we increase m_0 by 1 at each step. We have two different $G(N, L)$: one has deterministic structure and the other is generated randomly at each step.

Our method to increase m_0 is due to Van Mieghem [34]: Adjusting the structure of the graph to have one more pair of non-adjacent nodes which have the same neighbors. This pair of nodes correspond to two same rows/columns in adjacency matrix, of which the rank hence decreases by 1. Therefore, there becomes one more zero eigenvalue.

3.1.1 Deterministic graphs

As shown in Figure 3.1 (a), G_k consists of three connected subgraphs, $K_{1,k}$, P_{199-k} and K_{100} , $k = 1, 2, 3, \dots, 198$. The multiplicity m_0 of zero eigenvalues of adjacency matrix of G_k equals,

$$m_0(G_k) = k - 1$$

The energy E_G of $G_k, k = 1, 2, 3, \dots, 198$, has been numerically computed. Figure 3.1 (b) depicts the energy E_G as a function of m_0 , fitted by a line $E_G = 458.47 - 1.1648m_0$.

3.1.2 Random graphs

The graph G_k at step k is illustrated in Figure 3.2 (a). The graph $G_k, k = 1, 2, 3, \dots, 30$, is generated by the following procedures:

1. Generate an Erdős-Rényi random graph $G_{ER}(N, L)$ with $N = 200 - 2k$ and $L = 3000 - 2k$.
2. Randomly choose k nodes from $200 - 2k$ nodes; And for each chosen node, attach two extra nodes to it.

The multiplicity m_0 of zero eigenvalues of adjacency matrix of G_k equals,

$$m_0(G_k) = k$$

The energy E_G of $G_k, k = 1, 2, 3, \dots, 30$, has been computed. Figure 3.2 (b) depicts the energy E_G as a function of m_0 , fitted by a line $E_G = 886.22 - 5.7254m_0$.

3.2 Graph energy vs. m_{-1}

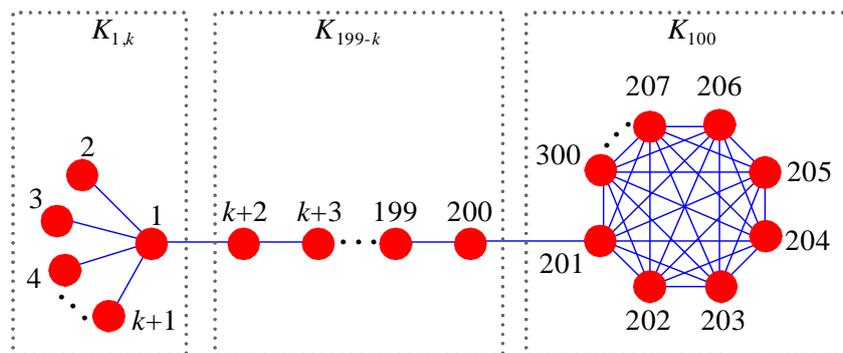
The section studies the relation between the energy E_G and the multiplicity m_{-1} of -1 eigenvalues for the graph $G(N, L)$. By proper design, at each step k , there are k pairs of adjacent nodes which have the same neighbors in the graph $G(N, L)$. These k pairs of nodes lead to $m_{-1} = k$. The graph G_k at step k is illustrated in Figure (a). The graph $G_k, k = 1, 2, 3, \dots, 30$, is generated by the following procedures:

1. Generate an Erdős-Rényi random graph $G_{ER}(N, L)$ with $N = 200 - 2k$ and $L = 3000 - 3k$.
2. Randomly choose k nodes from $200 - 2k$ nodes; And for each chosen node, attach two extra nodes to it and connect the two extra nodes with a link.

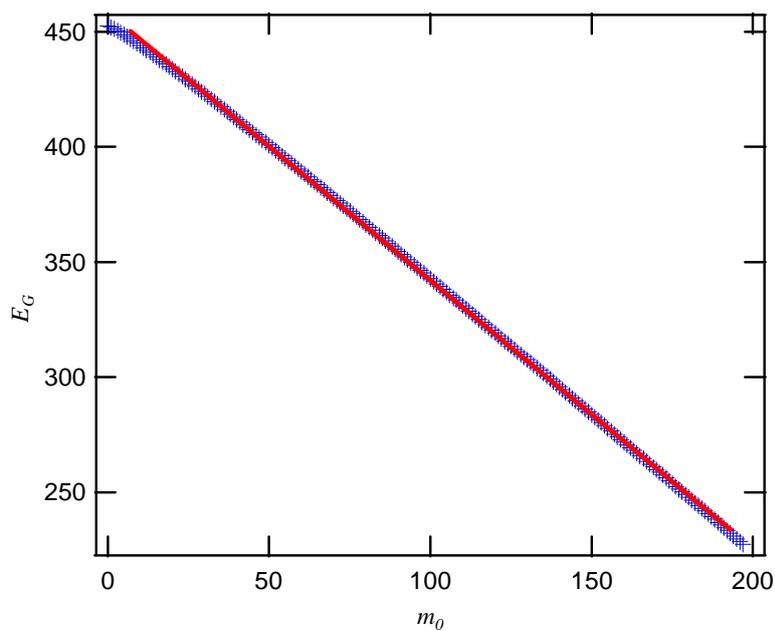
The multiplicity m_{-1} of -1 eigenvalues of adjacency matrix of G_k equals,

$$m_0(G_k) = k$$

The energy E_G of $G_k, k = 1, 2, 3, \dots, 30$, has been computed. Figure 3.3 (b) depicts the energy E_G as a function of m_{-1} , fitted by a line $E_G = 885.81 - 4.2316m_{-1}$.



(a)



(b)

Figure 3.1: (a) The group of graphs G_k , $k = 1, 2, 3, \dots, 198$, with $m_0(G_k) = k - 1$. (b) The energy E_G of the graph G_k , fitted by a linear function $E_G = 458.47 - 1.1648m_0$.

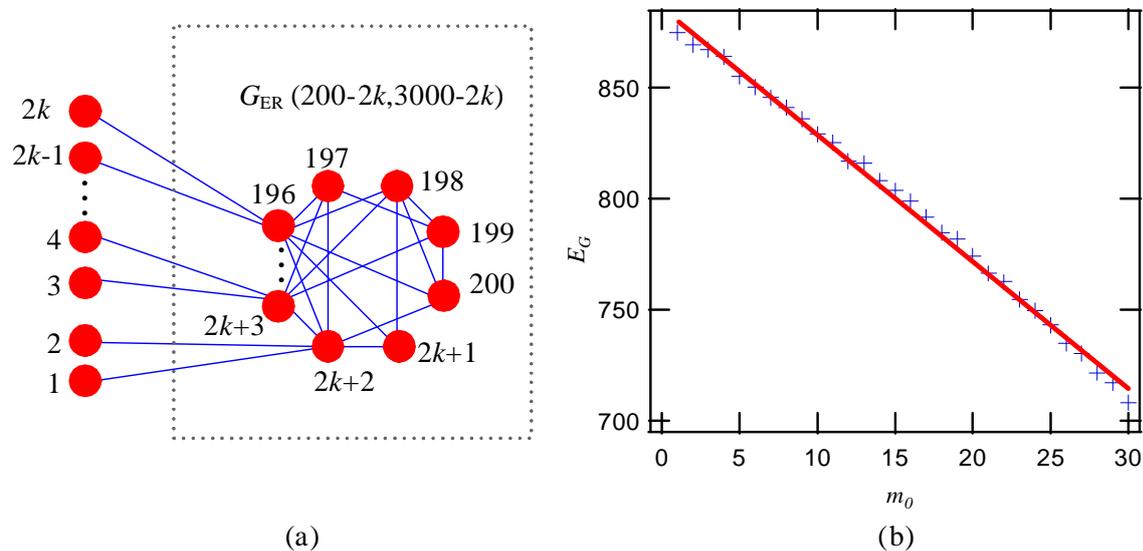


Figure 3.2: (a) The group of graphs G_k , $k = 1, 2, 3, \dots, 30$, with $m_0(G_k) = k$. (b) The energy E_G of graph G_k , fitted by a linear function $E_G = 886.22 - 5.7254m_0$.

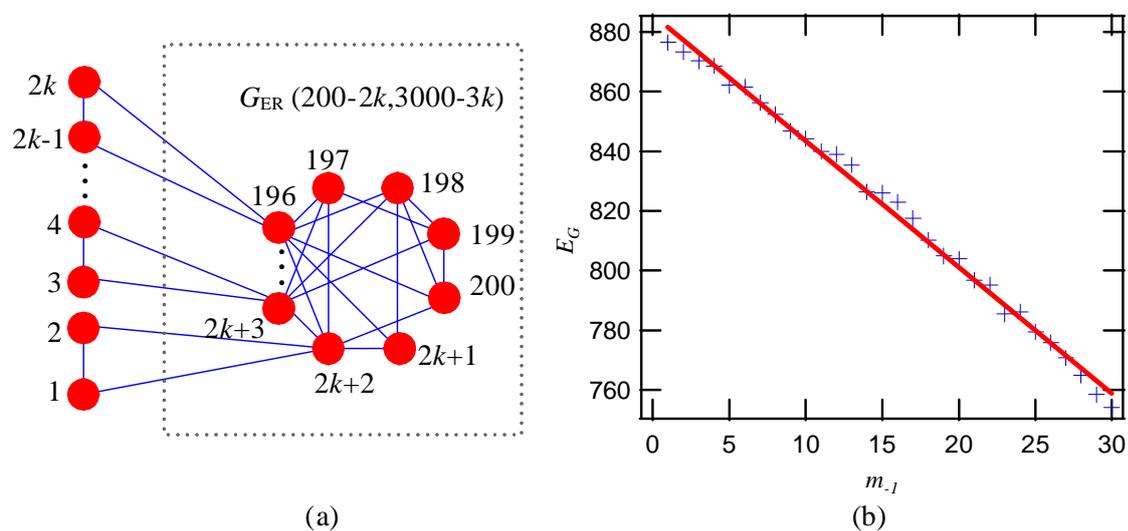


Figure 3.3: (a) The group of graphs G_k , $k = 1, 2, 3, \dots, 30$, with $m_{-1}(G_k) = k$. (b) The energy E_G of graph G_k , fitted by a linear function $E_G = 885.81 - 4.2316m_{-1}$.

3.3 Influence of assortativity on graph energy

In this section, we study the relation between the energy E_G and the assortativity ρ_D of a graph. Van Mieghem et al. [81] showed that the spectral radius λ_1 increases when the assortativity increases. Van Mieghem et al. [81] introduced a link rewiring method, the degree-preserving rewiring, which can gradually raise or decrease the assortativity ρ_D of a graph.

In degree-preserving rewiring, the degree of all the nodes remain unchanged. In each step, we randomly choose two links which do not share a node. The four nodes which are incident to the selected two links are denoted by n_1, n_2, n_3 , and n_4 , and the degrees of them satisfy $d_{n_1} \geq d_{n_2} \geq d_{n_3} \geq d_{n_4}$.

- Degree-preserving assortative rewiring: rewire the two links such that node n_1 is adjacent with node n_2 and node n_3 is adjacent with node n_4 . If any of the new links exists before rewiring, do nothing in this step. After each rewiring step, the assortativity ρ_D becomes larger or unchanged.
- Degree-preserving disassortative rewiring: rewire the two links such that node n_1 is adjacent with node n_4 and node n_2 is adjacent with node n_3 . If any of the new links exists before rewiring, do nothing in this step. After each rewiring step, the assortativity ρ_D becomes smaller or unchanged.

3.3.1 Random graphs

We generate an Erdős-Rényi random graph $G_{ER}(N, L)$ with $N = 60$ nodes and $L = 350$ links, and the initial assortativity $\rho_D = 0.010872$, which is very close to zero, and it is in accordance with theory, since the degrees of two adjacent nodes in random graphs are completely uncorrelated. We then use the degree-preserving assortative rewiring for $10L = 3500$ steps to increase the assortativity of the graph, as shown in the left diagram of Figure 3.4 (a). At each rewiring step, the energy E_G is also computed, as shown in the right diagram of Figure 3.4 (a). The energy E_G decreases roughly linearly with the rewiring steps r and is fitted by a line function $E_G = 164.33 - 0.0054928r$. With the degree-preserving disassortative rewiring, we rewire the graph $G_{ER}(N, L)$ for $10L = 3500$ steps and ρ_D decreases from 0.010872 to -0.87605 , as illustrated in the left diagram of Figure 3.4 (b). Surprisingly, as shown in the right diagram of Figure 3.4 (b), the energy E_G decreases again roughly linearly with the rewiring steps r , and is fitted by a line function $E_G = 164.43 - 0.0062606r$.

In Figure 3.4 (c), we show by numerical computations that the energy E_G is at its maximum when the graph is approximately non-assortative, and the energy E_G decreases when the graph becomes more assortative/disassortative.

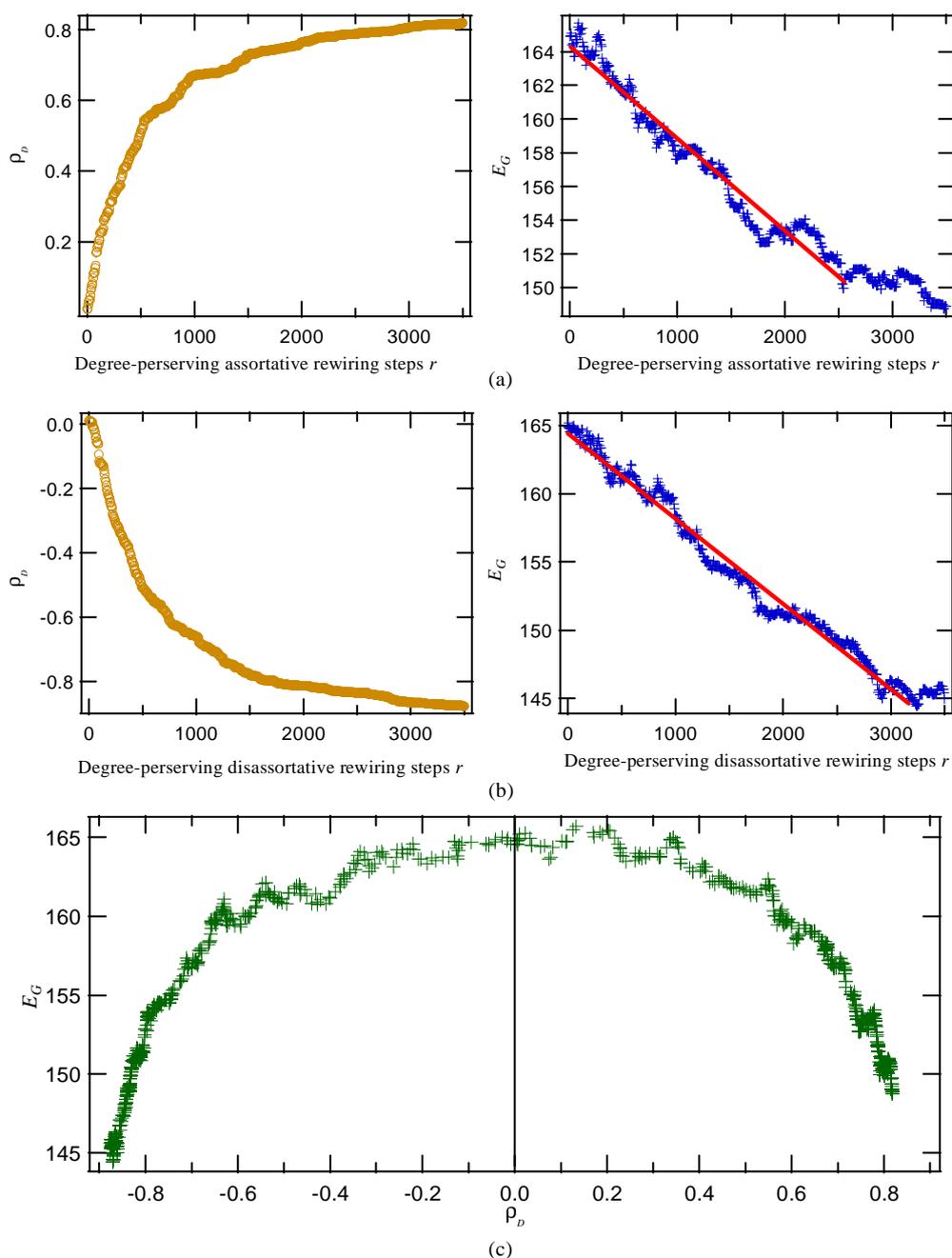


Figure 3.4: The degree-preserving rewiring is applied to an ER random graph with 60 nodes and 350 links. (a) The degree-preserving assortative rewiring increases the assortativity ρ_D of the graph (left) and decreases the graph energy E_G roughly linearly (right), fitted by a linear function $E_G = 164.33 - 0.0054928r$. (b) The degree-preserving disassortative rewiring decreases the assortativity ρ_D of the graph (left) and also decreases the graph energy E_G roughly linearly (right), fitted by a linear function $E_G = 164.43 - 0.0062606r$. (c) The energy E_G as a function of the assortativity ρ_D .

3.3.2 Grid graph

A $m \times n$ grid graph is the graph whose nodes correspond to the points in the $m \times n$ rectangle with integer coordinates, x -coordinates being in the range $1, 2, \dots, n$, y -coordinates being in the range $1, 2, \dots, m$, and two nodes are connected by a link whenever the corresponding points are at distance 1. The grid graphs are fully deterministic graphs. We generate a 10×15 grid graph with 150 nodes and 275 links. The degree-preserving rewiring is applied to the grid 10×15 graph. The initial assortativity coefficient $\rho_D = 0.5879$. The grid graphs are not regular graphs, but they are very close to regular graphs with degree 4, since most of nodes have degree 4 except those nodes locating at four edges and four corners. This explains why the initial assortativity coefficient of the grid graphs are high. When the disassortative degree-preserving rewiring is applied, the assortativity can decrease down to -0.30215 . Although there is no much space for the assortativity to grow, ρ_D still increases up to 0.85165 when we use the assortative degree-preserving rewiring. The top plot of Figure 3.5 shows that the energy E_G of the grid graph is at its minimum before rewiring. The energy E_G start to increase when either of degree-preserving rewirings is employed.

3.3.3 Random trees

A tree is an undirected graph where there is only one path between each pair of nodes. A random tree with N nodes is a tree uniformly chosen from the space of the trees with N nodes. According to the definition of trees, any random tree with N nodes has only $N - 1$ links, and thus has a very low link density. The random tree we study in this section has 200 nodes and 199 links, with initial assortativity coefficient $\rho_D = -0.0727$, close to zero. The second top plot of Figure 3.5 shows the relation of the graph energy E_G and the assortativity ρ_D for trees. We observe that the graph energy E_G and the assortativity ρ_D are highly correlated for trees. The energy E_G of grid graphs increases roughly linearly with the assortativity coefficient ρ_D .

3.3.4 Small-world graphs

The concept of the Watts-Strogatz small-world graph model has been introduced in Section 2.3.2, Chapter 2. The Watts-Strogatz small-world graph we study here has 100 nodes, and each nodes initially is connected to 8 neighboring nodes (400 links in total), and the rewiring probability is 0.1. The initial assortativity coefficient of the Watts-Strogatz small-world graph $\rho_D = 0.006$, which is almost zero, suggesting that the graph is very random and the degrees of adjacent nodes are uncorrelated. When the degree-preserving rewiring is applied to the small-world graph, the assortativity can reach from -0.77 to 0.78 . As shown in the third plot (counting from the top) of Figure 3.5, the graph energy E_G appears to increase with the absolute value of assortativity

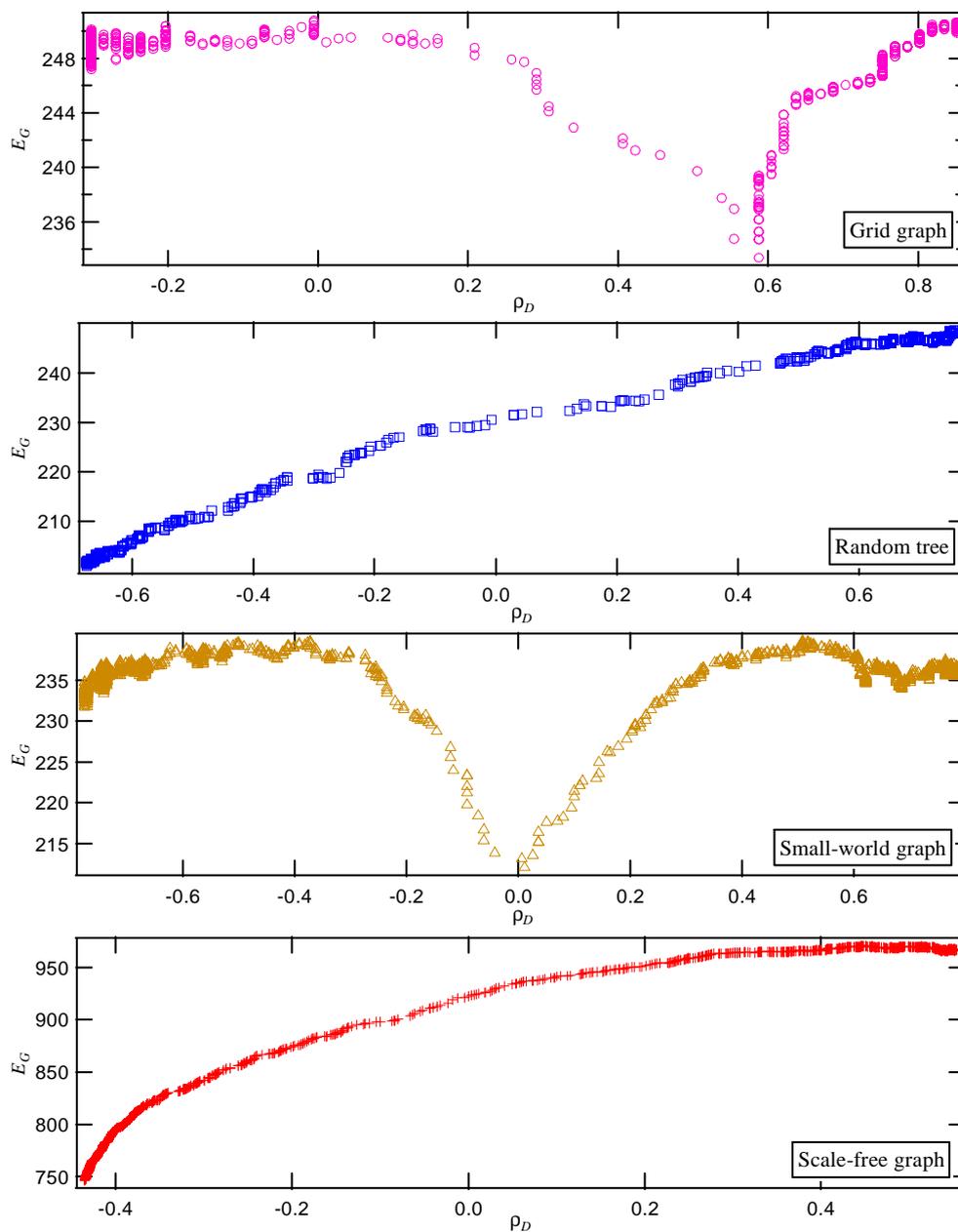


Figure 3.5: The relations of the graph energy E_G and the assortativity coefficient ρ_D for a 10×15 grid graph (pink circle), a random tree with 200 nodes (blue square), a Watts-Strogatz small-world graph with 100 nodes, 400 nodes, and rewiring probability 0.1 (brown triangle), and a Barabasi-Albert scale-free graph with 600 nodes and 1196 links (red cross).

$|\rho_D|$ in the range from -0.4 to 0.4 . In the other range, the relation of energy E_G and the assortativity coefficient ρ_D seems to be very fuzzy.

3.3.5 Scale-free graphs

The Barabási-Albert scale-free graph model has been introduced in Section 2.3.1, Chapter 2. The Barabási-Albert scale-free graph has a power law degree distribution, which is followed by many natural and artificial real-world complex networks. The Barabási-Albert scale-free graph we consider here was generated from a seed graph of two isolated nodes. One node is added at one step. Each new node has two links connecting to existing nodes in the graph. For each new link, the existing node is chosen with the probability proportional to its degree plus 1. The finally generated scale-free graph has 600 nodes and 1196 links. The scale-free graph before degree-preserving rewiring is disassortative, with $\rho_D = -0.12$, since most low-degree nodes (the new coming nodes at later steps) are preferably connected to the high-degree nodes. When the degree-preserving rewiring is applied to the scale-free graph, the assortativity can only reach from -0.44 to 0.55 , because of the heterogeneous degree distribution. As shown in the bottom plot of Figure 3.5, the graph energy E_G and the assortativity ρ_D are positively correlated.

3.4 Chapter conclusion

We have studied the relations between the energy E_G and three other metrics, the multiplicity m_0 of zero adjacency eigenvalues, the multiplicity m_{-1} of -1 adjacency eigenvalues, and the assortativity coefficient ρ_D of graph G with constant number of nodes and constant number of links. We show that the energy E_G has a linear relation with m_0 , not only for the molecular graphs coined by Ivan Gutman, but also for quite general graphs: deterministic graphs and random graphs. We also show that the energy E_G decreases linearly with m_{-1} for the graphs constructed by combining Erdős-Rényi random graphs and triangles. We use the degree-preserving rewiring to increase/decrease the assortativity of an Erdős-Rényi random graph step by step towards $1/-1$, and show that the energy decreases roughly linearly with the rewiring steps, and the energy is at its maximum when the graph is non-assortative, and the energy decreases when the graph becomes assortative/disassortative. The claims of linearity for graph energy are supported by extensive simulations. However, the analytical results or proofs are still waiting to be found.

Part II

Line Graphs and Root Graphs

Chapter 4

MARINLINGA

We propose a new algorithm MARINLINGA for reverse line graph computation, i.e., constructing the original graph from a given line graph. Based on the completely new and simpler principle of link relabeling and endnode recognition, MARINLINGA does not rely on Whitney's theorem while all previous algorithms do. MARINLINGA has a worst case complexity of $O(N^2)$, where N denotes the number of nodes of the line graph.

4.1 Link adjacency matrix (LAM) and line graph

Two nodes of a graph are said to be adjacent if there is a link directly connecting them. The adjacency matrix A of a graph contains all information of node adjacency: if node i and node j are adjacent, the entry $a_{ij} = 1$, otherwise $a_{ij} = 0$. Similarly, two links are adjacent if they are incident to the same node.

Definition 1 *The link adjacency matrix (LAM) C of a graph G with N_G nodes and L_G links is the $L_G \times L_G$ symmetric matrix with the entry $c_{ij} = 1$ if link i and link j of G are adjacent, else $c_{ij} = 0$.*

The line graph $l(G)$ of the graph G has $N_{l(G)}$ nodes and $L_{l(G)}$ links, and consequently we have $L_G = N_{l(G)}$. According to the definitions of the line graph and the LAM, evidently, the LAM C of G is equal to the adjacency matrix $A_{l(G)}$ of $l(G)$,

$$C = A_{l(G)} \tag{4.1}$$

Due to Whitney's theorem and ignoring isomorphisms, for any graph except K_3 and $K_{1,3}$, one can construct the graph exclusively from its LAM. Usually, the (node) adjacency matrix is used to represent a graph. Here we use the LAM to specify any graph, except for K_3 and $K_{1,3}$. Constructing the original graph of a line graph is equivalent to converting a graph representation from the LAM to the adjacency matrix. By constructing the original graph directly from the line graph, confusion will arise concerning

the links in the original graph and the nodes in the line graph. By introducing the concept of LAM, we can avoid confusion and facilitate the description of our algorithm MARINLINGA.

4.2 Properties of the LAM

For a simple (undirected, unweighted and without self-loops) graph $G(N_G, L_G)$ with N_G nodes and L_G links, the LAM C has more constraints than the corresponding adjacency matrix A , besides being symmetric and containing only 0 and 1 entries.

A link i has two endnodes, the endnode i^+ and the endnode i^- . Link j also has endnodes j^+ and j^- . There are four configurations where link i is adjacent to link j , as shown in Figure 4.1. For each single pair of links, the LAM only indicates whether they are adjacent. If they are adjacent, we still do not know in which of the four possible configurations this pair of links is adjacent. Fortunately, by combining the adjacency relation of 3 or more links, we can determine the configuration of those links.

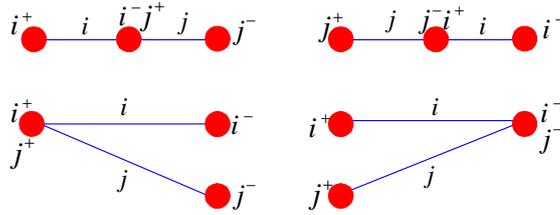


Figure 4.1: The four possible configurations in which link i is adjacent to link j .

Definition 2 If m links ($m \geq 2$) are adjacent to link i and incident to the same endnode of link i , these m links are pairwise adjacent.

Definition 3 The links, which are adjacent to link i , are defined as the neighboring links of link i .

Definition 4 The links incident to the endnode i^+ of a link i are defined as the left-neighboring links of i , and the links incident to the endnode i^- are defined as the right-neighboring links of i .

If we can recognize the link adjacency pattern of a link and its neighboring links, we can specify the graph entirely.

Figure 4.2 (a) depicts an example of a link and its neighboring links. The link i has 5 left neighboring links at its left endnode i^+ , denoted as i_{+1}, \dots, i_{+5} , and 4 right

neighboring links at its endnode i^- , denoted as i_{-1}, \dots, i_{-4} . The link adjacency pattern of these 10 links is shown in Figure 4.2 (b). In the link adjacency pattern, the labels of the left-neighboring links i_{+1}, \dots, i_{+5} are larger than link i , and smaller than the right-neighboring links i_{-1}, \dots, i_{-4} .

Given the configuration of link i and its neighboring links, the corresponding link adjacency pattern conforms to the following rules:

1. the left-neighboring links (such as i_{+1}, \dots, i_{+5} in the example of Figure 4.2 (a)) are incident to the same endnode i^+ , and are said (Definition 2) to be pairwise adjacent. Similarly, the right-neighboring links (such as i_{-1}, \dots, i_{-4} in the example of Figure 4.2 (a)) are also pairwise adjacent. This explains the two all-1-triangles (surrounded by the dashed lines) in Figure 4.2 (b), the upper one corresponding to i^+ and the second triangle corresponding to pairwise adjacent links i_{-1}, \dots, i_{-4} .
2. Since there is **at most one** link between two nodes (multi-links are forbidden), each of the left-neighboring links can be adjacent to **at most one** right neighboring link and vice versa. Hence in Figure 4.2 (b), there exists **at most one** 1-entry in each row/column of the submatrix in yellow.

We summarize this observation:

Criterion 5 *If the given link adjacency pattern has the following features, it is the link adjacency pattern of a link i and its neighboring links (the labels of the left-neighboring links are larger than link i , and smaller than the right-neighboring links),*

- *All entries of the first row are 1-entries;*
- *The triangle bounded by the $(n_{i^+} + 1)$ th column (including the $(n_{i^+} + 1)$ th column) is an all-1-triangle, where n_{i^+} denotes the number of the left-neighboring links of link i and $n_{i^+} \geq 3$;*
- *There is at most one 1-entry in each row/column of the submatrix, which is from the 2nd to the $(n_{i^+} + 1)$ th row and from the $(n_{i^+} + 2)$ th to the $(n_{i^+} + n_{i^-} + 2)$ th column, where n_{i^-} denotes the number of the right-neighboring links;*
- *The triangle bounded by the $(n_{i^+} + 2)$ th row (including the $(n_{i^+} + 2)$ th row) is an all-1-triangle.*

Theorem 6 *Consider three links i, j and k are pairwise adjacent. If each of the other m links is adjacent to all the three links i, j and k , then all the $m + 3$ links are pairwise adjacent.*

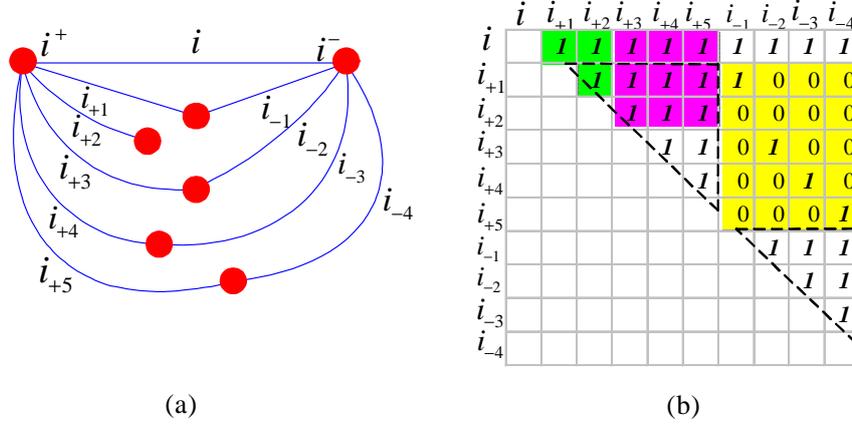


Figure 4.2: (a) The configuration of a link i and its neighboring links. (b) The corresponding link adjacency pattern. there is at most one 1-entry in each row/column of the submatrix in yellow. If all the entries in green and magenta are 1-entries, the entries of the triangle in white must be also 1-entries.

Proof. The three links i, j and k are pairwise adjacent and the configuration of i, j and k can be K_3 or $K_{1,3}$, as shown in Figure C.1 (b). If the configuration is K_3 , other links can be adjacent to at most two of i, j and k . However, if the other m links are adjacent to i, j and k , the configuration of i, j and k must be $K_{1,3}$, and i, j and k have a common endnode. Since each of the m links is adjacent to i, j and k , the common endnode of i, j and k must be also an endnode of each of the m links. According to Definition 2, all these $m + 3$ links are pairwise adjacent. ■

In Figure 4.2 (b), links i, i_{+1} and i_{+2} are pairwise adjacent, as shown by entries in green. Links i_{+3}, i_{+4} and i_{+5} are adjacent to i, i_{+1} and i_{+2} , as shown by entries in magenta. By Theorem 6, links $i, i_{+1}, i_{+2}, i_{+3}, i_{+4}$ and i_{+5} are pairwise adjacent.

4.2.1 The basic forbidden link adjacency patterns in a LAM

Figure 4.3 (a) depicts the smallest forbidden link adjacency pattern in a LAM. The configuration of links i, j and k is a path on four nodes. Since link i has neighboring links at both of its two endnodes, and if link r is adjacent with link i , then link r must be also adjacent with link j or k . Hence, the pattern in Figure 4.3 (a) will not appear in a LAM.

There are 6 forbidden link adjacency patterns of links i, j, k, r and t , as shown in Figure 4.5. Since the number of the left-neighboring links of link i is smaller than 3, we cannot use Criterion 5 to prove that the 6 link adjacency patterns are forbidden. However, Figure 4.4, which exhibits the possible configurations of the link adjacency

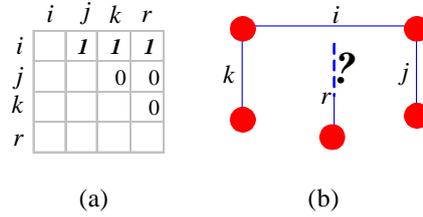


Figure 4.3: The smallest forbidden link adjacency pattern.

patterns of links i, j, k and r , will facilitate the proof that the 6 link adjacency patterns in Figure 4.5 are forbidden.

The link adjacency pattern of links i, j, k and r in Figure 4.5 (a), (b) and (c) are the same as the link adjacency pattern of links i, j, k and r in Figure 4.4 (a). There are only two possible configurations of this link adjacency pattern. As we can observe in Figure 4.4 (a), it is impossible to have a new link t which is only adjacent with link i , or only adjacent with links i and j , or adjacent with all of i, j, k and r . Hence, the patterns in Figure 4.5 (a), (b) and (c) are forbidden. In the same way, we observe that the patterns in Figure 4.5 (d), (e) and (f) are also forbidden.

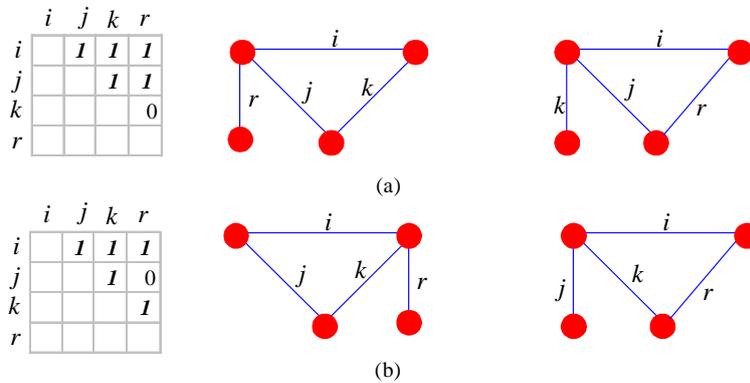


Figure 4.4: The possible configurations for two link adjacency patterns of 4 links. This figure helps to prove that the patterns of 5 links in Figure 4.5 are forbidden.

When the number of the left-neighboring links of link i is not smaller than 3 (which implies that the number of 1-entries in the first all-1-triangle is not smaller than 6), we can use Criterion 5 to determine whether a link adjacency pattern is forbidden.

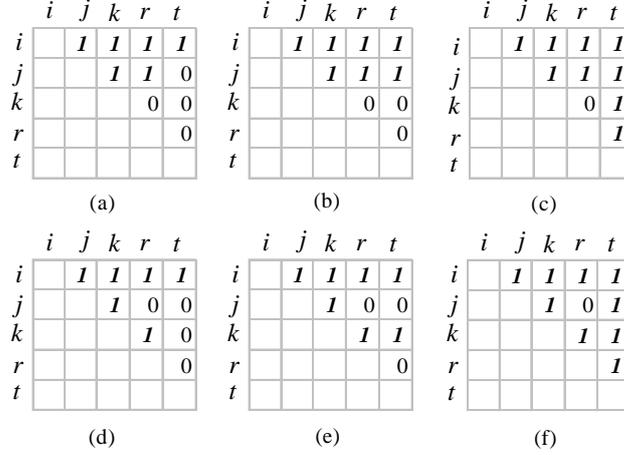


Figure 4.5: The forbidden link adjacency patterns of 5 links.

4.3 MARINLINGA

MARINLINGA is the algorithm that we designed to compute the original graph of a line graph, given the adjacency matrix of that line graph. Although MARINLINGA is designed for connected line graphs, it is also convenient to compute the original graph of a disconnected line graph component by component. In the description of MARINLINGA, the connectedness of the concerned graph is always assumed.

As explained in Section 4.1, the adjacency matrix $A_{l(G)}$ of $l(G)$ is equal to the LAM C_G of G . Constructing the original graph of a line graph, is equivalent to constructing a graph given the LAM of that graph. MARINLINGA only deals with the upper triangle of the given LAM C .

4.3.1 Matrix relabeling

The matrix relabeling algorithm rearranges the LAM C in such a way that the left and right neighboring links of the first link can be recognized via Theorem 6 and the construction algorithm can work efficiently. In each column there are some 1-entries (red dots). If after relabeling the top 1-entries of all the columns are connected by a curve, the curve should be non-increasing. For example, by the LAM C of a graph with 50 links in Figure 4.6 (a), we can only determine which links are adjacent to the first link, without any information about which endnode of the first link that the neighboring links are incident to. Fortunately, according to Theorem 6, the relabeled LAM C in Figure 4.6 (b) tells that links 2-5 are the left-neighboring links of the first link and links 6-10 are the right-neighboring links.

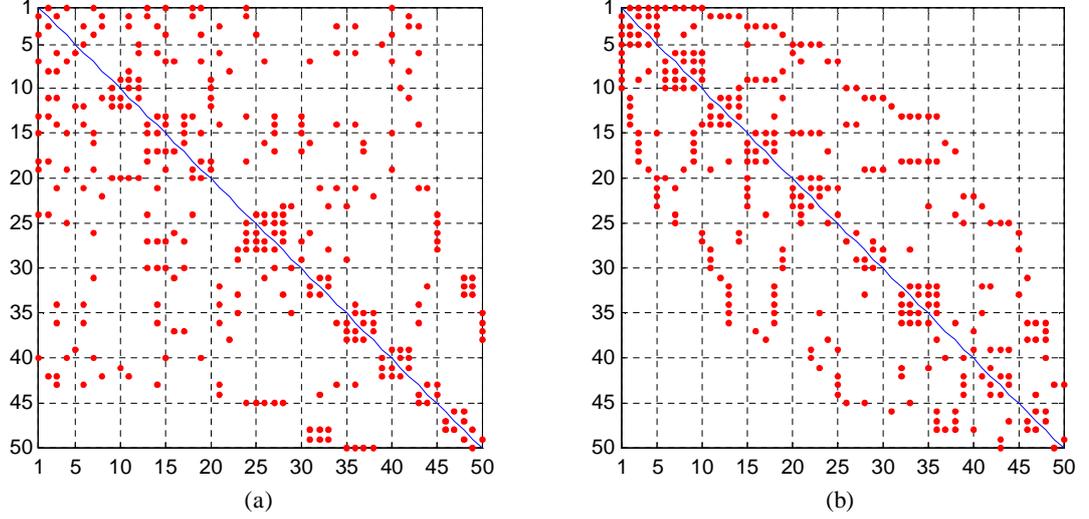


Figure 4.6: Matrix relabelling on the LAM C of a graph with 50 links. The red dots represent 1-entries. (a) Before relabelling; (b) After relabelling.

Let us first introduce the meaning of swapping the labels of two links in a LAM $C_{L_G \times L_G}$. The entry c_{ij} indicates whether links i and j are adjacent. Swapping the labels of links j and k ($j < k$) implies that links which are previously adjacent to link j are now adjacent to link k , and links which are previously adjacent to link k , are now adjacent to link j , but the adjacency relation between links j and k is the same as before, namely the entry c_{jk} of $C_{L_G \times L_G}$ is unchanged. Hence, swapping the labels of links j and k ($j < k$) means to swap the entries c_{ij} and c_{ik} for $i = 1, 2, \dots, j-1$ (shown in the example of Figure 4.8 in green), the entries c_{ji} and c_{ik} for $i = j+1, \dots, k-1$ (in magenta), the entries c_{ji} and c_{ki} , $i = k+1, \dots, L_G-1, L_G$ (in yellow).

```

C ← SWAPLABEL(C, j, k)
1 for i = 1 to j - 1 do
2   swap(cij, cik)
3 for i = j + 1 to k - 1 do
4   swap(cji, cik)
5 for i = k + 1 to LG do
6   swap(cji, cki)

```

Figure 4.7: Meta-code for SWAPLABEL.

Lines 1-3 of the metacode of Algorithm 4.7 swap the entries c_{ji} and c_{ik} , $i = j + 1, \dots, k - 1$, and lines 4-6 swap the entries c_{ji} and c_{ik} , $i = j + 1, \dots, k - 1$, and lines 7-9 swap the entries c_{ji} and c_{ki} , $i = k + 1, \dots, L_G - 1, L_G$. The code $swap(c_{ij}, c_{ik})$ of line 2 is equivalent to the codes: $t = c_{ij}$; $c_{ij} = c_{ik}$; $c_{ik} = t$.

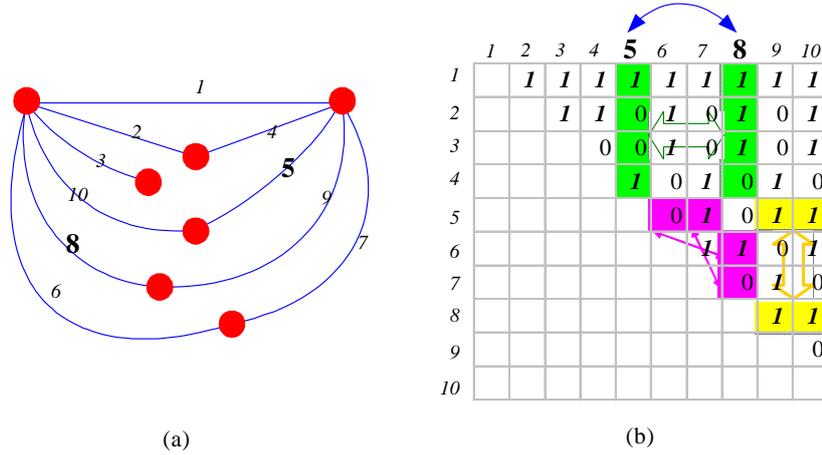


Figure 4.8: The illustration of swapping the labels of link 5 and 8. The entries in green, magenta and yellow ought to be swapped respectively.

Next, we will explain the matrix relabeling algorithm. We will first give an example showing how the matrix relabeling algorithm relabels the LAM C in Figure 4.6 (a) into the matrix in Figure 4.6 (b). In the first row of the matrix in Figure 4.6 (a) there are 9 1-entries in total. There are 6 0-entries from $c_{1,2}$ to $c_{1,10}$ and 6 1-entries from $c_{1,11}$ to $c_{1,50}$: $c_{1,3} = c_{1,5} = c_{1,6} = c_{1,8} = c_{1,9} = c_{1,10} = 0$ and $c_{1,13} = c_{1,15} = c_{1,18} = c_{1,19} = c_{1,24} = c_{1,40} = 1$. We swap the labels of links 3 and 13, links 5 and 15, links 6 and 18, links 8 and 19, links 9 and 24, links 10 and 40 by Algorithm 4.10 and the LAM C is shown in Figure 4.9. In the second row, there are 3 1-entries from $c_{2,3}$ to $c_{2,10}$. There are 2 0-entries from $c_{2,3}$ to $c_{2,5}$ and 2 1-entries from $c_{2,6}$ to $c_{2,10}$: $c_{2,4} = c_{2,5} = 0$ and $c_{2,6} = c_{2,9} = 1$. We swap the labels of links 4 and 6, links 5 and 9. By similar operations, we relabel the LAM C into the order shown in Figure 4.6 (b).

Now we give the general description of the matrix relabeling algorithm. In the k th row of C , Lines 1-7 of Algorithm 4.10 store the value of i in X when the entry c_{ki} is 0, $i = u + 1, \dots, a + u$. Lines 8-14 store the value of i in Y when the entry c_{ki} is 1, $i = a + u + 1, \dots, b$. If $a = \sum_{i=u+1}^b c_{ki}$, X and Y have the same number of elements. Lines 15-17 swap the labels of X_i and Y_i , where X_i and Y_i are the i th element of X and Y respectively. For instance in Figure 4.8 (b), if we take $u = 2$, $k = 2$, $b = 10$ and $a = \sum_{i=3}^{10} c_{2i} = 5$, by Algorithm 4.10, $X = [5 \ 7]^T$, $Y = [8 \ 10]^T$, the labels of links

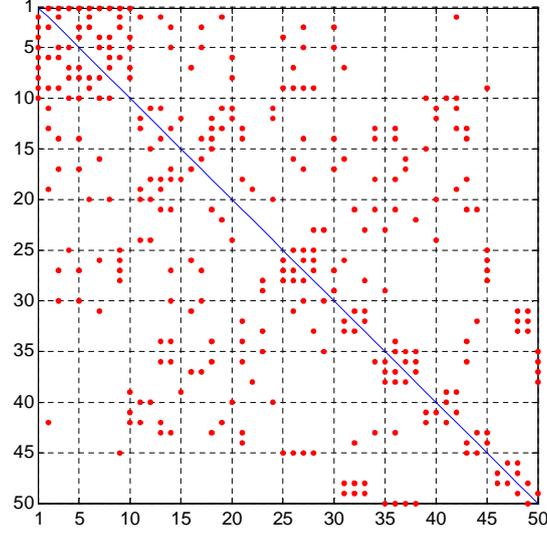


Figure 4.9: The LAM C after the relabeling of the first row.

5 and 8, 7 and 10 are swapped respectively.

Lines 1-2 of Algorithm 4.11 make the neighboring links of link 1 have the smaller labels than the other links. By lines 3-4, the labels of the links which are adjacent to both link 1 and 2 are smaller than those of the remaining links. Further, lines 5-6 let the labels of the links which are adjacent to all of links 1, 2 and 3 are smaller than those of the remaining links. Lines 7-14 make that the labels of the links which are adjacent to link i but not adjacent to links $1, \dots, i-1$, are smaller than the labels of the links which are not adjacent to link $1, \dots, i$, for $i = 2, \dots, L_G$. Figure 4.6 and 4.12 show examples of C before and after matrix relabeling.

Let $s_1 = \sum_{i=2}^{L_G} c_{1i}$, $s_2 = \sum_{i=3}^{s_1+1} c_{2i}$ and $s_3 = \sum_{i=4}^{s_2+2} c_{3i}$. After relabeling by Algorithm 4.11, the given LAM C satisfies:

- For $i = 2, \dots, s_1 + 1$, $c_{1i} = 1$; and for $i = s_1 + 2, \dots, L_G$, $c_{1i} = 0$.
- For $i = 3, \dots, s_2 + 2$, $c_{2i} = 1$ if $s_2 \geq 1$; and for $i = s_2 + 3, \dots, s_1 + 1$, $c_{2i} = 0$ if $s_1 \geq s_2 + 2$.
- For $i = 4, \dots, s_3 + 3$, $c_{3i} = 1$ if $s_3 \geq 1$; and for $i = s_3 + 4, \dots, s_2 + 2$, $c_{3i} = 0$ if $s_2 \geq s_3 + 2$.
- If link j ($j \geq s_{1+1}$) is adjacent to link i but not adjacent to links $1, 2, \dots, i-1$

```

C ← GROUPLABELSWAPPING(C, u, k, a, b)
1  m ← 0
2  for i = u + 1 to a + u do
3    if cki = 0 then
4      m ← m + 1
5      Xm ← i
6  m ← 0
7  for i = a + u + 1 to b do
8    if cki = 1 then
9      m ← m + 1
10   Ym ← i
11  for i = 1 to m do
12   C ← SwapLabel(C, Xi, Yi)

```

Figure 4.10: Meta-code for GROUPLABELSWAPPING.

($i \geq 2$), and link k ($k \geq s_{1+1}$) is not adjacent to all of links $1, 2, \dots, i$ ($i \geq 2$), then $j < k$.

If $s_3 \geq 1$ (which implies that $s_2 \geq 2$ and $s_1 \geq 3$), according to Theorem 6, links $2, 3, \dots, s_3 + 3$ are the left-neighboring links of links 1 and the links $s_3 + 4, \dots, s_1 + 1$ are the right-neighboring links of link 1, as illustrated in the example of Figure 4.13 where $s_1 = 9$ and $s_3 = 3$.

4.3.2 Construction algorithm

The construction algorithm converts the relabeled C into the matrix $E_{2 \times L_G}$, where the entries e_{1i} and e_{2i} denotes the two endnodes of link i . During the process of the construction, the zero entries of $E_{2 \times L_G}$ mean that the endnodes have not been determined yet. We will first show an example of graph construction, and then describe the general construction algorithm.

An example of graph construction from C

From the given LAM C in Figure 4.6 (b), we deduce that the graph has 50 links. Based on the LAM C , we will determine the endnodes of the 50 links. The construction consists of the following steps:

1. Let nodes 1 and 2 be the endnodes of link 1. According to Theorem 6, node 1 is also the endnode of links 2-5 and node 2 is also the endnode of links 6-10, as

```

(C, s1, s2, s3) ← MATRIXRELABELING(C)
1  s1 ← the sum of c1i, where i = 2 to LG
2  C ← GroupLabelSwapping(C, 1, 1, s1, LG)
3  E s2 ← the sum of c2i, where i = 3 to s1 + 1
4  C ← GroupLabelSwapping(C, 2, 2, s2, s1 + 1)
5  s3 ← the sum of c3i, where i = 4 to s2 + 2
6  C ← GroupLabelSwapping(C, 3, 3, s3, s2 + 2)
7   $\bar{s}$  ← s1 + 1
8  k ← 2
9  while  $\bar{s} < L_G$  and k ≤ LG do
10  s ← the sum of cki, where i =  $\bar{s}$  + 1 to LG
11  C ← GroupLabelSwapping(C,  $\bar{s}$ , k, s, LG)
12  k ← k + 1
13   $\bar{s}$  ←  $\bar{s}$  + s

```

Figure 4.11: Meta-code for MATRIXRELABELING.

shown in Figure 4.14 (a) and equation (4.2) below, where the numbers above the matrix are the link numbers.

$$E = \left[\begin{array}{cccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & \cdots & 50 \\ \hline 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{array} \right] \quad (4.2)$$

Let node 3 be the other endnode of link 2. The 2nd row of the LAM C shows that links 11-14 are adjacent to link 2. Hence, node 3 is also the endnode of links 11-14, as shown in Figure 4.14 (b) and equation (4.3).

$$E = \left[\begin{array}{cccccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & \cdots & 50 \\ \hline 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 0 & \cdots & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{array} \right] \quad (4.3)$$

Similarly, let node 4 be the endnode of link 3, 6 and 15-18 as shown in Figure 4.14 (c) and equation (4.4),

$$E = \left[\begin{array}{cccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ \hline 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 2 & 3 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 16 & 17 & 18 & 19 & \cdots & 50 \\ \hline 4 & 4 & 4 & 4 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{array} \right] \quad (4.4)$$

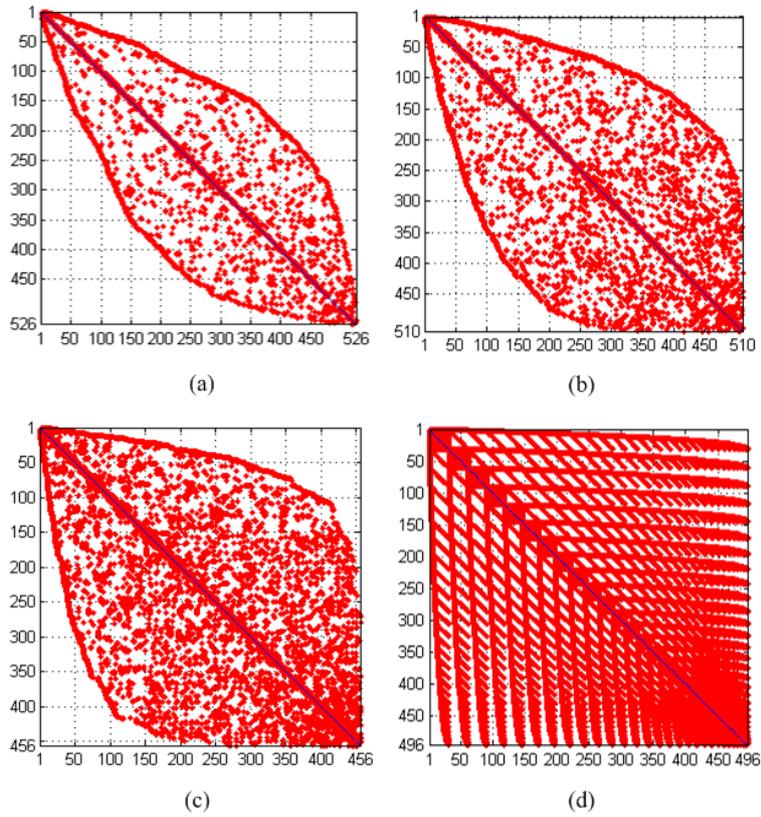


Figure 4.12: The relabeled C of four ER random graphs $G(N, p)$: (a) $N = 350$, $p = \frac{\log(N)}{2N}$; (b) $N = 200$, $p = \frac{\log(N)}{N}$; (c) $N = 100$, $p = \frac{2\log(N)}{N}$; (d) $N = 32$, $p = 1$, where $p = \frac{\log(N)}{N}$ is the threshold probability for the connectivity of the graph.

and let node 5 be the endnode of link 4, 8 and 19 as shown in Figure 4.14 (d) and equation (4.5),

$$E = \left[\begin{array}{cccccccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ \begin{array}{c} 1 \\ 2 \\ 15 \\ 4 \\ 0 \end{array} & \begin{array}{c} 1 \\ 3 \\ 16 \\ 4 \\ 0 \end{array} & \begin{array}{c} 1 \\ 4 \\ 17 \\ 4 \\ 0 \end{array} & \begin{array}{c} 1 \\ 5 \\ 18 \\ 4 \\ 0 \end{array} & \begin{array}{c} 1 \\ 0 \\ 19 \\ 5 \\ 0 \end{array} & \begin{array}{c} 2 \\ 4 \\ 20 \\ 0 \\ 0 \end{array} & \begin{array}{c} 2 \\ 0 \\ \dots \\ \dots \\ \dots \end{array} & \begin{array}{c} 2 \\ 5 \\ 50 \\ 0 \\ 0 \end{array} & \begin{array}{c} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \end{array} \right] \quad (4.5)$$

and let node 6 be the endnode of link 5, 16 and 20-23 as shown in Figure 4.14 (e)

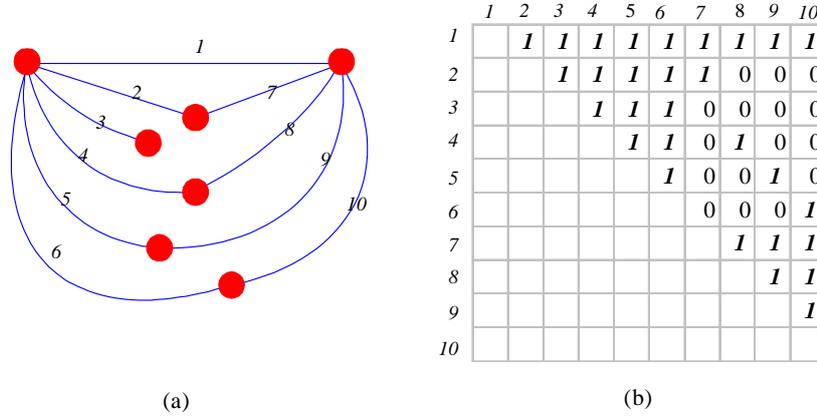


Figure 4.13: The LAM (a) relabeled by Algorithm 4.11 and its corresponding graph (b).

and equation (4.6).

$$E = \left[\begin{array}{cccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ \hline 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 2 & 3 & 4 & 5 & 6 & 4 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & \dots & 50 \\ \hline 4 & 4 & 4 & 4 & 5 & 6 & 6 & 6 & 6 & 0 & \dots & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{array} \right] \quad (4.6)$$

Then compute the LAM of the constructed part of the graph as shown in Figure 4.14 (f). The red dots are 1-entries which are from the given LAM in Figure 4.6 (b). The green dots are 1-entries which are determined by the red 1-entries. If the corresponding entries in the given matrix are not 1, then the matrix is not a LAM.

2. In the second step, we scan rows 6 to 10 of the LAM, since links 6 to 10 are incident to the same endnode. Let node 7 be the endnode of link 7, 21 and 24-25, and let node 8 be the endnode of link 9 and 20, and let node 9 be the endnode of

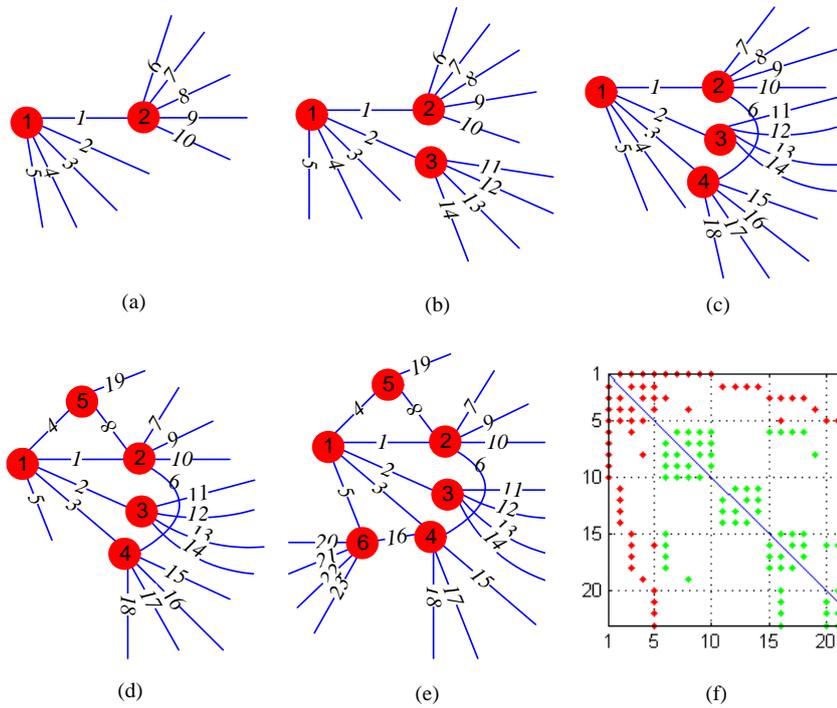


Figure 4.14: (a) The example of construction. The initialization is done in (a). Both or one of the two endnodes of links 1-23 are determined. (b) The LAM of the constructed part (links 1-23) of graph are computed. The green 1-entries are determined by the red 1-entries.

link 10, 14 and 26-27, as shown in Equation (4.7) and Figure 4.15.

$$E = \begin{bmatrix}
 \begin{array}{cccccccccccccccc}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\
 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\
 2 & 3 & 4 & 5 & 6 & 4 & 7 & 5 & 8 & 9 & 0 & 0 & 0 & 9 \\
 \hline
 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 \\
 4 & 4 & 4 & 4 & 5 & 6 & 6 & 6 & 6 & 7 & 7 & 9 & 9 & 0 \\
 0 & 6 & 0 & 0 & 0 & 8 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \hline
 29 & \dots & 50 \\
 0 & \dots & 0 \\
 0 & \dots & 0
 \end{array} \\
 \end{bmatrix} \tag{4.7}$$

3. Similarly, let node 10 be the endnode of link 11, 19 and 28-30, and let node 11 be the endnode of link 12, 18 and 31-35, and let node 12 be the endnode of link 13

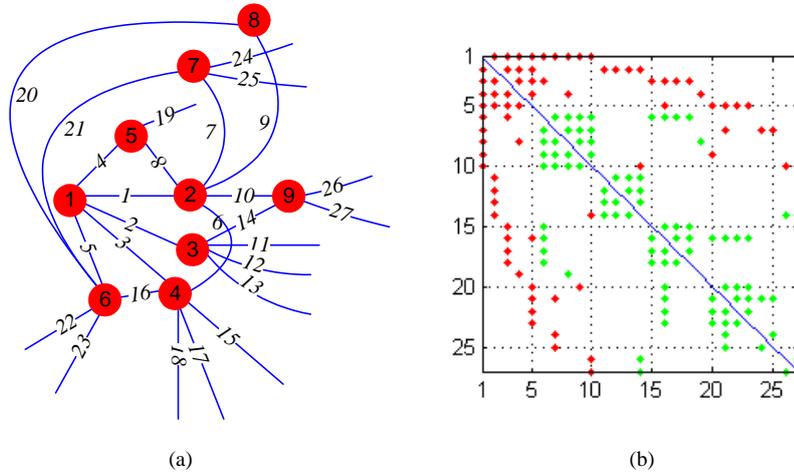


Figure 4.15: (a) The example of construction. Both or one of the two endnodes of links 1-27 are determined. (b) The LAM of the constructed part (links 1-27) of graph are computed. The green 1-entries are determined by the red 1-entries.

and 36, as shown in Equation (4.8) and Figure 4.16 (a).

$$E = \begin{bmatrix}
 \begin{array}{cccccccccccccccc}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\
 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\
 2 & 3 & 4 & 5 & 6 & 4 & 7 & 5 & 8 & 9 & 10 & 11 & 12 & 9 \\
 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 \\
 4 & 4 & 4 & 4 & 5 & 6 & 6 & 6 & 6 & 7 & 7 & 9 & 9 & 10 \\
 0 & 6 & 0 & 11 & 10 & 8 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & \dots & 50 \\
 10 & 10 & 11 & 11 & 11 & 11 & 11 & 12 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0
 \end{array} \\
 \end{bmatrix} \tag{4.8}$$

4. Constructing in this way, the two endnodes of all the links are eventually determined, as shown in Equation (4.9) and Figure 4.17 (a). The final structure of the matrix E exhibits the link list of the original graph G which consists of 30 nodes and 50 links. For example, link 36 connects node 12 and node 15 in G . The

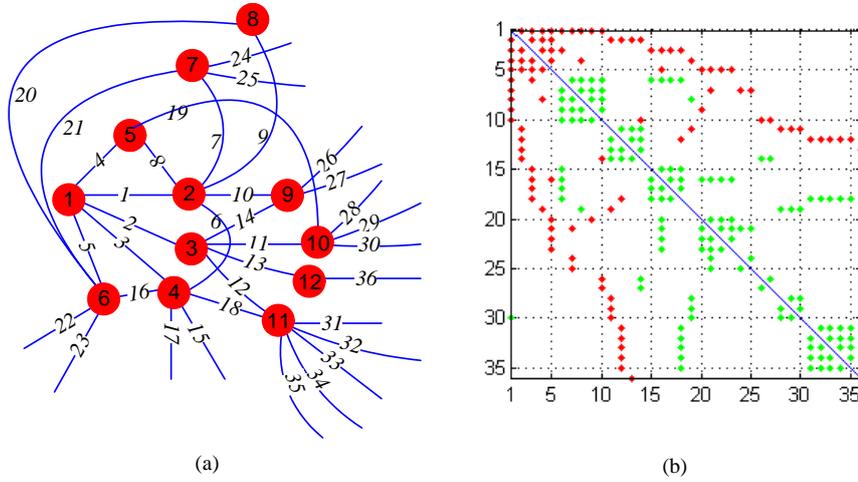


Figure 4.16: (a) The example of construction. Both or one of the two endnodes of links 1-36 are determined. (b) The LAM of the constructed part (links 1-36) of graph are computed. The green 1-entries are determined by the red 1-entries.

matrix E is readily transformed into the adjacency matrix of G .

$$E = \begin{bmatrix}
 \begin{array}{cccccccccccccccc}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\
 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\
 2 & 3 & 4 & 5 & 6 & 4 & 7 & 5 & 8 & 9 & 10 & 11 & 12 & 9 \\
 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 \\
 4 & 4 & 4 & 4 & 5 & 6 & 6 & 6 & 6 & 7 & 7 & 9 & 9 & 10 \\
 13 & 6 & 14 & 11 & 10 & 8 & 7 & 15 & 16 & 13 & 17 & 18 & 19 & 19 \\
 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 & 41 & 42 \\
 10 & 10 & 11 & 11 & 11 & 11 & 11 & 12 & 13 & 14 & 15 & 15 & 16 & 17 \\
 20 & 21 & 22 & 23 & 24 & 21 & 16 & 25 & 23 & 26 & 17 & 26 & 24 & 24 \\
 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 \\
 17 & 17 & 18 & 23 & 23 & 23 & 27 & 27 \\
 27 & 28 & 21 & 26 & 25 & 28 & 29 & 30
 \end{array} \\
 \end{bmatrix} \tag{4.9}$$

Initialization (The recognition of the endnodes of the first link and its neighboring links)

When $s_3 \geq 1$, Theorem 6 implies that $s_2 \geq 2$, $s_1 \geq 3$ and links $2, 3, \dots, s_3 + 3$ are incident to the endnode i_1^+ of link 1 and links $s_3 + 4, \dots, s_1 + 1$ are incident to the endnode i_1^- of link 1. Therefore, line 1-2 of Algorithm 4.18 initialize E by \mathcal{E} . The

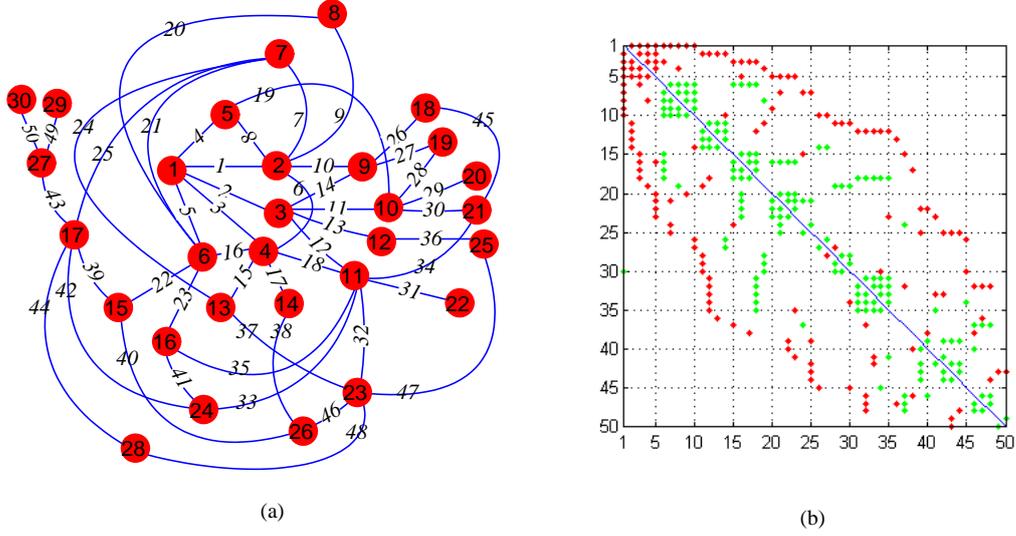


Figure 4.17: (a) The example of construction. The two endnodes of all links are determined. (b) The LAM of the constructed graph is computed. The green 1-entries are determined by the red 1-entries.

numbers above the matrix \mathcal{E} in (4.10) are the column numbers, which indicate the link numbers, and \mathcal{E} has the following structure,

$$\mathcal{E} = \left[\begin{array}{c|cccccccc} & 1 & 2 & \cdots & s_3 + 3 & s_3 + 4 & \cdots & s_1 + 1 & s_1 + 2 & \cdots & L_G \\ \hline 1 & 1 & 1 & \cdots & 1 & 2 & \cdots & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right] \quad (4.10)$$

When $s_3 = 0$, Theorem 6 cannot be used. However, the limited number of cases of $s_3 = 0$ enables us to accomplish the initialization with the detailed analysis in the Appendix C.

The recognition of the endnodes of the whole graph

Lines 1-2 of Algorithm 4.19 relabel the given LAM C and determine the initial state. In the initial state, link 1 is always incident to node 1 and 2. Some of the neighboring links of link 1 are incident to node 1, and the other neighboring links are incident to node 2. The second endnodes of the neighboring links of link 1 have not decided yet in the initial state.

Line 3 initiates the number of nodes N_G to 2. The two endnodes of link 1 are already determined. Starting with link 2 until link L_G (line 4), the number of nodes N_G increases by 1 (line 6) if the second endnode of link i is not determined (line 5).

```

 $E_{2 \times L_G} \leftarrow \text{INITIALIZATION}(C, s_1, s_2, s_3)$ 
1  if  $s_3 \geq 1$  then
2     $E \leftarrow \mathcal{E}$ 
3  else if  $s_1 = 1$  then
4     $E \leftarrow \mathcal{E}_1$ 
5  else if  $s_1 = 2$  then
6     $E_{2 \times L_G} \leftarrow \text{Initialization2}(C, s_2, s_3)$ 
7  else if  $s_1 = 3$  then
8     $E_{2 \times L_G} \leftarrow \text{Initialization3}(C, s_2, s_3)$ 
9  else if  $s_1 \geq 4$  then
10    $E_{2 \times L_G} \leftarrow \text{Initialization4}(C, s_1, s_2, s_3)$ 

```

Figure 4.18: Meta-code for INITIALIZATION.

Let the second endnode of link i be N_G (line 7). When link i is adjacent to link j , $j = i + 1, \dots, L_G$ (lines 8-9), let the first endnode of link j be N_G (line 11) if the first endnode of link j is not determined (line 10). If the first endnode of link j is determined but the second endnode is not determined and links i and j do not share the first endnode (line 12), let the second endnode of link j be N_G (line 13).

4.3.3 Worst case complexity of MARINLINGA

Algorithm 4.7 has a complexity of $O(L_G)$. The complexity of Algorithm 4.11 can be computed as follows. Line 1 has a complexity of $O(L_G)$. In the worst case, the function of line 2, Algorithm 4.10 has a complexity of $O(L_G^2)$, if m in line 15 of Algorithm 4.10 is proportional to L_G . The worst case complexity of lines 3-6 is also $O(L_G^2)$. Hence, lines 1-6 have a complexity of $O(L_G^2)$. Neglect $O(1)$ operations of lines 7-8. The times that lines 9-14 are executed is stored in k . If k is proportional to L_G , m in line 15 of Algorithm 4.10 must be bounded by a constant, then the complexity of line 11 is $O(L_G)$. If k is bounded, the complexity of line 11 will be $O(L_G^2)$. Therefore, lines 9-14 have a worst case complexity of $O(L_G^2)$. Hence, the complexity of Algorithm 4.11 is $O(L_G^2)$.

Algorithm C.2, C.4 and C.6 have a worst case complexity of $O(1)$, hence the complexity of Algorithm 4.18 is also $O(1)$. Lines 4-18 of the main Algorithm 4.19 have a worst case complexity of $O(L_G^2)$. In summary, the worst case complexity of the MARINLINGA is $O(L_G^2)$. Since the number of links of *the original graph* G and the number of nodes of *the line graph* $l(G)$ are equal, $L_G = N_{l(G)}$, the worst case complexity of the MARINLINGA is written as $O(N_{l(G)}^2)$.

```

 $E_{2 \times L_G} \leftarrow \text{MARINLINGA}(C)$ 
1  $(C, s_1, s_2, s_3) \leftarrow \text{MATRIXRELABELING}(C)$ 
2  $E_{2 \times L_G} \leftarrow \text{INITIALIZATION}(C, s_1, s_2, s_3)$ 
3  $N \leftarrow 2$ 
4 for  $i = 2$  to  $L_G$  do
5     if  $e_{2i} = 0$  then
6          $N \leftarrow N + 1$ 
7          $e_{2i} \leftarrow N$ 
8     for  $j = i + 1$  to  $L_G$  do
9         if  $c_{ij} = 1$  then
10            if  $e_{1j} = 0$  then
11                 $e_{1j} \leftarrow N$ 
12            else if  $e_{2j} = 0$  and  $e_{1i} \neq e_{1j}$  then
13                 $e_{2j} \leftarrow N$ 

```

Figure 4.19: Meta-code for MARINLINGA.

4.4 Chapter conclusion

We have presented a new algorithm MARINLINGA for reverse line graph construction. By introducing the concept of LAM, we transformed the problem of reverse line graph construction into the problem of constructing a graph from the LAM. MARINLINGA consists of two sub-algorithms: the matrix relabeling algorithm and the construction algorithm. The matrix relabeling algorithm preprocesses the LAM into the special order by which we can determine the neighboring links of the first link and the endnodes of the first link incident to the neighboring links. The construction algorithm makes the first two nodes be the endnodes of the first link by default, and thereafter, determines the endnodes of the remaining links. MARINLINGA has a worst case complexity of $O(N_{l(G)}^2)$, where $N_{l(G)}$ denotes the number of nodes of the line graph. The complexity of Roussopoulos' algorithm mentioned in [49] is $O(N_{l(G)} + L_{l(G)})$, where $N_{l(G)}$ and $L_{l(G)}$ are number of nodes and links of the line graph. Since $L_{l(G)} = O(N_{l(G)}^2)$ in worst case, the complexity of Roussopoulos' algorithm is also $O(N_{l(G)}^2)$ in worst case. However, this analysis neglects the computational time of a sub-algorithm that determines the maximal connected common subgraph in each iteration. Finding a maximally connected common subgraph is an *NP*-complete problem [52], implying that Roussopoulos' algorithm is, in fact, not polynomial in worst case.

Chapter 5

ILIGRA

This chapter presents a new and efficient algorithm, ILIGRA, for inverse line graph construction. Given a line graph H , ILIGRA constructs its root graph G with the time complexity being linear in the number of nodes in H . If ILIGRA does not know whether the given graph H is a line graph, it firstly assumes that H is a line graph and starts its root graph construction. During the root graph construction, ILIGRA checks whether the given graph H is a line graph and ILIGRA stops once it finds H is not a line graph. The time complexity of ILIGRA with line graph checking is linear in the number of links in the given graph H . For sparse line graphs of any size and for dense line graphs of small size, numerical results of the running time show that ILIGRA outweighs all currently available algorithms.

This chapter is organized as follows. ILIGRA is presented in Section 5.1. Section 5.6 demonstrates how ILIGRA works on a descriptive example. Numerical comparisons of ILIGRA with the algorithms of Lehot, Roussopoulos, and Degiorgi and Simon for different types of line graphs and non-line graphs are presented in Section 5.7. Finally, we conclude in Section 5.8.

5.1 Notation

Table 5.1 summarizes the notation, which is used in the presentation of ILIGRA. According to the definition of the line graph, each node in a line graph $H(N_H, L_H)$ corresponds to a link in its root graph $G(N, L)$. Hence, the number of nodes N_H in the line graph H and the number of links L in the root graph G are equal, $N_H = L$. We always use n (or n with subscript) to denote a node in H . The link in G corresponding to node n in H is denoted by l_n . In the remainder of the paper, we use v (or v with subscript) to denote a node in G . Denote by $\mathcal{N}_b(n)$ the set of the nodes in H which are adjacent to node n , which are also called the neighbors¹ of node n . Denote by $\mathcal{L}_b(l_n)$ the set of the

¹A node is the neighbor of another node if they are adjacent with each other.

Table 5.1: Notation

$G(N, L)$	The root graph with N nodes and L links
$H(N_H, L_H)$	The line graph of G with N_H nodes and L_H links
n	The node n in H
\mathcal{N}	The set of all the nodes in H
\mathcal{N}_w	The set of the nodes in H , corresponding to the links in G whose incident nodes are not yet determined
\mathcal{N}_h	The set of the nodes in H , corresponding to the links in G of which one incident node is determined
$\mathcal{N}_b(n)$	The set of the neighbors of node n in H
l_n	The link in G which corresponds to node n in H
$\mathcal{L}_b(l_n)$	The set of the links in G which correspond to the nodes in $\mathcal{N}_b(n)$
v_{l_n}	The first identified incident node of link l_n in G
$\text{ADDNODE}(G, v)$	The function which adds a node v to G
$\text{ADDLINK}(G, v_1, v_2)$	The function which adds a link $v_1 \sim v_2$ to G

links in G which correspond to the nodes in $\mathcal{N}_b(n)$. Every link in the root graph G has two incident nodes². In order to construct the root graph G from a given line graph H , we have to determine the two incident nodes of every link in G . In the root graph G , we denote by v_{l_n} the incident node of link l_n which is firstly determined during the algorithm's execution. The set of the nodes in H , which correspond to the links in G whose incident nodes are not yet determined, is denoted by \mathcal{N}_w . The set of the nodes in H , which correspond to the links in G of which one incident node is determined, is denoted by \mathcal{N}_h .

5.2 Concept

The nodes in a line graph $H(N_H, L_H)$ are denoted by $n_1, n_2, n_3, \dots, n_{N_H}$, and the corresponding links in the root graph G are denoted by $l_{n_1}, l_{n_2}, l_{n_3}, \dots, l_{n_{N_H}}$. Before reading the information from H , it is unknown how the links $l_{n_1}, l_{n_2}, l_{n_3}, \dots, l_{n_{N_H}}$ connect the nodes in G , and even the number of nodes N in G is unknown.

Suppose that link l_{n_1} is incident to v_1 and v_2 in G . From the line graph H , the set $\mathcal{N}_b(n_1)$ of the neighbors of node n_1 in H is known, and the set $\mathcal{L}_b(l_{n_1})$ of the links in G , which correspond to the nodes in $\mathcal{N}_b(n_1)$, is also known. By the definition of a line graph, the links in $\mathcal{L}_b(l_{n_1})$ are the neighboring links of link l_{n_1} , hence, the links in $\mathcal{L}_b(l_{n_1})$ should be incident to either v_1 or v_2 . If the links in $\mathcal{L}_b(l_{n_1})$ which are incident

²A node is an incident node of a link if it is incident to that link.

to v_1 are known, the rest of links in $\mathcal{L}_b(l_{n_1})$ must be incident to v_2 . Unfortunately, it is unknown which links in $\mathcal{L}_b(l_{n_1})$ are incident to v_1 .

When considering links $l_{n_2}, l_{n_3}, \dots, l_{n_{N_H}}$, the same problem appears. The difficulty in constructing the root graph G lies in partitioning the set of the neighboring links into two complementary subsets of links: the links which are incident to the first incident node of the concerned link, and the other links which are incident to the second incident node of that link.

Without loss of generality, suppose that $\mathcal{L}_b(l_{n_1}) = \{l_{n_2}, l_{n_3}, \dots, l_{n_{10}}\}$. Suppose that the set $\mathcal{L}_b(l_{n_1})$ of the neighboring links of l_{n_1} are partitioned successfully into two subsets: $\mathcal{L}_{b,v_1}(l_{n_1}) = \{l_{n_2}, l_{n_3}, l_{n_4}\}$ where the links are incident to v_1 , and $\mathcal{L}_{b,v_2}(l_{n_1}) = \{l_{n_5}, l_{n_6}, \dots, l_{n_{10}}\}$ where the links are incident to v_2 . Then, the set $\mathcal{L}_b(l_{n_2})$ of the neighboring links of l_{n_2} is automatically partitioned: the links $l_{n_2}, l_{n_3}, l_{n_4}$ are incident to v_1 , and the rest of links in $\mathcal{L}_b(l_{n_2})$ are incident to the second incident node of l_{n_2} . Similarly, the sets of the neighboring links of links $l_{n_3}, l_{n_4}, \dots, l_{n_{10}}$, are also automatically partitioned. Assuming H is a connected line graph, the sets of the neighboring links of all the links in G can be partitioned by iterating the described process. This is the basic idea of ILIGRA.

Partitioning the set $\mathcal{L}_b(l_{n_1})$ of the neighboring links of l_{n_1} becomes a crucial task in the root graph construction. The theorems in Section 5.3 provide the theoretical basis for this task.

5.3 Theoretical preliminaries

Theorem 7 *Suppose that two adjacent nodes n_1 and n_2 in H correspond to links l_{n_1} and l_{n_2} , respectively in G , where l_{n_1} is incident to v_1 and v_2 and where v_1 is also incident to l_{n_2} , as shown in Figure 5.1 (a). Then, for each $n \in \mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$, the corresponding link l_n in G must be incident to v_2 , and the nodes in $\mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$ must form a clique in H .*

Proof. For each $n \in \mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$, the corresponding link l_n in G has to be incident to either v_1 or v_2 , since n is adjacent to n_1 in H . Because n is not adjacent to n_2 , l_n in G can only be incident to v_2 . Since the corresponding links of all the nodes $\in \mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$ are incident to v_2 , the nodes in $\mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$ must be fully connected with each other and form a clique in H . ■

Using Theorem 7, ILIGRA determines the first incident node of the links corresponding to the nodes in $\mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$. The nodes in $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ may form a clique in H with the corresponding links being incident to v_1 in G , as shown in Figure 5.1 (a). There may also exist a node $n_u \in \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ which is not adjacent to any other node in $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ and whose corresponding link is incident to v_2 and v_3 , as shown in Figure 5.1 (b) where l_{n_1} , l_{n_2} and l_{n_u} form a triangle in G . Using Theorem

7, ILIGRA determines the first incident node of the links corresponding to the nodes in $\mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$. The nodes in $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ may form a clique in H with the corresponding links being incident to v_1 in G , as shown in Figure 5.1 (a). There may also exist a node $n_u \in \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ which is not adjacent to any other node in $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ and whose corresponding link is incident to v_2 and v_3 , as shown in Figure 5.1 (b) where l_{n_1} , l_{n_2} and l_{n_u} form a triangle in G . If there are three or more nodes in the set $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$, we can identify this special node n_u from the set $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ by Theorem 8.

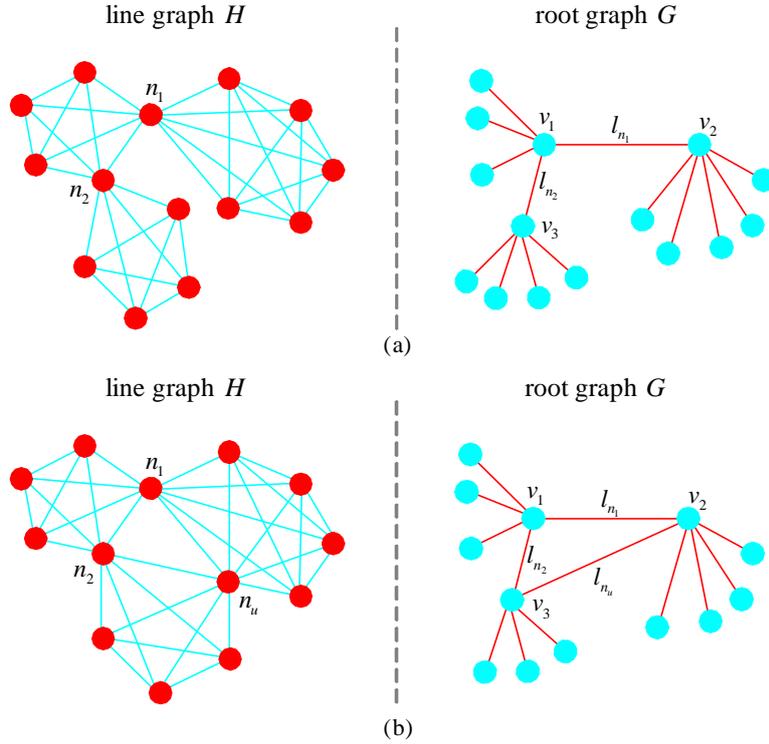


Figure 5.1: Scenarios in Theorem 7 and 8. Each node (red) in H corresponds to a link (red) in G .

Theorem 8 Suppose that two adjacent nodes n_1 and n_2 in H correspond to links l_{n_1} and l_{n_2} respectively in G , where l_{n_1} is incident to v_1 and v_2 and where l_{n_2} is incident to v_1 and v_3 . Suppose that $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| \geq 3$. If there exists $n_u \in \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ such that n_u is not adjacent to any other node in $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$, then link l_{n_u} must be incident to v_2 and v_3 in G .

Proof. Since $n_u \in \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$, l_{n_u} can be incident to v_1 or be incident to both v_2 and v_3 . If l_{n_u} is incident to v_1 , n_u should be adjacent to at least one other node in

$\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$, since $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| \geq 3$. Because n_u is not adjacent to any other node in $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$, l_{n_u} can only be incident to v_2 and v_3 , as shown in Figure 5.1 (b). ■

If there are only one or two nodes in the set $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$, we can identify this special node n_u from the set $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ by Theorem 11, which can be proved by the combination of Lemma 9 and 10.

Lemma 9 *Suppose that two adjacent nodes n_1 and n_2 in H correspond to links l_{n_1} and l_{n_2} respectively in G , where l_{n_1} is incident to v_1 and v_2 and where l_{n_2} is incident to v_1 and v_3 . Suppose that there are more than 3 nodes in H and $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| \leq 2$. If there exists $n_u \in \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ such that $\mathcal{N}_b(n_u) \subseteq \mathcal{N}_b(n_1) \cup \mathcal{N}_b(n_2)$, then link l_{n_u} must be incident to v_2 and v_3 in G , unless H is one of the line graphs in Figure 5.2.*

Proof. Since $n_u \in \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$, l_{n_u} can be incident to v_1 or be incident to both v_2 and v_3 . Given that there are more than 3 nodes in H , if l_{n_u} is incident to v_1 and H is not one of the line graphs in Figure 5.2, there must be at least one neighbor of n_u which is adjacent to neither n_1 nor n_2 . Because $\mathcal{N}_b(n_u) \subseteq \mathcal{N}_b(n_1) \cup \mathcal{N}_b(n_2)$, l_{n_u} can only be incident to v_2 and v_3 . ■

Lemma 10 *Suppose that two adjacent nodes n_1 and n_2 in H correspond to links l_{n_1} and l_{n_2} respectively in G , where l_{n_1} is incident to v_1 and v_2 and where l_{n_2} is incident to v_1 and v_3 . Suppose that H is one of the line graphs in Figure 5.2, then l_{n_u} can be incident to v_1 or be incident to both v_2 and v_3 and the resulting root graphs are isomorphic.*

Theorem 11 *Suppose that two adjacent nodes n_1 and n_2 in H correspond to links l_{n_1} and l_{n_2} respectively in G , where l_{n_1} is incident to v_1 and v_2 and where l_{n_2} is incident to v_1 and v_3 . Suppose that there are more than 3 nodes in H and $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| \leq 2$. If there exists $n_u \in \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ such that $\mathcal{N}_b(n_u) \subseteq \mathcal{N}_b(n_1) \cup \mathcal{N}_b(n_2)$, then we can let link l_{n_u} be incident to v_2 and v_3 and obtain the correct root graph G .*

Proof. Theorem 11 is proved by combining the Lemma 9 and 10. ■

ILIGRA uses Theorem 7, 8 and 11 to determine which links in $\mathcal{L}_b(l_{n_1})$ are incident to v_1 and which else are incident to v_2 . Then, for each link in $\mathcal{L}_b(l_{n_1})$, the first incident node has been determined.

5.4 Algorithm description

ILIGRA starts by setting G to an empty graph (*line 1*). Initially, nothing in G is determined, hence $\mathcal{N}_w = \mathcal{N}$ and $\mathcal{N}_h = \emptyset$ (*line 2*), where \mathcal{N}_w and \mathcal{N}_h are defined in Table 5.1. Then ILIGRA picks an arbitrary node n_1 in the set \mathcal{N}_w and picks an arbitrary neighbor n_2 of n_1 in the set $\mathcal{N}_b(n_1)$ (*lines 3-4*). Two nodes v_1 and v_2 are added to the

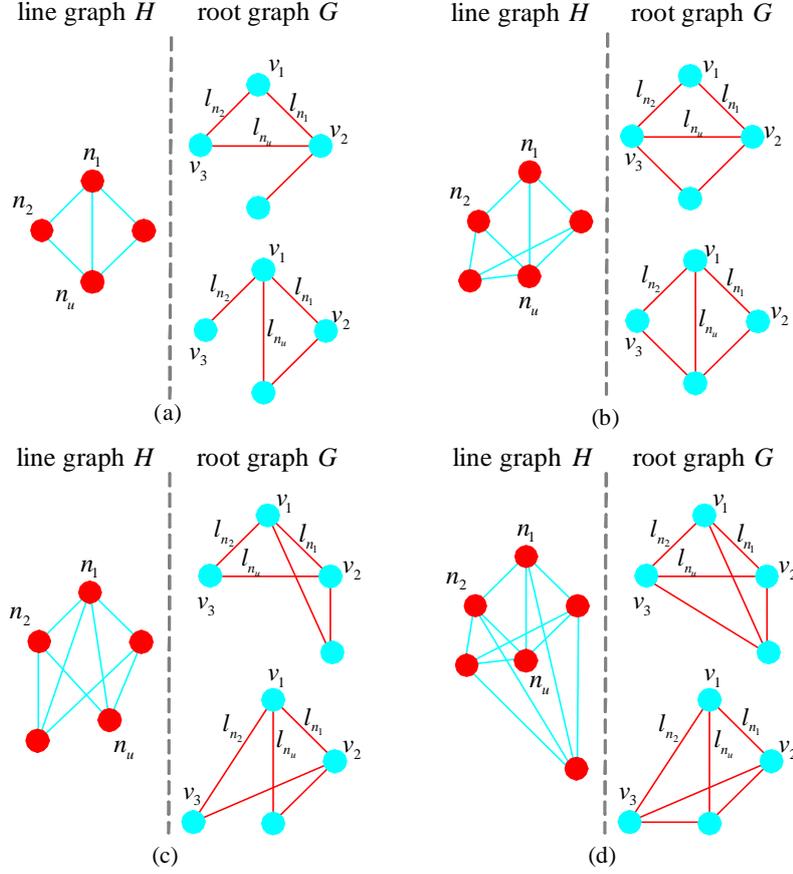


Figure 5.2: Scenarios in Lemma 9 and 10. (a) $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| = 1$ and $L = 4$. (b) $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| = 1$ and $L = 5$. (c) $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| = 2$ and $L = 5$. (d) $|\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)| = 2$ and $L = 6$.

root graph G (line 5), and link $l_{n_1} = v_1 \sim v_2$ is added to the root graph G (line 6). Since the incident nodes of link l_{n_1} have been determined in G , node n_1 is removed from \mathcal{N}_w (line 6). Then v_1 is chosen³ to be incident to link l_{n_2} (line 7). Since the first incident node of link l_{n_2} is determined, node n_2 is moved from \mathcal{N}_w to \mathcal{N}_h (line 7).

According to the definition of the line graph, the links in $\mathcal{L}_b(l_{n_1})$ have a node in common with link l_{n_1} in G . Since l_{n_1} is incident to v_1 and v_2 , the links in $\mathcal{L}_b(l_{n_1})$ should also be incident to either v_1 or v_2 . By Theorem 7, ILIGRA determines that the links in $\mathcal{L}_b(l_{n_1}) \setminus \mathcal{L}_b(l_{n_2})$, corresponding to the nodes in $n \in \mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$, are incident to v_2 . For each node n in $\mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$, ILIGRA sets the first identified incident node v_{l_n} of the corresponding link l_n to v_2 , and moves n from \mathcal{N}_w to \mathcal{N}_h (lines 8-9).

³ILIGRA arbitrarily chooses a node from v_1 and v_2 and lets it be incident to l_{n_2} .

```

 $G \leftarrow \text{ILIGRA}(H)$ 
1  $G \leftarrow$  an empty graph;
2  $\mathcal{N} \leftarrow$  the set of nodes in  $H$ ;  $\mathcal{N}_w \leftarrow \mathcal{N}$ ;  $\mathcal{N}_h \leftarrow \emptyset$ ;
3  $n_1 \leftarrow$  an arbitrary node  $\in \mathcal{N}_w$ ;
4  $n_2 \leftarrow$  an arbitrary node  $\in \mathcal{N}_b(n_1)$ ;
5  $\text{ADDNODE}(G, v_1)$ ;  $\text{ADDNODE}(G, v_2)$ ;
6  $\text{ADDLINK}(G, v_1, v_2)$ ;  $\mathcal{N}_w \leftarrow \mathcal{N}_w \setminus \{n_1\}$ ;
7  $v_{l_{n_2}} \leftarrow v_1$ ;  $\mathcal{N}_h \leftarrow \mathcal{N}_h \cup \{n_2\}$ ;  $\mathcal{N}_w \leftarrow \mathcal{N}_w \setminus \{n_2\}$ ;
8 for each  $n \in \mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2)$  do
9    $v_{l_n} \leftarrow v_2$ ;  $\mathcal{N}_h \leftarrow \mathcal{N}_h \cup \{n\}$ ;  $\mathcal{N}_w \leftarrow \mathcal{N}_w \setminus \{n\}$ ;
10  $J \leftarrow \mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2)$ ;
11 if  $1 \leq |J| \leq 2$  then
12   if  $|J| = 3$  then
13      $G$  is  $K_{1,3}$  or  $K_3$ . Exit.
14   else if  $\exists n_u \in J$  such that  $\mathcal{N}_b(n_u) \subseteq \mathcal{N}_b(n_1) \cup \mathcal{N}_b(n_2)$  then
15      $v_{l_{n_u}} \leftarrow v_2$ ;  $\mathcal{N}_h \leftarrow \mathcal{N}_h \cup \{n_u\}$ ;
16      $\mathcal{N}_w \leftarrow \mathcal{N}_w \setminus \{n_u\}$ ;  $J \leftarrow J \setminus \{n_u\}$ ;
17 else if  $|J| \geq 3$  and  $\exists n_u \in J$  such that  $n_u$  is not adjacent to
    any other node in  $J$  then
18    $v_{l_{n_u}} \leftarrow v_2$ ;  $\mathcal{N}_h \leftarrow \mathcal{N}_h \cup \{n_u\}$ ;
19    $\mathcal{N}_w \leftarrow \mathcal{N}_w \setminus \{n_u\}$ ;  $J \leftarrow J \setminus \{n_u\}$ ;

```

Figure 5.3: ILIGRA: part I.

ILIGRA sets J to the intersection of $\mathcal{N}_b(n_1)$ and $\mathcal{N}_b(n_2)$ (line 10). If there are only 1 or 2 nodes in J , and if there are only 3 nodes in H (namely 3 links in G), the root graph G can be $K_{1,3}$ or K_3 (lines 11-13). If $1 \leq |J| \leq 2$ and if there exists $n_u \in J$ such that any neighbor of n_u is also a neighbor of either n_1 or n_2 , according to Theorem 11, link l_{n_u} should be incident to v_2 . ILIGRA sets $v_{l_{n_u}}$ to v_2 , and adds n_u to \mathcal{N}_h and removes n_u from \mathcal{N}_w and removes n_u from J (lines 14-16). If $|J| \geq 3$ and if there exists $n_u \in J$ such that n_u is not adjacent to any other node in J , according to Theorem 8, link l_{n_u} should be incident to v_2 . ILIGRA sets $v_{l_{n_u}}$ to v_2 , and adds n_u to \mathcal{N}_h and removes n_u from \mathcal{N}_w and removes n_u from J (lines 17-19).

Since node n_u has been removed from J , the rest of links in J should be incident to v_1 . For each n in J which are adjacent with both n_1 and n_2 , ILIGRA sets v_{l_n} to v_1 , and adds n to \mathcal{N}_h and removes n from \mathcal{N}_w (lines 20-21). The nodes in J should be fully connected to each other, since the corresponding links are all incident to v_1 . If the nodes in J do not form a clique in H , then H is not a line graph (lines 22-23). The nodes in $\mathcal{N}_b(n_1) \setminus J$ should also be fully connected to each other, since the corresponding

```

 $G \leftarrow \text{ILIGRA}(H)$ 
20 for each  $n \in J$  do
21    $v_{l_n} \leftarrow v_1; \mathcal{N}_h \leftarrow \mathcal{N}_h \cup \{n\}; \mathcal{N}_w \leftarrow \mathcal{N}_w \setminus \{n\};$ 
22 if  $J \neq \emptyset$  and  $J$  is not a clique in  $H$  then
23    $H$  is not a line graph. Exit.
24 if  $|\mathcal{N}_b(n_1) \setminus J| \neq \emptyset$  and  $\mathcal{N}_b(n_1) \setminus J$  is not a clique then
25  $H$  is not a line graph. Exit.
26 while  $\mathcal{N}_h \neq \emptyset$  do
27    $n \leftarrow$  an arbitrary node in  $\mathcal{N}_h$ ;
28    $\text{ADDNODE}(G, v); \text{ADDLINK}(G, v_{l_n}, v);$ 
29    $\mathcal{N}_h \leftarrow \mathcal{N}_h \setminus \{n\}; \mathcal{C} \leftarrow \emptyset;$ 
30   for each  $n_r \in \mathcal{N}_b(n)$  do
31     if  $n_r \in \mathcal{N}_h$  and  $v_{l_n} \neq v_{l_{n_r}}$  do
32        $\mathcal{C} \leftarrow \mathcal{C} \cup \{n_r\};$ 
33        $\text{ADDLINK}(G, v_{l_{n_r}}, v); \mathcal{N}_h \leftarrow \mathcal{N}_h \setminus \{n_r\};$ 
34     else if  $n_r \in \mathcal{N}_w$  then
35        $\mathcal{C} \leftarrow \mathcal{C} \cup \{n_r\}; v_{l_{n_r}} \leftarrow v;$ 
36        $\mathcal{N}_h \leftarrow \mathcal{N}_h \cup \{n_r\}; \mathcal{N}_w \leftarrow \mathcal{N}_w \setminus \{n_r\};$ 
37   if  $\mathcal{C} \neq \emptyset$  and  $\mathcal{C}$  is not a clique in  $H$  then
38      $H$  is not a line graph. Exit.

```

Figure 5.4: ILIGRA: part II.

links are all incident to v_2 . If the nodes in $\mathcal{N}_b(n_1) \setminus J$ do not form a clique in H , then H is not a line graph (lines 24-25).

The loop (lines 26-38) runs until \mathcal{N}_h is an empty set. ILIGRA picks an arbitrary node n in \mathcal{N}_h (line 27). ILIGRA adds a node v and a link l_n between v_{l_n} and v to G (line 28), and removes n from \mathcal{N}_h (line 29). ILIGRA sets \mathcal{C} to an empty set (line 29). For each neighbor n_r of n , if $n_r \in \mathcal{N}_h$ and $v_{l_n} \neq v_{l_{n_r}}$, ILIGRA adds link l_{n_r} between $v_{l_{n_r}}$ and v to G , and removes n_r from \mathcal{N}_h , and adds n_r to \mathcal{C} (lines 30-33); If $n_r \in \mathcal{N}_w$, ILIGRA sets $v_{l_{n_r}}$ to v , and moves n_r from \mathcal{N}_w to \mathcal{N}_h , and adds n_r to \mathcal{C} (lines 34-36). The nodes in \mathcal{C} should be fully connected with each other, since the corresponding links are all incident to v . If the nodes in \mathcal{C} do not form a clique in H , H is not a line graph (lines 37-38). If H is a connected graph, \mathcal{N}_w should be an empty set when \mathcal{N}_h becomes an empty set. While $\mathcal{N}_w \neq \emptyset$, repeat lines 3-38. For each component of a given disconnected line graph, lines 3-38 will be executed once. If the input graphs are line graphs, lines 22-25 and 37-38 can be skipped, which are used to check whether the given graph is a line graph.

5.5 Complexity

The lines 1-21 of ILIGRA examines all the neighbors of the n_1 in H , with the complexity $O(N_H)$, where N_H is the number of nodes in H . The lines 22-25, which check whether H is a line graph, have the complexity $O(N_L)$, where N_L is the number of links in H . The lines 26-36 have the complexity $O(N_H)$. The lines 37-38 check whether H is a line graph and have the complexity $O(N_L)$. Hence, the overall complexity of ILIGRA with checking if H is a line graph is $O(N_L)$, and the complexity of ILIGRA without checking is $O(N_H)$.

5.6 An example

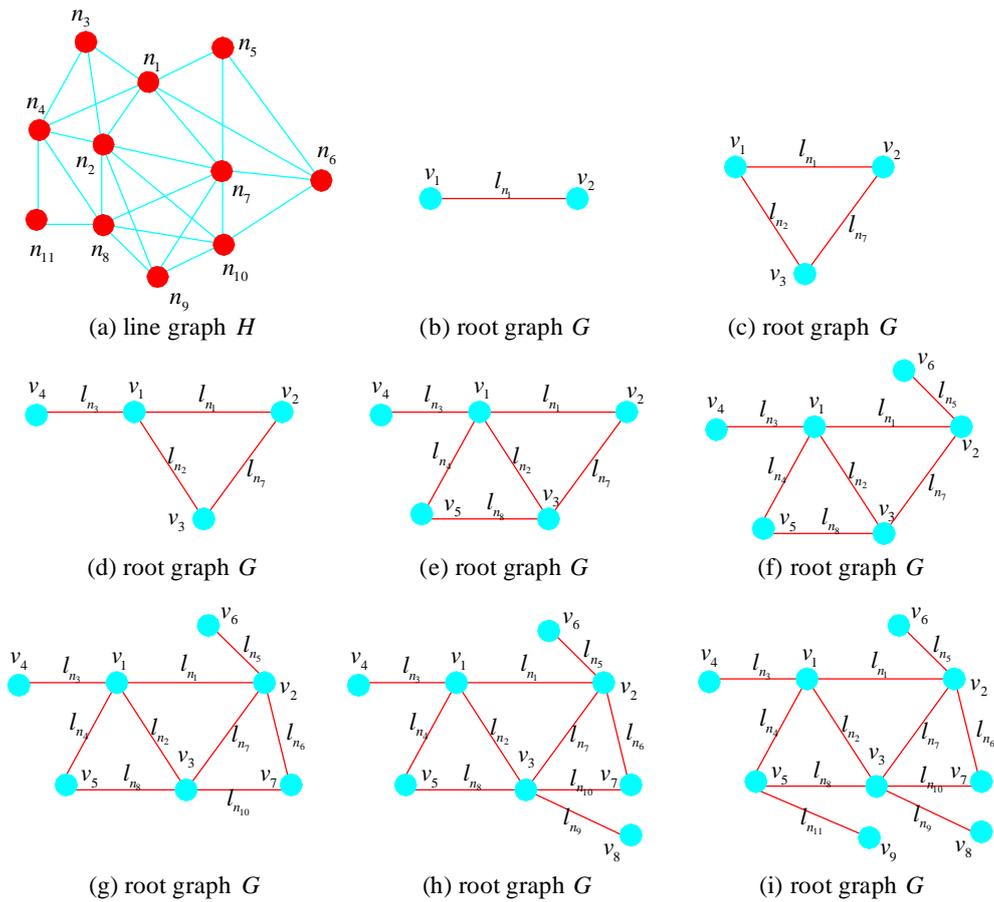


Figure 5.5: An example shows how ILIGRA constructs G from a given H .

In this section, we use an example depicted in Figure 5.5 to show how ILIGRA works. Given a line graph H shown in Figure 5.5 (a), ILIGRA constructs its root graph G incrementally as shown in Figure 5.5 (b) to (i).

Initially, set G to an empty graph. We have $\mathcal{N}_w = \{n_1, n_2, \dots, n_{11}\}$ and $\mathcal{N}_h = \emptyset$. Add nodes v_1 and v_2 to G , and add link l_{n_1} between v_1 and v_2 to G as shown in Figure 5.5 (b), and $\mathcal{N}_w = \{n_2, n_3, \dots, n_{11}\}$. Set $v_{l_{n_2}}$ to v_1 , $\mathcal{N}_w = \{n_3, n_4, \dots, n_{11}\}$ and $\mathcal{N}_h = \{n_2\}$. Since $\mathcal{N}_b(n_1) \setminus \mathcal{N}_b(n_2) = \{n_5, n_6\}$, according to Theorem 7, set $v_{l_{n_5}}$ to v_2 and also set $v_{l_{n_6}}$ to v_2 . We have $\mathcal{N}_w = \{n_3, n_4, n_7, n_8, \dots, n_{11}\}$ and $\mathcal{N}_h = \{n_2, n_5, n_6\}$. Since $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2) = \{n_3, n_4, n_7\}$ and none of n_3 and n_4 is not adjacent to n_7 , according to Theorem 8, set $v_{l_{n_7}}$ to v_2 . Now $\mathcal{N}_w = \{n_3, n_4, n_8, n_9, n_{10}, n_{11}\}$ and $\mathcal{N}_h = \{n_2, n_5, n_6, n_7\}$. For the two nodes n_3 and n_4 in $\mathcal{N}_b(n_1) \cap \mathcal{N}_b(n_2) \setminus \{n_7\}$, the corresponding links should be incident to v_1 . Hence, set both $v_{l_{n_3}}$ and $v_{l_{n_4}}$ to v_1 . Now $\mathcal{N}_w = \{n_8, n_9, n_{10}, n_{11}\}$ and $\mathcal{N}_h = \{n_2, n_3, n_4, n_5, n_6, n_7\}$.

Take n_2 from \mathcal{N}_h . Add a node v_3 to G and add link l_{n_2} between v_3 and $v_{l_{n_2}}$ ($v_{l_{n_2}}$ has been found to be v_1 previously), as shown in Figure 5.5 (c). Now $\mathcal{N}_h = \{n_3, n_4, n_5, n_6, n_7\}$. We have $\mathcal{N}_b(n_2) = \{n_1, n_3, n_4, n_7, n_8, n_9, n_{10}\}$. Since $n_7 \in \mathcal{N}_h$ and $v_{l_{n_7}} = v_2 \neq v_{l_{n_2}} = v_1$, add l_{n_7} between v_2 and v_3 to G . Now $\mathcal{N}_h = \{n_3, n_4, n_5, n_6\}$. Since n_8, n_9 and n_{10} belong to \mathcal{N}_w , set $v_{l_{n_8}}, v_{l_{n_9}}$ and $v_{l_{n_{10}}}$ to v_3 . Now $\mathcal{N}_w = \{n_{11}\}$ and $\mathcal{N}_h = \{n_3, n_4, n_5, n_6, n_8, n_9, n_{10}\}$.

Take n_3 from \mathcal{N}_h . Add a node v_4 to G and add link l_{n_3} between v_4 and $v_{l_{n_3}}$, which is namely v_1 , as shown in Figure 5.5 (d). Now $\mathcal{N}_h = \{n_4, n_5, n_6, n_8, n_9, n_{10}\}$.

Take n_4 from \mathcal{N}_h . Add a node v_5 to G and add link l_{n_4} between v_5 and $v_{l_{n_4}}$, which is also v_1 , as shown in Figure 5.5 (e). Now $\mathcal{N}_h = \{n_5, n_6, n_8, n_9, n_{10}\}$. We have $\mathcal{N}_b(n_4) = \{n_1, n_2, n_3, n_8, n_{11}\}$. Since $n_8 \in \mathcal{N}_h$ and $v_{l_{n_8}} = v_3 \neq v_{l_{n_4}} = v_1$, add l_{n_8} between v_5 and v_3 to G . Now $\mathcal{N}_h = \{n_5, n_6, n_9, n_{10}\}$. Since $n_{11} \in \mathcal{N}_w$, set $v_{l_{n_{11}}}$ to v_5 . Now $\mathcal{N}_w = \emptyset$ and $\mathcal{N}_h = \{n_5, n_6, n_9, n_{10}, n_{11}\}$.

Take n_5 from \mathcal{N}_h . Add a node v_6 to G and add link l_{n_5} between v_6 and $v_{l_{n_5}}$, which is also v_2 , as shown in Figure 5.5 (f). Now $\mathcal{N}_h = \{n_6, n_9, n_{10}, n_{11}\}$.

Take n_6 from \mathcal{N}_h . Add a node v_7 to G and add link l_{n_6} between v_7 and $v_{l_{n_6}}$, which is also v_2 , as shown in Figure 5.5 (g). Now $\mathcal{N}_h = \{n_9, n_{10}, n_{11}\}$. We have $\mathcal{N}_b(n_6) = \{n_1, n_5, n_7, n_{10}\}$. Since $n_{10} \in \mathcal{N}_h$ and $v_{l_{n_{10}}} = v_3 \neq v_{l_{n_6}} = v_2$, add $l_{n_{10}}$ between v_7 and v_3 to G . Now $\mathcal{N}_h = \{n_9, n_{11}\}$.

Take n_9 from \mathcal{N}_h . Add a node v_8 to G and add link l_{n_9} between v_8 and $v_{l_{n_9}}$, which is also v_3 , as shown in Figure 5.5 (h). Now $\mathcal{N}_h = \{n_{11}\}$.

Take the only node n_{11} from \mathcal{N}_h . Add a node v_9 to G and add link $l_{n_{11}}$ between v_9 and $v_{l_{n_{11}}}$, which is also v_5 , as shown in Figure 5.5 (i). Now $\mathcal{N}_h = \emptyset$. Since \mathcal{N}_w is also an empty set, the construction of G is accomplished.

5.7 Evaluation

We compare ILIGRA's running time with the running times of three published line graph reconstruction algorithms: Roussopoulos' algorithm [49], the Lehot's algorithm [50] and the Degiorgi and Simon's algorithm [55]. All algorithms have been implemented⁴ in the same programming language (C++) and the same data structures and libraries [82] have been used⁵. The evaluation of all the algorithms has been conducted on the same machine⁶.

The performances of the above-mentioned algorithms have been compared using the same input graphs H . All the algorithms are able to construct the root graph G if the given graph H is a line graph, and can tell *non-line graph* when H is not a line graph. In the evaluation, both line graphs and non-line graphs have been used as inputs. The line graphs with link density⁷ $p_H = 0.05$ and 0.125 are generated by converting random graphs [5][65] with a fixed link density into line graphs. However, the line graphs of these random graphs can never have high link densities (explained with details in Appendix 5.7.1). Therefore, the line graphs with $p_H = 0.5$ and 0.65 have been generated by converting the scale-free graphs [16] into line graphs. The non-line graphs used in simulations are just random graphs with link density p_H . In addition, we have conducted simulations for scale-free graphs that are not line graphs and the running times are comparable as the presented ones of the random graphs.

Line graph inputs. Figure 5.6 reflects the trends for the running times of all the algorithms when the input graphs are line graphs with different link density p_H and different number of links L_H . Figure 5.6 (a) and Figure 5.6 (b) show the running times for line graphs with small link density $p_H = 0.05, 0.125$, where ILIGRA performs faster than all the other algorithms. Figures 5.6 (c), (d), (e) and (f) illustrate the trends for the algorithms' running times for line graphs with high link density $p_H = 0.50$ and $p_H = 0.65$. ILIGRA is the fastest algorithm for line graphs with small number of links $200 \leq L_H \leq 500$ and Lehot's algorithm has the shortest running time for line graphs with high number of links $650 \leq L_H \leq 18000$.

Non-line graph inputs. Figure 5.7 shows the trends of the running times of all the algorithms when the input graphs H are not line graphs, with link density $p_H = 0.05, 0.125, 0.5, 0.65$ and $200 \leq L_H \leq 18000$. The running times indicate how quickly an algorithm can tell that the input graph is not a line graph. ILIGRA is the fastest algorithm in detecting non-line graphs when $p_H = 0.5, 0.65$ and $200 \leq L_H \leq 1000$, as shown in Figures 5.7 (a) and (b).

⁴The implementations are available on the authors' web page:
<http://www.nas.ewi.tudelft.nl/people/Stojan/code/ILIGRA.zip>

⁵LEDA: <http://www.algorithmic-solutions.com/leda/>

⁶Intel(R) Core(TM) 2 Duo CPU T9600 on 2 x 2.80GHz; 4GB RAM memory

⁷The *link density* of a given line graph $H(L, L_H)$ is defined by $p_H = L_H / \binom{L}{2}$, where L is the number of nodes in H and L_H is the number of links in H .

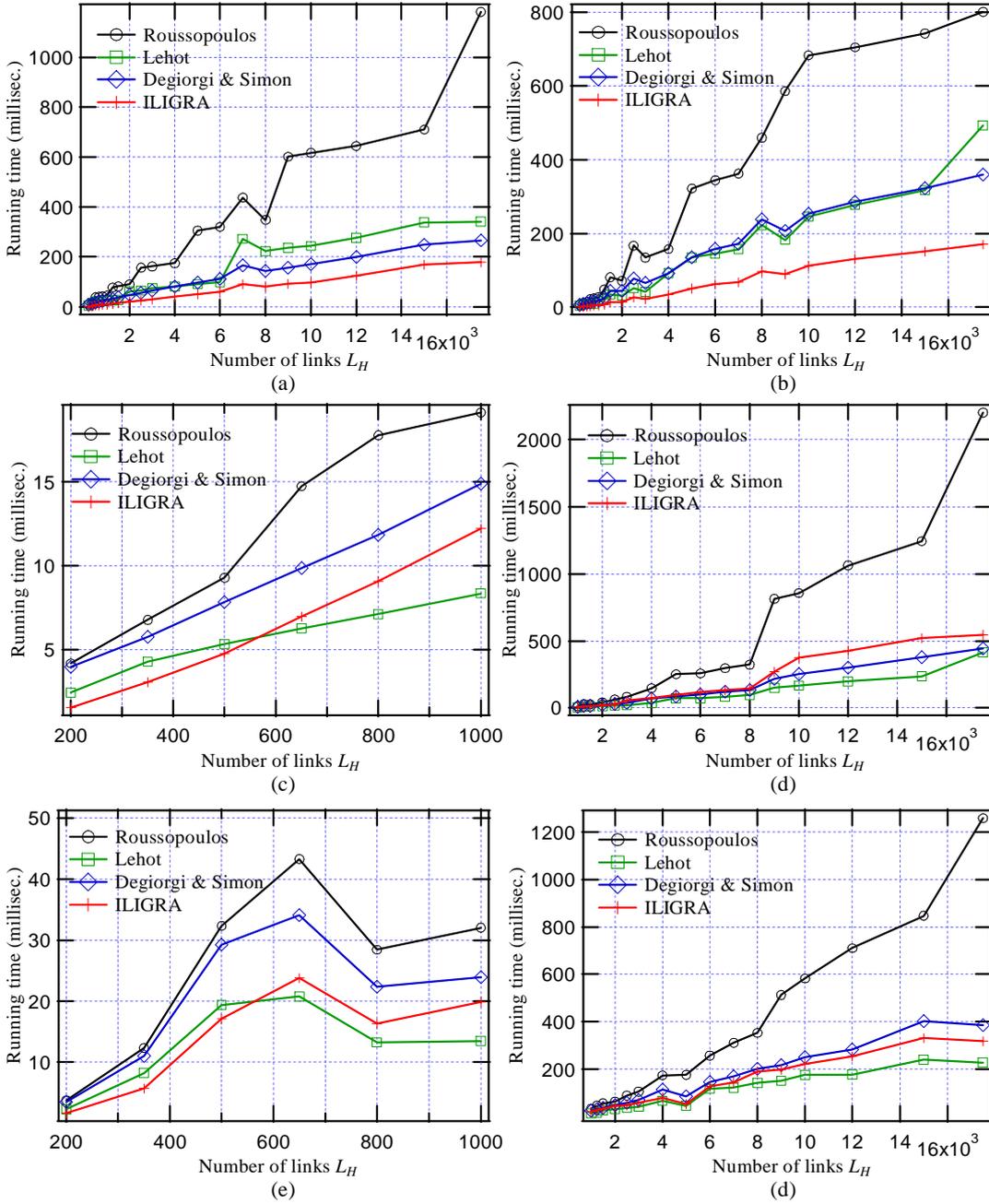


Figure 5.6: Algorithms' running times for **line graphs** with different p_H and L_H . (a) $p_H = 5\%$ and $200 \leq L_H \leq 18000$. (b) $p_H = 12.5\%$ and $200 \leq L_H \leq 18000$. (c) $p_H = 50\%$ and $200 \leq L_H \leq 1000$. (d) $p_H = 50\%$ and $1000 \leq L_H \leq 18000$. (e) $p_H = 65\%$ and $200 \leq L_H \leq 1000$. (f) $p_H = 65\%$ and $1000 \leq L_H \leq 18000$.

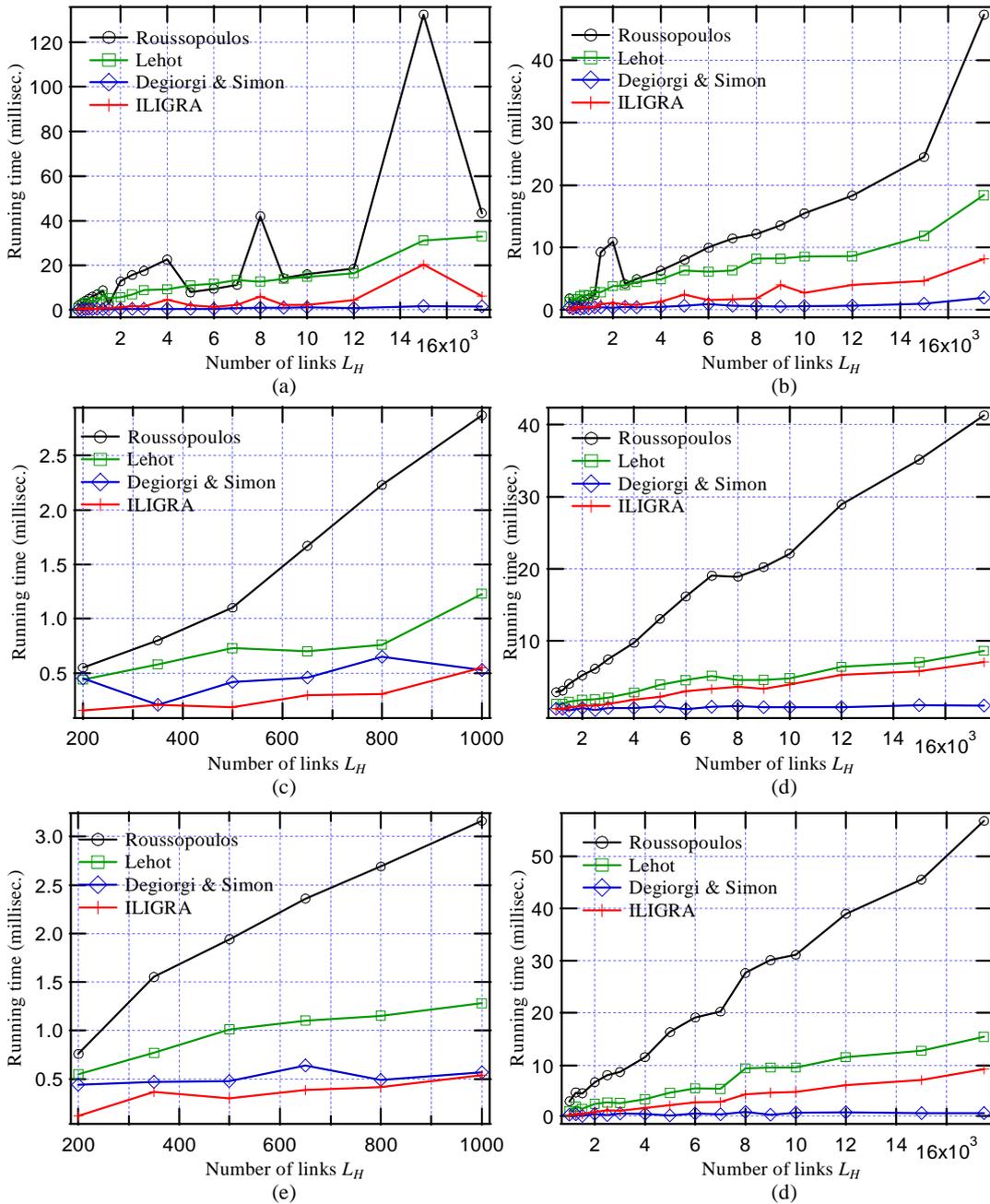


Figure 5.7: Algorithms' running times for **non-line graphs** with different p_H and L_H . (a) $p_H = 5\%$ and $200 \leq L_H \leq 18000$. (b) $p_H = 12.5\%$ and $200 \leq L_H \leq 18000$. (c) $p_H = 50\%$ and $200 \leq L_H \leq 1000$. (d) $p_H = 50\%$ and $1000 \leq L_H \leq 18000$. (e) $p_H = 65\%$ and $200 \leq L_H \leq 1000$. (f) $p_H = 65\%$ and $1000 \leq L_H \leq 18000$.

Table 5.2: The fastest algorithm for different input line graphs

graph types	line graphs		non-line graphs	
L_H \ / p_H	≤ 500	> 500	≤ 1000	> 1000
0.05	ILIGRA		Degiorgi & Simon	
0.125	ILIGRA		Degiorgi & Simon	
0.5	ILIGRA	Lehot	ILIGRA	Degiorgi & Simon
0.65	ILIGRA	Lehot	ILIGRA	Degiorgi & Simon

The best algorithms, reflected by the running time for all the cases, are summarized in Table 5.2.

5.7.1 The link density of line graphs

The link density is an important characteristic for the topology of line graphs. This section discusses the relation between the link density of line graph H and the number of nodes N and the number of links L in the root graph $G(N, L)$.

The number of nodes N_H in the line graph H is equal to the number of links L in the root graph G . For the number of links L_H in the line graph H , we have,

$$L_H = \frac{1}{2} \sum_{i=1}^N d_i^2 - L \quad (5.1)$$

where $d = [d_1, d_2, \dots, d_N]$ is the degree sequence of G .

The link density p_H of H equals

$$\begin{aligned} p_H &= \frac{L_H}{\binom{L}{2}} = \frac{\frac{1}{2} \sum_{i=1}^N d_i^2 - L}{\binom{L}{2}} \\ &= \frac{\sum_{i=1}^N d_i^2 - 2L}{L^2 - L} \end{aligned} \quad (5.2)$$

Using the basic law of degrees, $\sum_{i=1}^N d_i = 2L$, and the Cauchy inequality [83][84]

$$\begin{aligned} \sum_{i=1}^N d_i^2 &= \frac{(\sum_{i=1}^N d_i)^2}{N} + \frac{1}{2N} \sum_{i=1}^N \sum_{j=1}^N (d_i - d_j)^2 \\ &\geq \frac{(\sum_{i=1}^N d_i)^2}{N} = \frac{4L^2}{N} \end{aligned} \quad (5.3)$$

Hence,

$$\begin{aligned} p_H &= \frac{\sum_{i=1}^N d_i^2 - 2L}{L^2 - L} \\ &\geq \frac{2L(\frac{2L}{N} - 1)}{L^2 - L} = \frac{2}{N} \frac{2L - N}{L - 1} \end{aligned} \quad (5.4)$$

Equality holds for regular root graphs G , where $d_i = \frac{2L}{N}$, for $i = 1, 2, \dots, N$. When $L \gg N$, the link density p_H asymptotically tends to $\frac{4}{N}$. Hence, the line graphs of dense root graphs with $L \gg N$ have small link densities.

We derive an upper bound for the link density p_H . With $L = (\sum_{i=1}^N d_i)/2$ and using the inequality $(\sum_{i=1}^N x_i)^2 \geq \sum_{i=1}^N x_i^2$, we obtain

$$\begin{aligned} \sum_{i=1}^N d_i^2 &= \sum_{i=1}^N (d_i - 1)^2 - N + 2 \sum_{i=1}^N d_i \\ &= \sum_{i=1}^N (d_i - 1)^2 - N + 4L \\ &\leq 4L - N + \left(\sum_{i=1}^N (d_i - 1) \right)^2 \\ &= (2L - N + 1)^2 + N - 1 \end{aligned} \quad (5.5)$$

Finally, p_H is bounded by

$$\frac{4L - 2N}{N(L - 1)} \leq p_H \leq \frac{(2L - N + 1)^2 + N - 2L - 1}{L^2 - L} \quad (5.6)$$

Equality in (5.5) is achieved if and only if $(d_i - 1)(d_j - 1) = 0$ for all $i, j \in 1, 2, \dots, N$. The star graph $K_{1,N}$ satisfies the condition for equality in (5.5), indicating that the line graph of $K_{1,N}$ reaches the upper bound of link density p_H . In fact, the line graph of $K_{1,N}$ is complete graph K_{N-1} with maximum link density of 1. The graph with $L = N/2$, where the degrees of all the nodes are 1, also satisfies the condition for equality in (5.5), but its line graph is an graph with $N/2$ nodes and link density of 0. In conclusion, dense line graphs can be obtained if the original graph has one node with a high degree and the other nodes have relatively small degrees. On the other hand, the line graph of a regular graph has the minimum link density.

5.8 Chapter conclusion

We present ILIGRA algorithm for inverse line graph construction. Given a line graph H , ILIGRA constructs its root graph G and check whether the given graph is a line graph

during the construction. ILIGRA also works for disconnected line graphs and finds the connected components of input line graphs during their root graph constructions. The time complexity of ILIGRA is linear in the number of nodes in the given line graphs H without checking if the given graph is a line graph. The time complexity of ILIGRA with full functionality is linear in the number of links in the given line graphs. Numerical comparisons with the algorithms of Lehot, Roussopoulos, and Degiorgi and Simon have been demonstrated. Given line graphs with small link density (i.e. sparse graphs), ILIGRA is the fastest algorithm in root graph construction, as shown in Table 5.2. In addition to the inverse line graph construction, ILIGRA is the fastest algorithm in detecting non-line graphs with $p_H = 0.5, 0.65$ and $200 \leq L_H \leq 1000$.

Chapter 6

Random Line Graphs

For a fixed number N of nodes, the number of links L in the line graph $H(N, L)$ can only appear in consecutive intervals, called a band of L . We prove that some consecutive integers can never represent the number of links L in $H(N, L)$, and they are called a bandgap of L . We give the exact expressions of bands and bandgaps of L . We propose a model which can randomly generate simple graphs which are line graphs of other simple graphs. The essence of our model is to merge step by step a pair of nodes in cliques, which we use to construct line graphs. Obeying necessary rules to ensure that the resulting graphs are line graphs, two nodes to be merged are randomly chosen at each step. If the cliques are all of the same size, the assortativity of the line graphs in each step are close to 0, and the assortativity of the corresponding root graphs increases linearly from -1 to 0 with the steps of the nodal merging process. If we dope the constructing elements of the line graphs - the cliques of the same size - with a relatively smaller number of cliques of different size, the characteristics of the assortativity of the line graphs is completely altered. We also generate line graphs with the cliques whose sizes follow a binomial distribution. The corresponding root graphs, with binomial degree distributions, zero assortativity and semicircle eigenvalue distributions, are equivalent to Erdős-Rényi random graphs.

This chapter is organized as follows. Theoretical preliminaries for constructions line graphs are given in Section 6.1. The random line graph model is presented in Section 6.2. Section 6.3 provides insights of the topological properties of the line graphs during the merging process. We conclude in Section 6.4.

6.1 Theoretical preliminaries

6.1.1 Formation of line graphs

All the line graphs are simple graphs, but not all simple graphs are line graphs. Krausz's Theorem gives the criterion to determine whether a simple graph is a line graph. According to Krausz's Theorem [23][34][51], line graphs can be partitioned into cliques which may have nodes in common.

Theorem 12 (Krausz) *A graph is a line graph if and only if its sets of links can be partitioned into nontrivial cliques such that (i) two cliques have at most one node in common and (ii) each node belongs to at most two cliques.*

Our method to construct line graphs consists of combining separate cliques, obeying certain rules to ensure that the resulting graphs satisfy Krausz's Theorem. Before explaining the details of our method, we introduce the concept of "half-node".

Definition 13 *A half-node is the comprising part of a node and two merged half-nodes form a node. A half-node is the map of a half-link (stub) in the configuration model [66][67].*

In order to construct a graph of size N where node j has degree d_j with the configuration model [66][67], we need N separate nodes where d_j half-links (also called stubs by some authors) are incident to node j . Two combined half-links form a link. Every half-link has to be combined with another half-link. Inspired by the configuration model for the root graphs, we develop a method to construct the line graphs. We need separate cliques consisting of fully connected half-nodes, as shown in Figure 6.1 (a). A half-node is the map of a half-link in the configuration model. Two merged half-nodes form a node in the line graph. Like a node, a half-node is an abstract concept without any quantity. When two half-nodes merge into a new node, the links incident to either of the two half-nodes are attached to the new node, and the link (if any) between the two half-nodes is deleted, as shown in Figure 6.1 (a).

To construct a line graph, every half-node has to be merged with another half-node. We randomly choose and merge a pair of half-nodes, under the constraints that (1) *the two half-nodes belong to different cliques and (2) the cliques, to which the two half-nodes belong, have no nodes in common.* Once merged, two half-nodes form a node of the line graph. The construction continues until all half-nodes are merged. The rules assure that the graphs constructed by merging the half-nodes of cliques satisfy the criteria in Theorem 12 and thus are line graphs.

The "elements" for construction of line graphs, which are the cliques of half-nodes, can be regarded as the atoms, hence the formation of line graphs is analogous to the formation of a molecule. The merging of two half-nodes is analogous to the formation

of the chemical bond. Interestingly, we never see more than one chemical bond between two atoms in a molecule or a chemical bond formed with a single atom, which conforms to our rules of forming line graphs. Figure 6.1 (a) depicts a line graph constructed from a clique of K_8 , a clique of K_5 , a clique of K_4 , two cliques of K_3 , and three cliques of K_2 . The root graph of the line graph (a) is shown in Figure 6.1 (b).

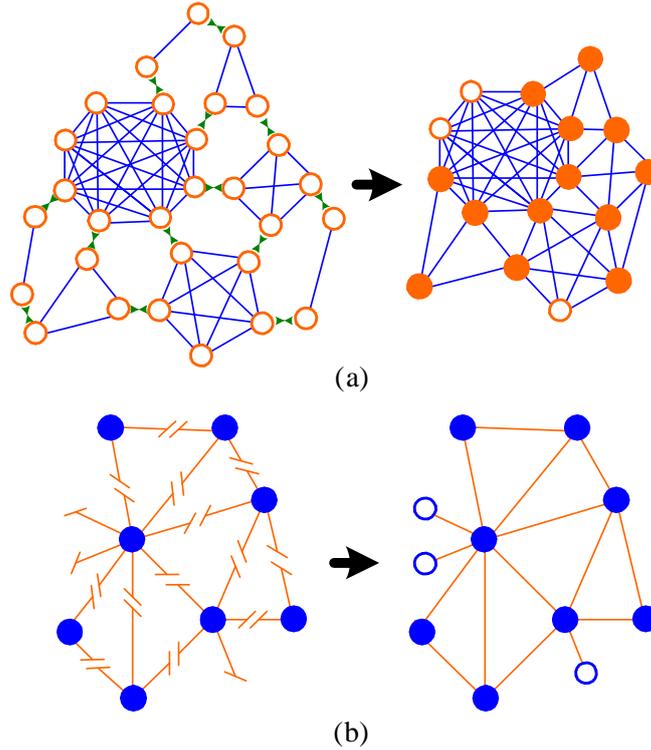


Figure 6.1: (a) The example of constructing a line graph by merging the half-nodes of cliques, and (b) the example of constructing a simple graph by the configuration model. The circles and disks denote the half-nodes and nodes respectively. The lines with slash ending and the normal lines denote the half-links and links, respectively.

6.1.2 The bandgaps of the number of links L in line graph $H(N, L)$

In this section, we investigate which integers can occur as the number of links L in the line graph $H(N, L)$.

The number of links L in the line graph $H(N, L)$ with N nodes satisfies $L \leq L_{\max} = \binom{N}{2}$, and $L = \binom{N}{2}$ only if the line graph H is a complete graph K_N . The principal clique in a line graph $H(N, L)$ is defined by the largest clique in H .

Lemma 14 Suppose that the principal clique K_{N-k+1} , where $2 \leq k \leq \lfloor \frac{N+1}{2} \rfloor$, in the line graph $H(N, L)$ consists of nodes n_k, n_{k+1}, \dots, n_N , as shown in Figure 6.2 (a). The minimum number of links in $H(N, L)$ is $L = \binom{N-k+1}{2}$, and the maximum number of links in $H(N, L)$ is $L = \binom{N-k+1}{2} + \binom{k}{2} + k - 1$.

Proof. The number of links in the principal clique K_{N-k+1} is $\binom{N-k+1}{2}$. When n_1, n_2, \dots, n_{k-1} are isolated nodes, the number of links in H is minimal and equals $L = \binom{N-k+1}{2}$. According to Theorem 12, each node of n_1, n_2, \dots, n_{k-1} belongs to at most two cliques, each of which contains a node which also belongs to the principal clique. For instance, node n_1 in Figure 6.2 (a) belongs to a clique K_2 , containing node n_{k+1} , and a clique K_k , containing node n_k . Hence, each node of n_1, n_2, \dots, n_{k-1} can have at most two links connecting itself and two nodes of the principal clique, contributing at most in total $2k - 2$ links to the line graph. There are at most $\binom{k-1}{2}$ links to fully connect the nodes n_1, n_2, \dots, n_{k-1} , thus, the maximum number of links in $H(N, L)$ is $L = \binom{N-k+1}{2} + \binom{k-1}{2} + 2k - 2 = \binom{N-k+1}{2} + \binom{k}{2} + k - 1$. ■

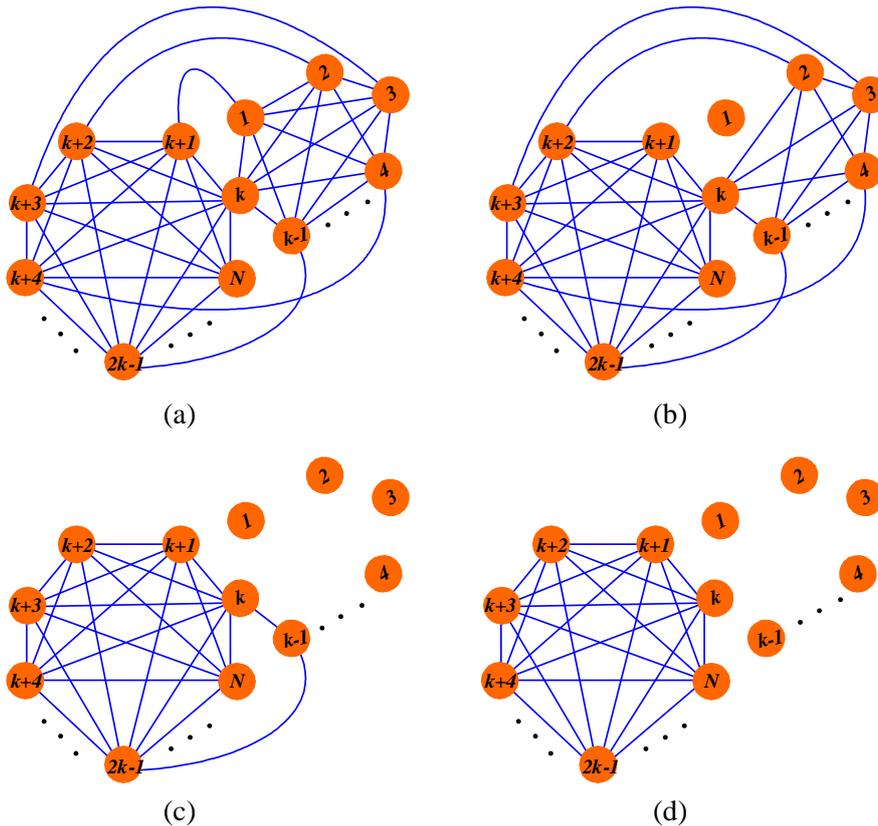


Figure 6.2: The configuration of $H(N, L)$ for (a) W_k ; (b) W_{k-1} ; (c) W_2 ; (d) W_1 .

Theorem 15 Let $V_1 = \left\{ \binom{N}{2} \right\}$. For $2 \leq k \leq \lfloor \frac{N+1}{2} \rfloor$, let $V_k = \left\{ \binom{N-k+1}{2}, \binom{N-k+1}{2} + 1, \dots, \binom{N-k+1}{2} + \binom{k}{2} + k - 1 \right\}$. Then L is the number of links in the line graph $H(N, L)$, if and only if L is a integer and

$$L \in \left(\bigcup_{k=1}^{\lfloor \frac{N+1}{2} \rfloor} V_k \right) \cup \left\{ 0, 1, 2, \dots, \binom{\lfloor \frac{N+1}{2} \rfloor}{2} + \left\lfloor \frac{N+1}{2} \right\rfloor - 1 \right\}$$

Proof. The only element $\binom{N}{2}$ in V_1 is the number of links in the line graph $H(N, L)$ when the line graph H is a complete graph K_N . Next, we prove that L is the number of links in $H(N, L)$ if L is an integer and $L \in \bigcup_{k=2}^{\lfloor \frac{N+1}{2} \rfloor} V_k$. When $H(N, L)$ contains the principal clique K_{N-k+1} and the clique K_k , sharing node k , as shown in Figure 6.2 (a), the number of links L can take the consecutive numbers in $\left\{ \binom{N-k+1}{2} + \binom{k}{2}, \binom{N-k+1}{2} + \binom{k}{2} + 1, \dots, \binom{N-k+1}{2} + \binom{k}{2} + (k-1) \right\}$, since each of the other $k-1$ nodes in K_k can be connected by a link to a node in K_{N-k+1} . Similarly, when $H(N, L)$ is constructed by two cliques K_{N-k+1} and K_{k-1} and an isolated node, the number of links L can take the consecutive numbers in $\left\{ \binom{N-k+1}{2} + \binom{k-1}{2}, \dots, \binom{N-k+1}{2} + \binom{k-1}{2} + (k-2) \right\}$, as shown in Figure 6.2 (b). In general, if $H(N, L)$ is constructed by two cliques K_{N-k+1} and K_j ($2 \leq j \leq k$), which have node k in common, and $k-j$ isolated nodes, all the integers in the set $W_j = \left\{ \binom{N-k+1}{2} + \binom{j}{2}, \dots, \binom{N-k+1}{2} + \binom{j}{2} + (j-1) \right\}$ can occur as the number of links L in the line graph $H(N, L)$. The case $j=2$ is shown in Figure 6.2 (c), while in Figure 6.2 (d), there is only a clique of K_{N-k+1} and $k-1$ isolated nodes in $H(N, L)$, the number of links can be only $L = \binom{N-k+1}{2}$. We define $W_1 = \left\{ \binom{N-k+1}{2} \right\}$. For $3 \leq j \leq k$, the smallest element of W_j , $\binom{N-k+1}{2} + \binom{j}{2}$, equals the largest element of W_{j-1} plus 1, $\binom{N-k+1}{2} + \binom{j-1}{2} + (j-2) + 1$,

$$\begin{aligned} & \binom{N-k+1}{2} + \binom{j-1}{2} + (j-2) + 1 \\ &= \binom{N-k+1}{2} + \binom{j}{2} \end{aligned}$$

The smallest element of W_2 equals the element of W_1 plus 1, $\binom{N-k+1}{2} + 1$,

$$\binom{N-k+1}{2} + 1 = \binom{N-k+1}{2} + \binom{2}{2}$$

Hence,

$$\begin{aligned} V_k &= \left\{ \binom{N-k+1}{2}, \dots, \binom{N-k+1}{2} + \binom{k}{2} + (k-1) \right\} \\ &= \bigcup_{j=1}^k W_j \end{aligned}$$

where $2 \leq k \leq \lfloor \frac{N+1}{2} \rfloor$. Thus, all the integers in the sets $\bigcup_{k=2}^{\lfloor \frac{N+1}{2} \rfloor} V_k$ can occur as the number of links L .

Lemma 14 states that, for each k between 2 and $\lfloor \frac{N+1}{2} \rfloor$ (the size of the principal clique is $N-k+1$), the set V_k covers the minimum and maximum number of links in $H(N, L)$. Hence, all the integers in the intervals $\Psi_k = \{ \binom{N-k}{2} + \binom{k+1}{2} + k + 1, \dots, \binom{N-k+1}{2} - 1 \}$, which are the gaps between V_{k+1} and V_k , $1 \leq k \leq \lfloor \frac{N+1}{2} \rfloor - 1$, cannot occur as the number of links L in $H(N, L)$.

Next, we prove that all the integers in the set $\{0, 1, 2, \dots, (\lfloor \frac{N+1}{2} \rfloor) + \lfloor \frac{N+1}{2} \rfloor - 1\}$ can occur as the number of links L . Taking $k = \lfloor \frac{N+1}{2} \rfloor$, we employ the same method which is used to prove the integers in V_k can occur as L , except deleting all the links in the principal clique K_{N-k+1} . For $2 \leq j \leq \lfloor \frac{N+1}{2} \rfloor$, suppose that $H(N, L)$ is constructed by a clique K_j consisting of nodes $n_k, n_{k-1}, \dots, n_{k-j+1}$, isolated nodes n_1, n_2, \dots, n_{k-j} , and the set of nodes $n_{k+1}, n_{k+2}, \dots, n_N$, among which any pair of nodes are not adjacent. The number of links L can take any integer in $\{ \binom{j}{2}, \dots, \binom{j}{2} + j - 1 \}$, since each of nodes $n_k, n_{k-1}, \dots, n_{k-j+1}$ can be connected by a link to a node in $\{n_{k+1}, n_{k+2}, \dots, n_N\}$, where $\binom{a}{b} = 0$ if $a, b \in \mathbb{N}$ and $a < b$. We further have $\{0, 1, 2, \dots, (\lfloor \frac{N+1}{2} \rfloor) + \lfloor \frac{N+1}{2} \rfloor - 1\} = \bigcup_{j=1}^{\lfloor \frac{N+1}{2} \rfloor} \{ \binom{j}{2}, \dots, \binom{j}{2} + j - 1 \}$. Hence, each integer in $\{0, 1, 2, \dots, (\lfloor \frac{N+1}{2} \rfloor) + \lfloor \frac{N+1}{2} \rfloor - 1\}$ can occur as the number of links L .

If N is odd, $(\lfloor \frac{N+1}{2} \rfloor) + \lfloor \frac{N+1}{2} \rfloor - 1 = \binom{\frac{N+1}{2}}{2} + \frac{N+1}{2} - 1$ and the smallest element of $V_{\lfloor \frac{N+1}{2} \rfloor}$, $\binom{N - \lfloor \frac{N+1}{2} \rfloor + 1}{2} = \binom{\frac{N+1}{2} + 1}{2} = \binom{\frac{N+1}{2}}{2} + \frac{N+1}{2}$. If N is even, $(\lfloor \frac{N+1}{2} \rfloor) + \lfloor \frac{N+1}{2} \rfloor - 1 = \binom{\frac{N}{2}}{2} + \frac{N}{2} - 1$, and the smallest element of $V_{\lfloor \frac{N+1}{2} \rfloor}$, $\binom{N - \lfloor \frac{N+1}{2} \rfloor + 1}{2} = \binom{\frac{N}{2} + 1}{2} = \binom{\frac{N}{2}}{2} + \frac{N}{2}$. Hence, there is no gap between the set $\{0, 1, 2, \dots, (\lfloor \frac{N+1}{2} \rfloor) + \lfloor \frac{N+1}{2} \rfloor - 1\}$ and $V_{\lfloor \frac{N+1}{2} \rfloor}$.

We have proven that (i) all the integers in $\bigcup_{k=1}^{\lfloor \frac{N+1}{2} \rfloor} V_k$ can occur as L , and (ii) all the integers in $\{0, 1, 2, \dots, (\lfloor \frac{N+1}{2} \rfloor) + \lfloor \frac{N+1}{2} \rfloor - 1\}$ can occur as L , and (iii) all the natural numbers in the gaps between $V_{\lfloor \frac{N+1}{2} \rfloor}, V_{\lfloor \frac{N+1}{2} \rfloor + 1}, \dots, V_1$, cannot occur as L , and

(iv) there is no gap between $\left\{0, 1, 2, \dots, \binom{\lfloor \frac{N+1}{2} \rfloor} + \lfloor \frac{N+1}{2} \rfloor - 1\right\}$ and $V_{\lfloor \frac{N+1}{2} \rfloor}$. All these conclusions together prove Theorem 15. ■

Corollary 16 *If $\left\lfloor \frac{-3+\sqrt{17+8N}}{2} \right\rfloor \leq k \leq \lfloor \frac{N+1}{2} \rfloor$, there is no gap between V_k and V_{k-1} .*

Proof. When the largest element of V_k plus 1 is not smaller than the smallest element of V_{k-1} , there is no gap between V_k and V_{k-1} .

$$\binom{N-k+1}{2} + \binom{k}{2} + k - 1 + 1 \geq \binom{N-(k-1)+1}{2}$$

which is equivalent to

$$k^2 + 3k - (2N + 2) \geq 0$$

from which Corollary 16 follows. ■

The width ΔV_k of the k th band V_k of L for the line graph $H(N, L)$, defined by the number of integers in the band, equals,

$$\begin{aligned} \Delta V_k &= \binom{N-k+1}{2} + \binom{k}{2} + k - 1 - \binom{N-k+1}{2} + 1 \\ &= \binom{k}{2} + k \end{aligned} \tag{6.1}$$

where $2 \leq k \leq \lfloor \frac{N+1}{2} \rfloor$. The k th bandgap Ψ_k of L for the line graph $H(N, L)$ is

$$\Psi_k = \left\{ \Gamma, \Gamma + 1, \Gamma + 2, \dots, \binom{N-k+1}{2} - 1 \right\}$$

where $\Gamma = \binom{N-k}{2} + \binom{k+1}{2} + k + 1$ and $1 \leq k \leq \lfloor \frac{N+1}{2} \rfloor - 1$. The width $\Delta \Psi_k$ of the k th bandgap of L is defined by the number of integers in the bandgap,

$$\begin{aligned} \Delta \Psi_k &= \binom{N-k+1}{2} - 1 - \left(\binom{N-k}{2} + \binom{k+1}{2} + k \right) \\ &= N - \frac{k^2 + 5k}{2} - 1 \end{aligned} \tag{6.2}$$

When $\Delta \Psi_k = N - \frac{k^2 + 5k}{2} - 1 < 1$, or equivalently when $1 \leq k \leq \lfloor (\sqrt{9 + 8N} - 5) / 2 \rfloor$, the k th bandgap of L vanishes. Figure 6.3 shows that there are no bandgaps when $N \leq 4$. We also observe that, for those N making $(\sqrt{9 + 8N} - 5) / 2$ an integer, the width of the bandgap $\Psi_{\lfloor (\sqrt{9+8N}-5)/2 \rfloor}$ is 1. As shown in Figure 6.3, when $N = 5, 9, 14$, we have $(\sqrt{9 + 8N} - 5) / 2 = 1, 2, 3$, and the width of the last bandgap is 1. The line graphs, whose number L of links falls into the band gaps, do not exist. If we define the energy of a line graph by the number of links in that line graph, the bands and the bandgaps of L can be regarded as energy bands and energy bandgaps of the line graph.

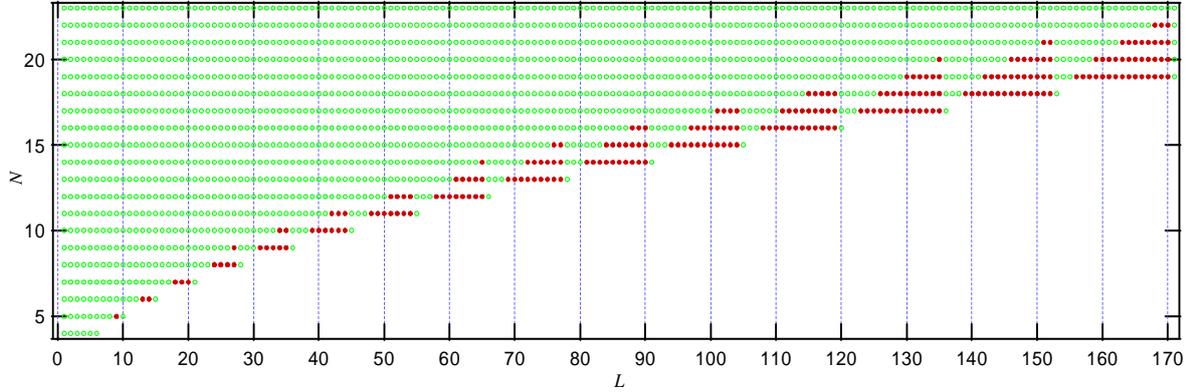


Figure 6.3: The bandgaps of L for $N = 4, 5, \dots, 14$. The solid (red) dots denote the forbidden integers, while the hollow (green) circles denote the possible integers as the number of links L .

6.2 A random line graph model

Based on the theory introduced in Section 6.1, we propose a model which generates random line graphs. In the description of the model, we do not distinguish half-nodes and nodes. The model starts with separate cliques and merges two randomly selected nodes at each step. The merging of two nodes j_1 and j_2 in $H(N, L)$ is defined by deleting node j_2 and the link connecting nodes j_1 and j_2 , and attaching the links, which are only incident to j_2 , to j_1 . The model is presented in Algorithm 6.4. Theorem 17 guarantees that the graphs constructed by Algorithm 6.4 are line graphs. In Theorem 17, l_{j_1, j_2} denotes the length of the shortest path between node j_1 and node j_2 .

Theorem 17 *The line graph H consisting of separate cliques remains a line graph after the merging of any pair of nodes j_1 and j_2 satisfying $l_{j_1, j_2} > 2$.*

Proof. The randomly chosen nodes j_1 and j_2 do not belong to the same clique, otherwise $l_{j_1, j_2} = 1$, contradicting with the fact $l_{j_1, j_2} > 2$. The two cliques, to which j_1 and j_2 belong respectively, do not share a node, otherwise there would be a hop $j_1 \sim j_0 \sim j_2$, where j_0 is the node shared by them, and thus $l_{j_1, j_2} = 2$, which contradicts with $l_{j_1, j_2} > 2$. Therefore, the nodes j_1 and j_2 are from two different cliques which have no nodes in common. After merging of nodes j_1 and j_2 , the graph H satisfies Theorem 12, hence, it remains a line graph. ■

A sequence of integers $s = [s_1 \ s_2 \ \dots \ s_C]^T$ are designated as the sizes of the cliques (line 1). All the integers are larger than one, $s_j \geq 2, j = 1, 2, \dots, C$. These numbers are actually the degrees of the nodes in the root graph, that correspond to the

```

 $H \leftarrow \text{RANDOMLINEGRAPH}(s)$ 
1 Construct a graph  $H$  consisting of the separate
  cliques whose sizes are the integers in the vector
   $s = [s_1 \ s_2 \ \cdots \ s_C]^T$ , and all the integers are larger than 1.
2  $\mathcal{N}_m \leftarrow$  the set of nodes in  $H$ 
3 repeat
4 randomly choose two nodes  $j_1, j_2$  in  $\mathcal{N}_m$ , which satisfy  $l_{j_1, j_2} > 2$ 
5 Merge  $j_1$  and  $j_2$ 
6  $\mathcal{N}_m = \mathcal{N}_m \setminus \{j_1, j_2\}$ 
7  $N = N - 1$ 
8 until no nodes  $j_1, j_2$  in  $\mathcal{N}_m$  satisfying  $l_{j_1, j_2} > 2$ 

```

Figure 6.4: Meta-code for RANDOMLINEGRAPH.

cliques in the line graph. A graph $H(N, L)$ consisting of the separate cliques whose sizes are the given sequence of numbers is constructed (line 1). The number of nodes N equals $\sum_{j=1}^C s_j$, and the number of links L equals $\sum_{j=1}^C \binom{s_j}{2}$. Initially each of the nodes in H belongs to only one clique, and hence, are expansive nodes. The set of all expansive nodes in H is denoted by \mathcal{N}_m , which before the first merging is the set of nodes in H (line 2). Two nodes j_1 and j_2 are uniformly¹ chosen in \mathcal{N}_m , between which the shortest path length $l_{j_1, j_2} > 2$ (line 4). Nodes j_1 and j_2 are merged (line 5), and removed from \mathcal{N}_m (line 6), and the number of nodes N in the line graph H decreases by 1 (line 7). Lines 4-7 are repeated until there are no nodes j_1, j_2 in \mathcal{N}_m satisfying $l_{j_1, j_2} > 2$ (lines 3 and 8). Theorem 17 ensures that H remains a line graph after each execution of lines 4-7.

Theorem 18 *The maximal number η of mergings that are performed in Algorithm 6.4 satisfies $\eta \leq \min \left(\left\lfloor \frac{1}{2} \sum_{j=1}^C s_j \right\rfloor, \binom{C}{2} \right)$.*

Proof. In a line graph, each node belongs to at most two cliques, therefore, the maximal number $\eta \leq \frac{1}{2} \sum_{j=1}^C s_j$ if $\sum_{j=1}^C s_j$ is an even number, and the maximal number $\eta \leq \frac{1}{2} \sum_{j=1}^C s_j - \frac{1}{2}$ if $\sum_{j=1}^C s_j$ is an odd number. Hence, $\eta \leq \left\lfloor \frac{1}{2} \sum_{j=1}^C s_j \right\rfloor$. In a line graph, each pair of cliques can have at most one node in common, therefore, the maximal number of mergings is also bounded by $\binom{C}{2}$. Hence, the maximal number of mergings $\eta \leq \min \left(\left\lfloor \frac{1}{2} \sum_{j=1}^C s_j \right\rfloor, \binom{C}{2} \right)$. ■

¹Instead of uniformly at random choosing two nodes in the set \mathcal{N}_m , we can also choose them in another random way, such as choosing two nodes with higher probability which have similar nodal degree. Such a model can be called preferential merging line graph model.

6.3 The assortativity of (H, G) during the merging process

In the susceptible-infectious-susceptible (SIS) model [85][86] for network epidemics, the network is infected in the steady-state if the effective infection rate τ is larger than the epidemic threshold τ_c , and the network is virus free in the steady-state when $\tau < \tau_c$. By the N-Intertwined mean-field approximation (NIMFA) [85], the exact SIS epidemic threshold τ_c is lower bounded,

$$\tau_c \geq \tau_c^{(1)} = \frac{1}{\lambda_1(A)}$$

where $\lambda_1(A)$ is the largest eigenvalue of the adjacency matrix A of a network and is often called the spectral radius of the network. When the lower bound $\tau_c^{(1)}$ for the epidemic threshold is increased in a network, we are always sure that the real epidemic threshold (which is in most cases difficult to compute) is on the safe side. The largest eigenvalue $\lambda_1(A)$ also plays an important role in the phase-transition threshold of a network of coupled oscillators [87][86].

The largest eigenvalue $\lambda_1(A)$ is closely related to the assortativity coefficient ρ_D : $\lambda_1(A)$ increases with ρ_D . The minimum and maximum assortativity of a graph is computed in [88]. Several lower bounds for $\lambda_1(A)$ are given in [58]. The assortativity coefficient ρ_D can be increased or decreased by the degree-preserving rewiring [58]. However, $\rho_D(t)$ as a function of the step t of rewiring is unknown. Apart from altering the epidemic threshold by changing the graph's assortativity, link and node removals are another way to modify the largest eigenvalue of a networks [86]. In this section, we show that the assortativity coefficient $\rho_D(G, t)$ of the root graph G of the line graph at the step t is a linear function of t in the nodal merging process of the random line graph model described in Algorithm 6.4.

6.3.1 Random line graphs with cliques of the same size $s_j = S$ for $j = 1, 2, \dots, C$

We construct line graphs with the random line graph model in Algorithm 6.4. We take 50 cliques of the same size S , and randomly choose and merge two nodes with shortest path larger than 2 until there are no such pair of nodes. After each step t of merging two nodes, the assortativity coefficient ρ_D of the line graph H and the corresponding root graph G are computed. The plots of $\rho_D(H, t)$ and $\rho_D(G, t)$ with $S = 2, 3, 4, 5, 6, 7$ are shown respectively in Figure 6.5. The numerical results show that the assortativity of the line graph, $\rho_D(H, t)$, is close to 0 for $S = 3, 4, 5, 6, 7$, and the assortativity of the line graph, $\rho_D(G, t)$, increases linear with t for $S = 2, 4, 5, 6, 7$. We give the analytical analysis below.

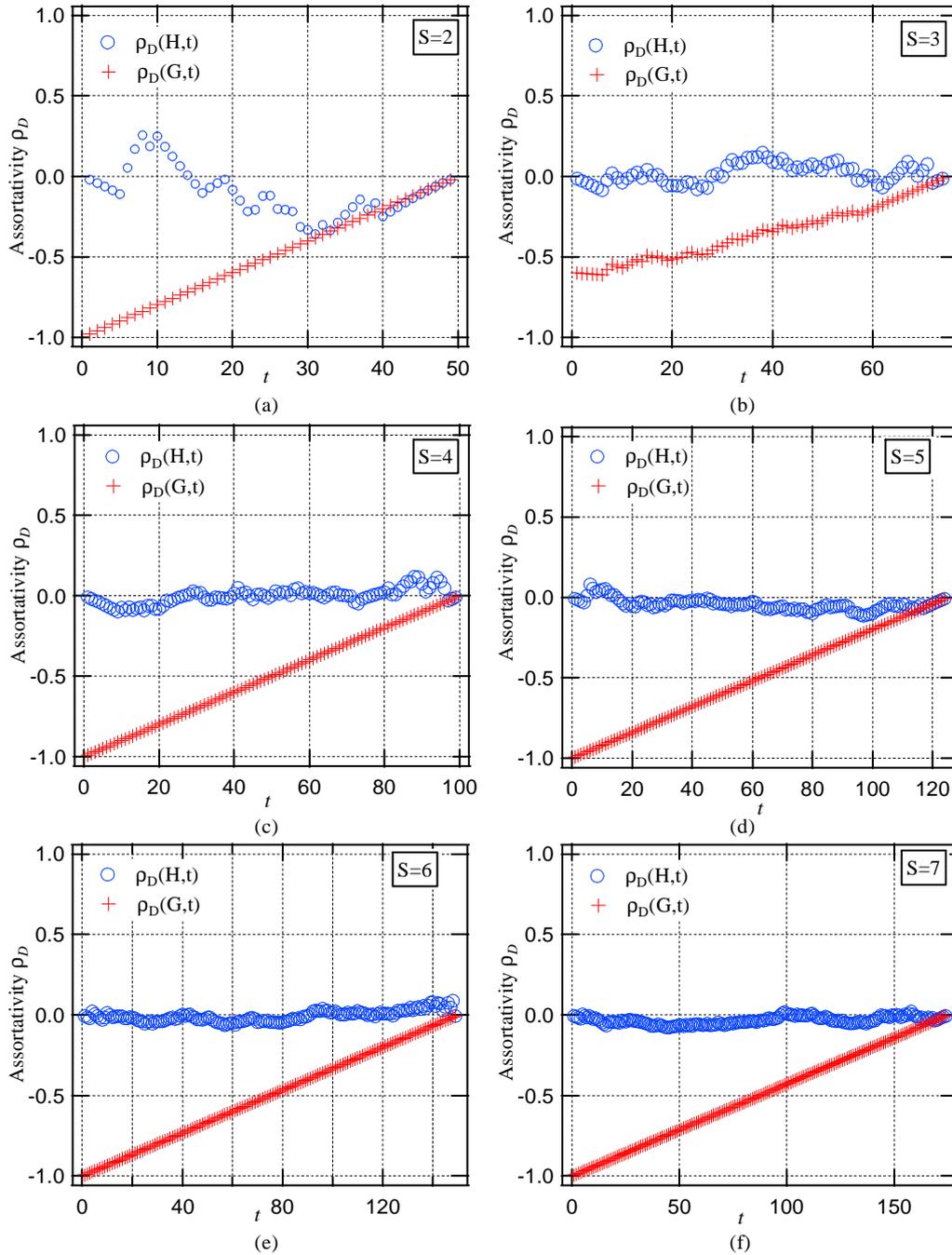


Figure 6.5: Using the line graph model, we construct line graphs with 50 cliques of the same size S . The assortativity coefficient of the line graphs and the corresponding root graphs are drawn as functions of the steps t of the nodal merging process. The root graphs of the line graphs are computed by ILIGRA, the inverse line graph construction algorithm. One can also use other algorithms [49, 50, 53] to compute the root graphs.

Assortativity of line graphs

In the random line graph model, there are initially C separate cliques of size S . Hence, $H(N, L)$ has $N = CS$ nodes with degree $d_j = S - 1$ for $j = 1, 2, \dots, N$. The number of links $L = C \binom{S}{2}$ is constant in the process of consecutive merging of two nodes. The assortativity coefficient ρ_D of a graph is expressed (see Eq. (7) in [58]) as,

$$\rho_D = 1 - \frac{\sum_{i \sim j} (d_i - d_j)^2}{\sum_{i=1}^N d_i^3 - \frac{1}{2L} (\sum_{i=1}^N d_i^2)^2} \quad (6.3)$$

where $\sum_{i \sim j}$ denotes the sum over all adjacent pairs of nodes. For simplicity, we denote the numerator by $A = \sum_{i \sim j} (d_i - d_j)^2$ and the denominator by $B = \sum_{i=1}^N d_i^3 - \frac{1}{2L} (\sum_{i=1}^N d_i^2)^2$, hence $\rho_D = 1 - \frac{A}{B}$.

When $t = 1$, we have 1 node with degree $2(S - 1)$ and $CS - 2$ nodes with degree $S - 1$. Further, when $t = 2$, we have 2 nodes with degree $2(S - 1)$ and $CS - 4$ with degree $S - 1$. After t steps of merging, there are t nodes with degree $2(S - 1)$ and $CS - 2t$ nodes with degree $S - 1$, and $N = CS - t$. The denominator B in (6.3) for $\rho_D(H, t)$ is

$$\begin{aligned} B &= \sum_{i=1}^N d_i^3 - \frac{1}{2L} \left(\sum_{i=1}^N d_i^2 \right)^2 \\ &= 8t(S - 1)^3 + (CS - 2t)(S - 1)^3 \\ &\quad - \frac{1}{CS(S - 1)} \left(4t(S - 1)^2 + (CS - 2t)(S - 1)^2 \right)^2 \\ &= \frac{(S - 1)^3 2t}{CS} (CS - 2t) \end{aligned} \quad (6.4)$$

For the numerator A in (6.3), we consider the following cases:

1. When $t \leq \frac{C}{2}$, each of the t nodes with degree $2(S - 1)$, is adjacent with on average $2(S - 1)$ nodes with degree $S - 1$. There is no degree difference among t nodes with degree $2(S - 1)$, and no degree difference among $2(S - 1)$ nodes with degree $S - 1$. Hence,

$$A = \sum_{i \sim j} (d_i - d_j)^2 \approx 2t(S - 1)^3 \quad (6.5)$$

Substituting (6.4) and (6.5) into (6.3) yields

$$\rho_D(H, t) \approx 1 - \frac{2t(S - 1)^3}{\frac{(S - 1)^3 2t}{CS} (CS - 2t)} = \frac{\frac{2t}{CS}}{\frac{2t}{CS} - 1}$$

Since $t \leq \frac{C}{2}$, the inequality $\frac{2t}{CS} \leq \frac{1}{S}$ holds. When S is large, $\rho_D(H, t)$ tends to 0.

2. When $t \approx C$, each of the t nodes with degree $2(S-1)$, is adjacent with on average $2(S-2)$ nodes with degree $(S-1)$. We have

$$A = \sum_{i \sim j} (d_i - d_j)^2 \approx 2t(S-2)(S-1)^2$$

Hence,

$$\rho_D(H, t) \approx 1 - \frac{(S-2)}{(S-1)\left(1 - \frac{2t}{CS}\right)}$$

The condition $t \approx C$ leads to $\frac{2t}{CS} \approx \frac{2}{S}$. The assortativity $\rho_D(H, t)$ is close to 0 for large S .

3. When $t \approx \frac{N}{2} = \frac{CS}{2}$, most of nodes in H have degree $2(S-1)$, therefore, $\sum_{i \sim j} (d_i - d_j)^2$ is close to 0. Since $t \approx \frac{CS}{2}$, the denominator is also close to 0, hence $\rho_D(H, t)$ is close to 0.

Results obtained in cases 1., 2. and 3. correspond to the simulation results for $\rho_D(H, t)$ in Figure 6.5. A node with degree $2(S-1)$ is adjacent with on average $2(S-1)$ nodes of degree $S-1$ when $t \leq \frac{C}{2}$, and with on average $2(S-2)$ nodes of degree $S-1$ when $t \approx C$. Hence, we deduce that a node with degree $2(S-1)$ is adjacent with on average $2\left(S - \frac{2t}{C}\right)$ nodes of degree $S-1$ after t steps of mergings. This provides another method to estimate the numerator in (6.3):

$$\begin{aligned} A &= \sum_{i \sim j} (d_i - d_j)^2 \approx t \cdot 2 \left(S - \frac{2t}{C} \right) (2(S-1) - (S-1))^2 \\ &= \frac{2t}{C} (CS - 2t) (S-1)^3 \end{aligned} \quad (6.6)$$

Hence, the assortativity of the line graph H is approximated by

$$\rho_D(H, t) \approx 1 - \frac{\frac{2t}{C} (CS - 2t) (S-1)^2}{\frac{(S-1)^{3 \cdot 2t}}{CS} (CS - 2t)} = \frac{1}{S}$$

This approximate result also agrees well with the simulations in Figure 6.5: When S increases, ρ_D becomes closer to 0.

If the selection procedure in line 4 of Algorithm 6.4 is not uniformly at random, the expression (6.4) of denominator B will be still valid, since the cliques are all of the same size S . However, the numerator A could be very different depending on how two nodes are selected at each step. The assortativity of the line graphs may not be close to 0. In case 1, $t \leq \frac{C}{2}$, and case 2, $t \approx C$, the line graphs could be very assortative or disassortative. In case 3, $t \approx \frac{N}{2} = \frac{CS}{2}$, it is still true that most of nodes in H has been merged, and most nodes have degree $2(S-1)$. Hence, we have the numerator $A = \sum_{i \sim j} (d_i - d_j)^2 \approx 0$ and the assortativity coefficient $\rho_D \approx 0$.

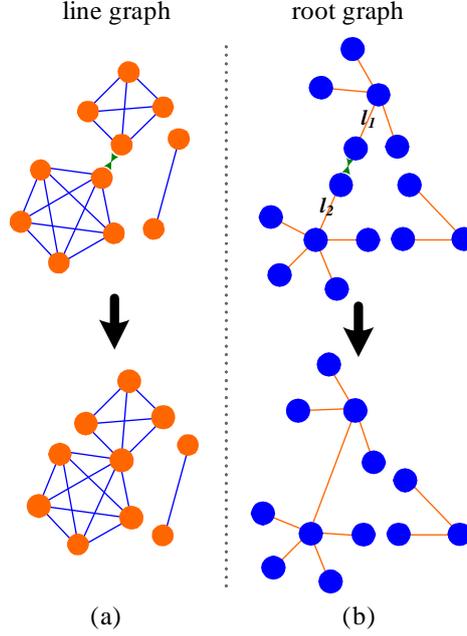


Figure 6.6: (a) The merging of two randomly selected nodes of the cliques in the line graph. (b) The corresponding root graphs before and after the nodal merging.

Assortativity of root graphs

When $t = 0$, H consists of C separate cliques with S nodes, and the corresponding root graph $G(N_G, L_G)$ consists of C separate complete bipartite graph $K_{1,S}$, which are star graphs. Hence, $\rho_D(G, t) = -1$ (see Eq.(9) in [58]). Each star graph $K_{1,S}$ has 1 node with degree S , and S nodes with degrees 1. Hence, $N_G = C(S + 1)$ and $L_G = CS$, and there are in total C nodes with degrees S , and CS nodes with degree 1. The root graph in the step t consists of interconnected star graphs (Figure 6.6), whose structure models the power law or scale-free structure of general complex networks well.

Theorem 19 *In the merging step t in the Algorithm 6.4, the assortativity coefficient of the root graph G is a linear function of t ,*

$$\rho_D(G, t) = \frac{2}{CS}t - 1 \quad (6.7)$$

where C are the number of cliques each with S nodes.

Proof. The merging of two nodes in the line graph H , corresponds to the following operations in the root graph G (as shown in the Figure. 6.6): (1) choose two links l_1 and l_2 from two different complete bipartite graphs which do not share a link; (2) delete

link l_1 , and delete the node with degree 1 which is incident to l_1 ; (3) delete the node with degree 1 which is incident to l_2 ; (4) let l_2 be incident to the node with degree S which was incident to l_1 . After these operations, the two nodes with degree S from two different complete bipartite graphs, are connected with a link. The degree of the remaining $C(S+1) - 2$ nodes keep unchanged.

After t steps of merging in the line graph, we have that the number of nodes in the root graph $N_G = C(S+1) - 2t$, and the number of links $L_G = CS - t$. There are C nodes with degree S and $CS - 2t$ nodes with degree 1 in the root graph G . The denominator B in (6.3) equals,

$$\begin{aligned} B &= CS^3 + (CS - 2t) - \frac{1}{2(CS - t)} (CS^2 + CS - 2t)^2 \\ &= \frac{CS(S-1)^2(CS - 2t)}{2(CS - t)} \end{aligned} \quad (6.8)$$

There is no degree difference among the C nodes with degree S . Each of the $(CS - 2t)$ nodes with degree 1, is adjacent with a node with degree S , therefore,

$$A = \sum_{i \sim j} (d_i - d_j)^2 = (CS - 2t)(S - 1)^2 \quad (6.9)$$

Substituting (6.8) and (6.9) into (6.3) proves Theorem 19. ■

This analytic result explains the linear increase of $\rho_D(G, t)$ with t , as shown in Figure 6.5, where the root graphs of the line graphs are computed by ILIGRA, the inverse line graph construction algorithm, although other algorithms [49, 50, 53] can also be used. Before the first merging, $t = 0$, $\rho_D(G, t) = -1$. When $t = \frac{CS}{2}$, the root graph G is a regular graph with degree S , and $\rho_D(G, t) = 0$.

The only exception from the linear law occurs when $S = 3$, of which the assortativity coefficients of the line graphs and corresponding root graphs in the nodal merging process are shown in Figure 6.5 (b). The line graphs are generated by the Algorithm 6.4. The corresponding root graphs of the line graphs are computed by ILIGRA. The root graph of K_3 can be $K_{1,3}$ or K_3 itself. The nonlinearity in Figure 6.5 (b) is originated from the fact that ILIGRA picks randomly from $K_{1,3}$ and K_3 as the root graph of line graph K_3 . If we modify ILIGRA and let it always choose $K_{1,3}$ as the root graph of line graph K_3 , the linear law (6.7) would be fulfilled in Fig 4 (b), just like the cases when $S \neq 3$. Before the line graph becomes connected in the merging process, there are always some separate cliques K_3 in the line graph. These separate cliques K_3 are translated into $K_{1,3}$ or K_3 randomly by ILIGRA, when the corresponding root graph is computed. Hence, the root graphs do not satisfy the linear law, as shown in Figure 6.5 (b). However, after the line graph becomes connected in the nodal merging process, there are no separate cliques K_3 in the line graph, hence, $\rho_D(G, t)$ increases exactly linearly with $t = 58, 59, \dots, 75$, as depicted in Figure 6.5 (b).

The linear law offers a possibility to construct graphs with a prescribed negative assortativity ρ_D by tuning different parameters. For an arbitrary small $\varepsilon > 0$, it is always possible to construct graphs with the assortativity in the interval $(-\varepsilon + \rho_D, \varepsilon + \rho_D]$. Indeed, for an arbitrary small enough ε , it is possible to take large enough C or S (one could be fixed), such that $\varepsilon CS > 1$. For such ε , C and S , taking $t = \lfloor \frac{\varepsilon}{2} CS + (1 + \rho_D) \frac{CS}{2} \rfloor$ boils down to

$$-\frac{\varepsilon}{2}CS + (1 + \rho_D) \frac{CS}{2} < t \leq \frac{\varepsilon}{2}CS + (1 + \rho_D) \frac{CS}{2} \quad (6.10)$$

as the difference of the right-hand and the left-hand sides in (6.10) is εCS . Relation (6.10) is equivalent to $-\varepsilon + \rho_D < -1 + \frac{2t}{CS} \leq \varepsilon + \rho_D$, hence we have a graph with the assortativity in the interval $(-\varepsilon + \rho_D, \varepsilon + \rho_D]$. Moreover, it is possible to find many graphs with a prescribed assortativity ρ_D : (i) by fixing the size of the clique C in one case; (ii) by fixing the number of clique S in another; or (iii) by tuning both C and S . In general, by tuning the slope $\frac{2}{CS}$, the desired negative assortativity ρ_D can be obtained.

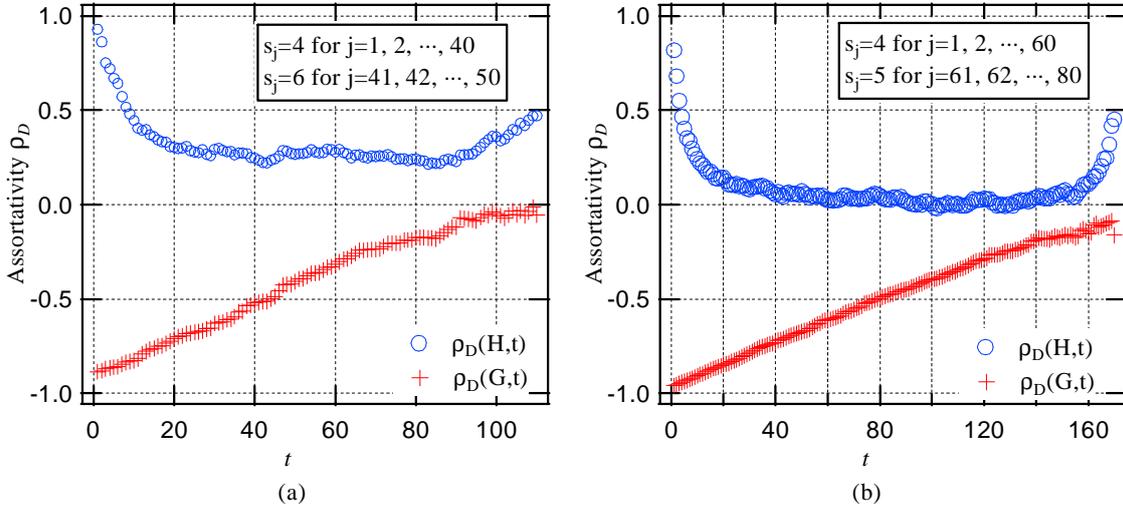


Figure 6.7: (a) The line graph H with 40 cliques of size 4 and 10 cliques of size 6, and (b) the line graph H with 60 cliques of size 4 and 20 cliques of size 5 have been constructed with Algorithm 6.4. The assortativity coefficient ρ_D of the line graphs and the corresponding root graphs in each merging step t is shown in this figure.

6.3.2 Heterogeneous random line graphs with cliques of different sizes

The characteristics of assortativity of the line graphs in Section 6.3.1 and the linear law of the assortativity presented in Theorem 19 are, however, sensitive to rather small topological changes as we exemplify in this section.

Random line graphs with cliques of two different sizes

We construct line graphs with cliques of two different sizes. The electrical properties of semiconductor materials can be manipulated by the addition of impurities, known as doping [89]. Inspired by doping in semiconductor industry, we investigate the assortativity change of the line graphs after the introducing of cliques of different size. Among all the cliques we use to construct line graphs, the majority of them are of size S_m , and the rest are of size S_d , called doping cliques. As shown in Figure 6.7 (a), for the line graph H constructed with 40 cliques of size 4 and 10 cliques of size 6, $\rho_D(H, t)$ is very high when t is small, and $\rho_D(H, t)$ ends at value close to 0.5 when the merging process finishes. During the whole merging process, $\rho_D(H, t)$ is positive, and never close to zero. In Figure 6.7 (b), the line graph H is constructed with 60 cliques of size 4 and 20 cliques of size 5. The assortativity coefficient of the line graph $\rho_D(H, t)$ first decreases rapidly from almost 1 to almost 0, and after remains close to 0 for a long range of t , $\rho_D(H, t)$ starts to increase quickly and ends at value close to 0.5. The assortativity of the line graph has been raised by adding a relatively smaller number of doping cliques to the line graph.

Random line graphs with cliques of binomial distributed size

In this section, we construct line graphs with the cliques of binomial size S . If the size of clique S follows a binomial distribution $S \sim b(N, p)$, the probability $\Pr[S = k] = \sum_{k=0}^N \binom{N}{k} p^k (1-p)^{N-k}$. In Figure 6.8 (a), the line graph H is constructed with 30 cliques where $S \sim b(20, 0.3)$ and $\sum_{j=1}^C s_j = 176$. After 88 steps of merging, H becomes a line graph of 88 nodes and 490 links, with the corresponding root graph with 30 nodes and 88 links. In Figure 6.8 (b), the line graph H constructed with 50 cliques whose size follows a binomial distribution $S \sim b(20, 0.4)$ and $\sum_{j=1}^C s_j = 327$. The line graph H has 189 nodes and 1381 links, after 188 steps of merging, and the corresponding root graph G has 51 nodes and 189 links. For the 50 cliques with size $S \sim b(20, 0.4)$, the merging process has been repeated for 1000 times, and 1000 line graphs and their root graphs were obtained. The adjacency eigenvalues of the root graphs appeared to follow a semicircle distribution, as shown in Figure 6.8 (c).

Both Figure 6.8 (a) and (b) illustrate that the assortativity of the line graph $\rho_D(H, t)$ at first drops from almost 1 to a certain level above 0, then it starts to increase and

ends at value close to 0.5. In both numerical experiments, the assortativity coefficient of the root graph $\rho_D(G, t)$ increases steadily to a value close to 0. The adjacency eigenvalues of Erdős-Rényi random graphs follow semicircle distributions [23]. The spectrum of a graph is the unique fingerprint of that graph [90]. The root graphs of the line graphs after the merging process have binomial degree distributions, and their adjacency eigenvalues follow semicircle distributions. Hence, the root graphs are believed to be equivalent to the Erdős-Rényi random graphs.

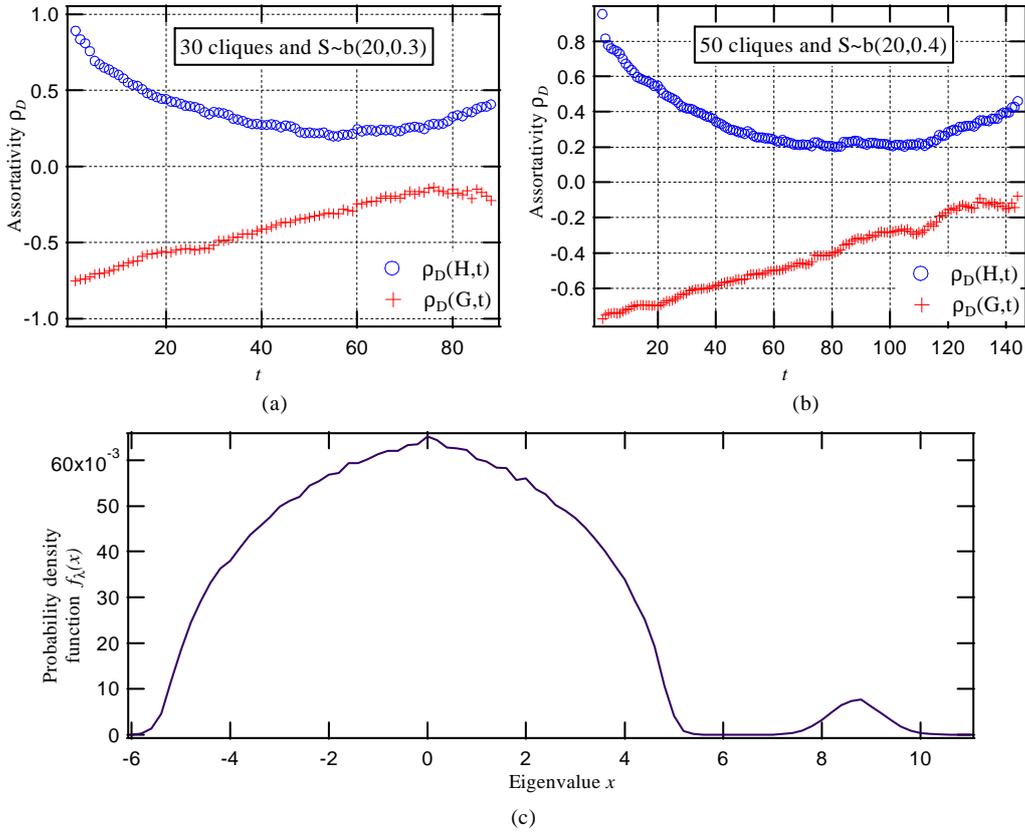


Figure 6.8: Using Algorithm 6.4, we construct (a) the line graph H with 30 cliques, the size of which follows a binomial distribution $S \sim \text{Bino}(20, 0.3)$, and (b) the line graph H with 50 cliques, the size of which follows a binomial distribution $S \sim \text{Bino}(20, 0.4)$. The assortativity coefficient ρ_D of the line graphs and the corresponding root graphs in each merging step t has been computed. (c) For the 50 cliques in (b), we repeat the merging process for 1000 times, and computed the probability density function of adjacency eigenvalues of the root graphs.

6.4 Chapter conclusion

Inspired by the configuration model [66][67] and Krausz's Theorem [34][51], we propose a model which can randomly generate simple graphs which are line graphs of other simple graphs. We show that consecutive integers can occur as the number of links L in the line graph $H(N, L)$. We also prove that there are multiple bands of consecutive integers, which can never appear as the number of links L in $H(N, L)$. The exact expressions of bands and bandgaps of L have been derived.

Our model constructs line graphs by merging step by step a pair of nodes of the cliques, which we use to construct line graphs. Obeying necessary rules to ensure that the resulting graphs are line graphs, two nodes to be merged are randomly chosen at each step. If the cliques are all of the same size, the assortativity of the line graphs are each step are close to 0, and the assortativity of the corresponding root graphs increases linearly from -1 to 0 with the steps of merging nodes. With the linear function ρ_D of the step t in Theorem 19, a graph with a prescribed negative assortativity coefficient can be constructed. The largest eigenvalue $\lambda_1(A)$ of the adjacency matrix A of a network is the only factor of the lower bound $\tau_c^{(1)}$ of the network's epidemic threshold τ_c , $\tau_c^{(1)} = \frac{1}{\lambda_1(A)} \leq \tau_c$. The largest eigenvalue $\lambda_1(A)$ can be adjusted by tuning the assortativity coefficient ρ_D . The linear law for the assortativity provides a new method to tune the assortativity besides the method of degree-preserving rewiring. If we "dope" the constructing elements of the line graphs - the cliques of the same size- with a relatively smaller number of cliques of different size, the characteristics of the assortativity of the line graphs is completely altered. We also generate line graphs with the cliques whose sizes follow a binomial distribution. The corresponding root graphs, with binomial degree distributions, zero assortativity and semicircle eigenvalue distributions, are equivalent to Erdős-Rényi random graphs.

Part III

Social Networks and Brain Networks

Chapter 7

Social Network Modeling

Social networks, as well as many other real-world networks, exhibit overlapping community structure. Affiliation networks, as a large portion of social networks, consist of cooperative individuals: Two individuals are connected by a link if they belong to the same organization(s), such as companies, research groups and hobby clubs. Affiliation networks naturally contain many fully connected communities/groups. In this chapter, we characterize the structure of the real-world affiliation networks, and propose a growing hypergraph model with preferential attachment for affiliation networks which reproduces the clique structure of affiliation networks. By comparing computational results of our model with measurements of the real-world affiliation networks of ArXiv coauthorship, IMDB actors collaboration and SourceForge collaboration, we show that our model captures the fundamental properties including the power-law distributions of group size, group degree, overlapping depth, individual degree and interest-sharing number of real-world affiliation networks, and reproduces the properties of high clustering, assortative mixing and short average path length of real-world affiliation networks.

This chapter is organized as follows: Section 7.1 introduces the hypergraph representation of affiliation networks. In Section 7.2, we present the analytical properties on the topology and spectra of the social networks. In Section 7.3, we characterize the overlapping community structure of social networks in the cases of the ArXiv coauthorship networks of subjects of "General Relativity and Quantum Cosmology" and "High Energy Physics - Theory", the IMDB movie actors collaboration network and the SourceForge software collaboration network. In Section 7.4, we propose a preferential attachment based growing hypergraph model for social networks. The nodes of the hypergraph model represent the groups of social networks, and the hyperedges, connecting multiple nodes, represent the individuals. Numerical analyses show that our hypergraph model reproduces all the properties of social networks.

7.1 The representation of social networks

7.1.1 Preliminaries

Suppose the network under consideration has N individuals and M groups, where an individual may belong to multiple groups. The membership number m_j of an individual j is defined by the number of groups of which j is a member. The degree d_j of an individual j equals the number of individuals who have the same membership in one or more groups. The interest-sharing number $\alpha_{i,j}$ of individuals i and j is defined by the number of groups to which they both belong, which indicates how many common interests they share. The group size s_k of group k is the number of individuals that belong to group k . The group degree u_k of group k equals the number of groups sharing individual(s) with group k . The overlapping depth $\beta_{k,l}$ of two groups k and l equals the number of individuals that they share. An affiliation network is linear if $\beta_{k,l} \leq 1$ for all $k, l \in [1, M]$, where M is the number of groups. If the membership number $m_j = m$ for $j \in [1, N]$, the affiliation network is called a m -uniform affiliation network.

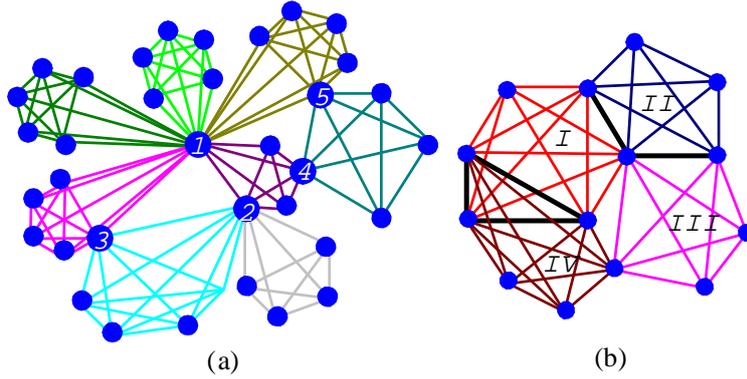


Figure 7.1: The example graph to illustrate the community structure. The nodes denote individuals. The communities overlap with each other on one or more nodes.

We use the graphs in Figure 7.1 to exemplify the definitions of d_j , m_j , $\alpha_{i,j}$, s_k , u_k , and $\beta_{k,l}$. The graph in Figure 7.1 (a) has labeled five nodes which are members of at least two groups. Obviously, $d_1 = 24$, $d_2 = 12$, $d_3 = 10$, $d_4 = 8$ and $d_5 = 9$. Nodes 1 – 5 belong to 5, 3, 2, 2 and 2 groups respectively, thus $m_1 = 5$, $m_2 = 3$ and $m_3 = m_4 = m_5 = 2$. Individual 1 and 2 belong to only one common group, hence $\alpha_{i,j} = 1$. As shown in Figure 7.1 (b), the groups $I - IV$ have 6, 5, 5 and 6 nodes respectively, hence, $s_I = s_{IV} = 6$ and $s_{II} = s_{III} = 5$. Evidently, the overlapping widths: $\beta_{I,II} = 2$, $\beta_{I,III} = 1$, $\beta_{I,IV} = 3$, $\beta_{II,III} = 2$, $\beta_{II,IV} = 0$ and $\beta_{III,IV} = 1$. The group degree: $u_I = u_{III} = 3$, $u_{II} = u_{IV} = 2$.

An affiliation network is usually described by a graph where the nodes represent the individuals and two nodes are connected by a link if they both belong to a group or several groups. If a set C_I of individuals belong to group I , the set C_I of individuals comprise a fully connected clique. If a set C_{II} ($C_{II} \subseteq C_I$) of individuals also belong to another group II , we cannot represent the group II by this graph description, because the set C_{II} of individuals are already fully connected inside the group I . Scott [91] discussed generating affiliation network with simple graphs. Newman et al. [92] suggested a bipartite graph model with all information preserved by representing a group with one type of nodes and individuals with the other type of nodes, where links only connect nodes of different types, as shown in Figure 7.2. Lattanzi and Sivakumar [93] proposed a bipartite-graph-based generative model for affiliation networks. However, the bipartite-graph-based model does not reproduce all the affiliation networks' topological properties shown in Section 7.2. Hence, we introduce the hypergraph representation of affiliation networks.

7.1.2 Hypergraph representation

A hypergraph is the generalization of a simple graph. A simple graph is an unweighted, undirected graph containing no self-loops nor multiple links between the same pair of nodes. A hypergraph $H(M, N)$ has M nodes and N hyperedges. We use the term "hyperedge" instead of "hyperlinks" in order not to make confusion with hyperlinks of WWW webs. Its nodes are of the same type as those of a simple graph, as shown in Figure 7.3 (a). The hyperedges of hypergraphs can connect multiple nodes, like hyperedge A in Figure 7.3 (a) connecting nodes I, II, \dots, V . A hypergraph is linear if each pair of hyperedges intersects in at most one node. Hypergraphs where all hyperedges connect the same number m of nodes are defined as m -uniform hypergraphs with the special case that 2-uniform hypergraphs are simple graphs. If an affiliation network is linear, the representing hypergraph is linear; if an affiliation network is m -uniform, the representing hypergraph is also m -uniform.

We propose to describe an affiliation network with M groups and N individuals by a hypergraph $H(M, N)$: M nodes represent the M groups; N hyperedges represent N individuals; and an hyperedge is incident to a node if the corresponding individual is a member of the corresponding group.

The line graph of a hypergraph $H(M, N)$ is defined as the graph $l(H)$, of which the node set is the set of the hyperedges of $H(M, N)$ and two nodes are connected by a link of weight t , when the corresponding hyperedges share t node(s). The degree d_j of an individual j , defined in subsection 7.1.1, equals the number of individuals that connect to j in the line graph $l(H)$. The line graph $l(H)$ is an unweighted graph when the corresponding hypergraph is linear; otherwise is weighted, and the weight of link $i \sim j$ equals the interest-sharing number $\alpha_{i,j}$.

Index	Names of communities	Members (individuals)
<i>I</i>	NAS-TU Delft	A, B, C, D, E, F
<i>II</i>	A research group-MIT	A, A_1, \dots, A_5
<i>III</i>	A research group-Cornell Univ.	A, A_6, \dots, A_{10}
<i>IV</i>	IEEE/ACM ToN editorial board	A, A_{11}, \dots, A_{15}
<i>V</i>	A research group-KSU	A, A_{16}, \dots, A_{20}
<i>VI</i>	A research group-Ericsson	B, B_1, \dots, B_4
<i>VII</i>	A research group-KPN	C, C_1, \dots, C_4
<i>VIII</i>	Piano club	C, C_5, \dots, C_8
<i>IX</i>	A research group-TNO	D, D_1, \dots, D_4
<i>X</i>	A rock band	D, D_5, D_6, D_7, G
<i>XI</i>	A soccer team	E, E_1, E_2, E_3, G
<i>XII</i>	Bioinformatics-TU Delft	F, F_1, \dots, F_4

Table 7.1: The names and the members of all the communities of the exemplary social network of NAS.

7.1.3 An illustrative example

In this subsection, we give an exemplary affiliation network and then represent it by a hypergraph. Table 7.1.3 describes an affiliation network based on the affiliations of members of the NAS research group (Network Architectures and Services Group at Delft University of Technology). Individuals A, B, C, D, E, F are members of NAS and the other individuals are the members of groups which overlap with the NAS group. Figure 7.2 depicts the bipartite graph representation of the NAS affiliation network with the blue circles representing the groups and the solid blue disks representing the individuals. Two nodes are linked when the corresponding individual belongs to the corresponding group.

We represent this network by the hypergraph $H(12, 53)$ shown in Figure 7.3 (a). The nodes of the hypergraph denote the groups and the individuals are denoted by the hyperedges. There are 12 groups as described in Table 7.1.3, corresponding to the 12 nodes of the hypergraph in Figure 7.3 (a), and there are 53 individuals among whom 6 NAS members with the membership number $m_A = 5$, $m_C = m_D = 3$, $m_B = m_E = m_F = 2$. If an individual belongs to multiple groups, the corresponding nodes are connected by the hyperedge specifying that individual.

Figure 7.3 (b) depicts the line graph $l(H)$ of the hypergraph $H(12, 53)$ in Figure 7.3 (a), which represents the exemplary NAS affiliation network. In the line graph $l(H)$, the individuals are denoted by nodes. The line graph $l(H)$ is unweighted since the NAS affiliation network is linear.

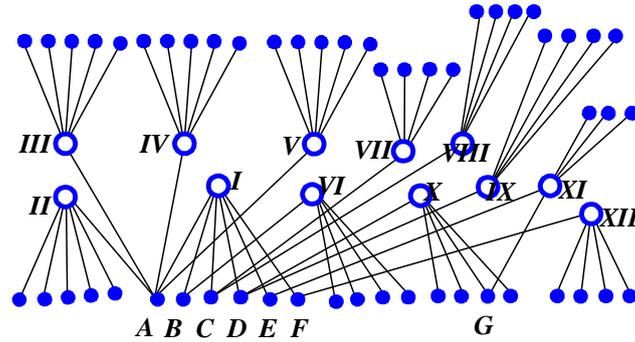


Figure 7.2: The bipartite graph representation of the network for the NAS group.

7.2 Properties of social networks

7.2.1 Topological properties

The line graph $l(H)$ has N nodes and L links. The topology of $l(H)$ can be described by its adjacency matrix A , a $N \times N$ matrix, where the element a_{ij} equals the linkweight of link $i \sim j$ if there is a link between node i and node j , else $a_{ij} = 0$. Since $l(H)$ is undirected, the adjacency matrix A is symmetric.

The following equalities are valid for all affiliation networks,

$$N = \sum_{k=1}^M s_k - \sum_{k=1, l=1}^M \beta_{k,l} \quad (7.1)$$

$$L = \frac{1}{2} \sum_{j=1}^N d_j = \sum_{k=1}^M \frac{s_k (s_k - 1)}{2} - \sum_{k=1, l=1}^M \frac{\beta_{k,l} (\beta_{k,l} - 1)}{2} \quad (7.2)$$

$$\sum_{j=1}^N (m_j - 1) = \sum_{k=1, l=1}^M \beta_{k,l} \quad (7.3)$$

If $\beta_{k,l} \leq 1$ for all $k, l \in [1, M]$, where M is the number of groups, which implies that the affiliation networks are linear, we have,

$$d_j = \sum_{\substack{\text{All the groups to} \\ \text{which individual } j \text{ belongs}}} (s - 1) \quad (7.4)$$

where s is the group size; And

$$u_k = \sum_{\substack{\text{All the individuals} \\ \text{that group } k \text{ contains}}} (m - 1) \quad (7.5)$$

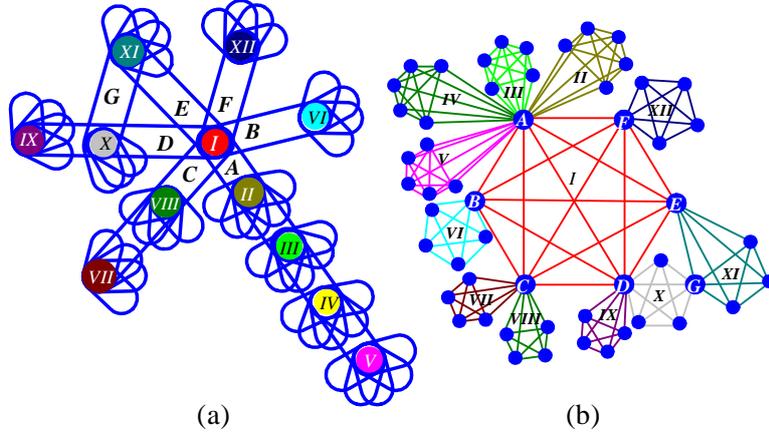


Figure 7.3: (a) The hypergraph representation of the network described in Table 7.1.3. The hyperedges are the blue ellipse-like closed curves. The nodes are the disks marked with Roman numerals. A node and a hyperedge are incident if the node is surrounded by the hyperedge. The hyperedges and nodes represent the individuals and the communities respectively. Individuals participate in multiple communities, implying that the communities overlap with each other. (b) The line graph of the hypergraph in (a), which is a simple graph. The nodes here denote the individuals. Note that this graph is also the line graph of the hypergraph.

where m is the membership number of an individual. When the affiliation network is linear, we also have $\alpha_{i,j} \leq 1$.

The adjacency matrix $A_{N \times N}^{l(H)}$ of the line graph $l(H)$ of a hypergraph $H(M, N)$ which represents an affiliation network with M groups and N individuals, can be expressed by the unsigned incidence matrices $R_{M \times N}$ of $H(M, N)$

$$A_{N \times N}^{l(H)} = (R^T R)_{N \times N} - \text{diag}(R^T R) \quad (7.6)$$

where the entry r_{ij} of R is 1 if node i and hyperedge j are incident, otherwise $r_{ij} = 0$. Basically, the adjacency matrix $A^{l(H)}$ equals the matrix $R^T R$ setting all the diagonal entries to zero. The interest-sharing number $\alpha_{i,j}$ of individual i and j equals the entry $a_{ij}^{l(H)}$ of $A^{l(H)}$

$$\alpha_{i,j} = a_{ij}^{l(H)} \quad (7.7)$$

The membership number m_j of an individual j equals,

$$m_j = \sum_{i=1}^M r_{ij} = (R^T R)_{jj} \quad (7.8)$$

The group size s_k of group k is

$$s_k = \sum_{l=1}^N r_{kl} = (RR^T)_{kk} \quad (7.9)$$

Let $W_{M \times M} = (RR^T)_{M \times M} - \text{diag}(RR^T)$, then the overlapping depth $\beta_{k,l}$ of two groups k and l equals,

$$\beta_{k,l} = w_{kl} \quad (7.10)$$

where w_{kl} is an entry of $W_{M \times M}$.

The individual degree d_j equals the number of nonzero entries in the j th row/column of $A_{N \times N}^{l(H)}$, with the special case $d_j = \sum_{i=1}^N a_{ij}^{l(H)}$ when the affiliation network is linear. Similarly, the group degree u_k equals the number of nonzero entries in the k th row/column of $W_{M \times M}$.

7.2.2 Spectral properties

The adjacency spectra of the line graph of m -uniform affiliation networks

A m -uniform affiliation network can be represented by m -uniform hypergraphs $H_m(M, N)$, of which the unsigned incidence matrix R has exactly m one-entries and $M - m$ zero-entries in each column. Thus, all the diagonal entries of $R^T R$ are m . The adjacency matrix of the line graph of $H_m(M, N)$ can be written as,

$$A_{N \times N}^{l(H_m)} = R^T R - mI \quad (7.11)$$

where $R^T R$ is a Gram matrix [34][94].

Lemma 20 For all matrices $A_{N \times M}$ and $B_{M \times N}$ with $N \geq M$, it holds that $\lambda(AB) = \lambda(BA)$ and $\lambda(AB)$ has $N - M$ extra zero eigenvalues

$$\lambda^{N-M} \det(BA - \lambda I) = \det(AB - \lambda I)$$

Lemma 20 and (7.11) yields,

$$\det\left(A_{N \times N}^{l(H_m)} - (\lambda - m)I\right) = \lambda^{N-M} \det\left((RR^T)_{M \times M} - \lambda I\right)$$

The adjacency matrix $A_{N \times N}^{l(H_m)}$ has at least $N - M$ eigenvalues $-m$. We have

$$x^T (R^T R) x = (Rx)^T Rx = \|Rx\|_2^2 \geq 0$$

and

$$x^T (RR^T) x = (R^T x)^T R^T x = \|R^T x\|_2^2 \geq 0$$

where $x_{L \times 1}$ is an arbitrary vector. Hence, both $(R^T R)_{N \times N}$ and $(RR^T)_{M \times M}$ are positive semidefinite, hence all eigenvalues of $(R^T R)_{N \times N}$ are non-negative. Due to (7.11), the adjacency eigenvalues of $A_{N \times N}^{l(H_m)}$ are not smaller than $-m$.

The adjacency spectra of the line graph of non-uniform affiliation networks

A non-uniform affiliation network with maximum membership number m_{\max} can be represented by a non-uniform hypergraph $H(M, N)$. The unsigned incidence matrix R of $H(M, N)$ has at most m_{\max} *one-entries* in each column. Therefore, the largest diagonal entry of $R^T R$ is m_{\max} . The adjacency matrix of the line graph of non-uniform hypergraph $H(M, N)$ is,

$$A_{N \times N}^{l(H)} = R^T R + C - m_{\max} I \quad (7.12)$$

where $C = \text{diag} (c_{11} \quad c_{22} \quad \cdots \quad c_{LL})$ and $c_{jj} = m_{\max} - (R^T R)_{jj} \geq 0$ for $j \in [1, N]$.

Since

$$\begin{aligned} x^T (R^T R + C) x &= x^T (R^T R) x + x^T (\sqrt{C}^T \sqrt{C}) x \\ &= \|Rx\|_2^2 + \|\sqrt{C}x\|_2^2 \geq 0 \end{aligned}$$

where $x_{L \times 1}$ is an arbitrary vector and $\sqrt{C} = \text{diag} (\sqrt{c_{11}} \quad \sqrt{c_{22}} \quad \cdots \quad \sqrt{c_{LL}})$, $R^T R + C$ is also positive semidefinite, thus, the adjacency eigenvalues of $A_{N \times N}^{l(H_m)}$ are not smaller than $-m_{\max}$.

7.3 Characterizing the real-world social networks

7.3.1 ArXiv coauthorship networks

In this section, we use the terms “community” and “group” interchangeably. We analyze the arXiv data of subjects of "General Relativity and Quantum Cosmology" (GR-QC) and "High Energy Physics - Theory" (HEP-TH) in the period from January 1993 to April 2003, which were collected by [95]. We construct the hypergraph with the papers as nodes and the authors as hyperedges. A hyperedge is incident to a node if the corresponding author authors/coauthors the corresponding paper. In this manner we construct the hypergraph of the arxiv GR-QC coauthorship network with 5855 authors and 13454 papers, and the hypergraph of the arXiv HEP-TH coauthorship network with 9877 authors and 21568 papers. We fit the data of s , β , m , d and α with the power function $f(x) = x^{-\gamma}$. The values of γ are shown in Table 7.4.2. The group size s follows a power-law distribution. In this case of coauthorship network, the group size s means the number of authors a paper has. As shown in Figure 7.4 and 7.5, We see that, in the coauthorship networks of both subjects, the papers with only one author and with more than ten authors are very rare. Most of papers have two or three authors. The group degree u in both Figure 7.4 and 7.5 has a power-law tail. The group overlapping depth β follows a power-law distribution. Most of the pairs of groups have no overlap. We only consider the group pairs which overlaps with each other. The membership number

m of an individual here means the number of papers he or she authors and coauthors. It also follows a power-law distribution. The interest-sharing number α , denoting the number of papers in which two individuals participate together, follows a power-law distribution. Only the individual pairs who have nonzero interest-sharing number are considered. The ArXiv coauthorship networks of both subjects possess high clustering coefficient, large assortativity coefficient and short average path length as shown in Table 7.4.2.

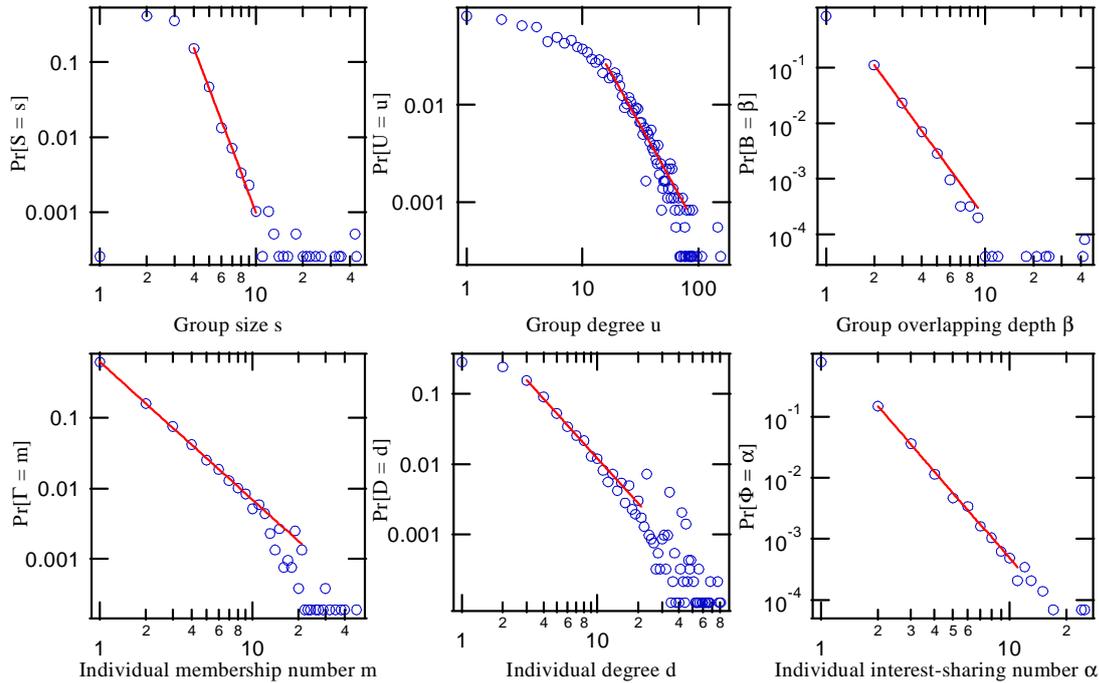


Figure 7.4: The probability density distribution of group size s , group degree u , group overlapping depth β (the first row from left to right), individual membership number m , individual degree d , individual interest-sharing number α (the second row from left to right) of ArXiv coauthorship networks of "General Relativity and Quantum Cosmology" category.

7.3.2 IMDB actor collaboration network

The data of IMDB movie actors collaboration network with 127823 movies and 392340 actors, were collected by Hawoong Heong from Internet Movie Database (based on www.imdb.com). We construct the hypergraph of IMDB movie actors collaboration network with the movies as nodes and the actors as hyperedges. A hyperedge is incident

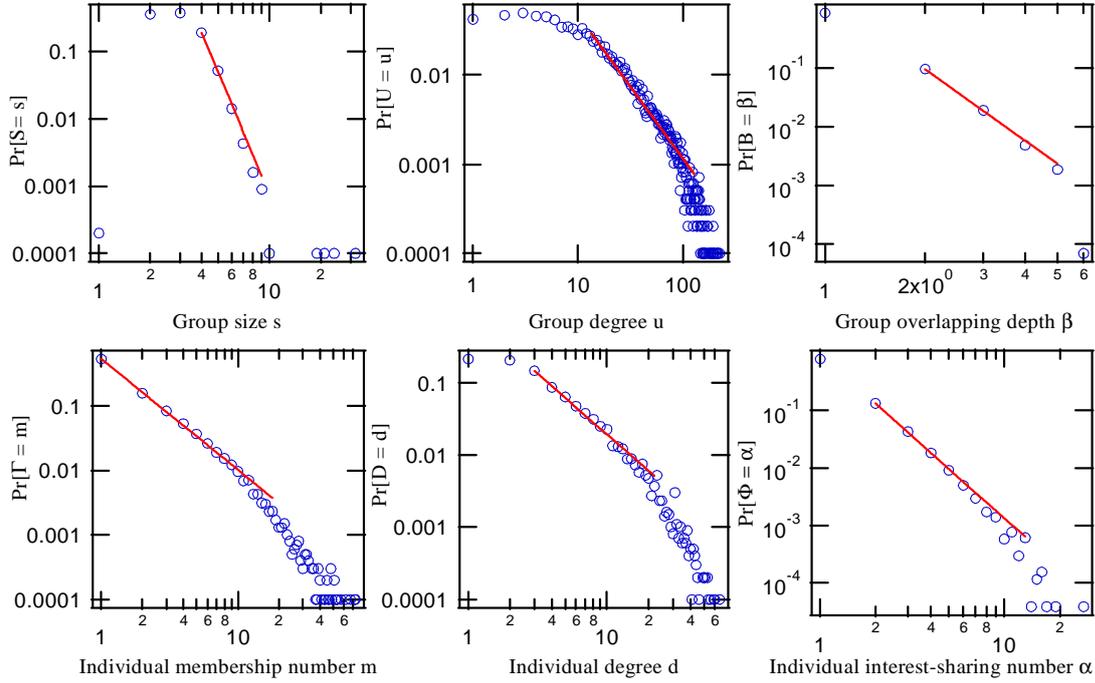


Figure 7.5: The probability density distribution of group size s , group degree u , group overlapping depth β (the first row from left to right), individual membership number m , individual degree d , individual interest-sharing number α (the second row from left to right) of ArXiv coauthorship networks of "High Energy Physics - Theory" category.

to a node if the corresponding actor appears in the corresponding movie. We fit the data of s , u , β , m , d and α with the power function $f(x) = x^{-\gamma}$, as shown in Figure 7.6 and Table 7.4.2. The data of s are fitted with two power functions in different regions. The group degree u appears also to follow two power-law distribution in two regions. All the values of γ are shown in Table 7.4.2. The IMDB movie actors collaboration network also exhibits high clustering, assortative mixing and short average path length as shown in Table 7.4.2.

7.3.3 The SourceForge software collaboration network

SourceForge is a web-based project repository assisting programmers to develop and distribute open source software projects. SourceForge facilitates developers by providing a centralized storage and tools to manage the projects. Each project has multiple developers. We construct the hypergraph of the SourceForge software collaboration network by taking software projects as nodes and the developers as hyperedges. A

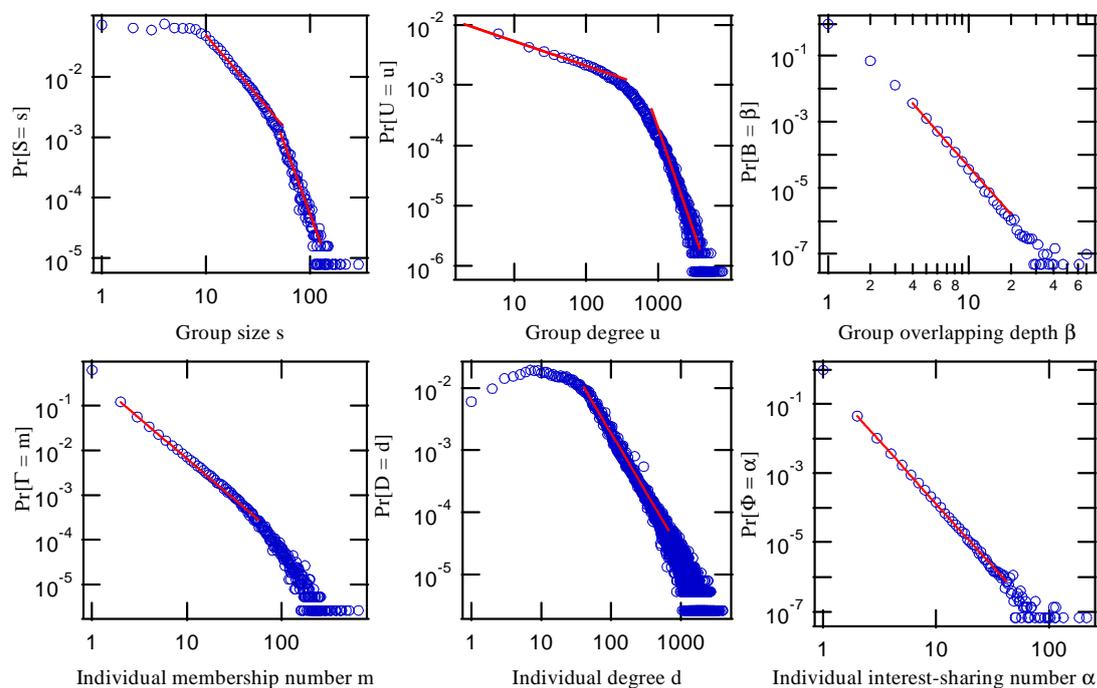


Figure 7.6: The probability density distribution of group size s , group degree u , group overlapping depth β (the first row from left to right), individual membership number m , individual degree d , individual interest-sharing number α (the second row from left to right) of IMDB movie actors collaboration networks.

hyperedge is incident to a node if the corresponding developer participates in the corresponding software project. The SourceForge software collaboration network has 259252 software projects and 161653 developers. We fit the data of s , u , β , m , d and α with the power function $f(x) = x^{-\gamma}$. As shown in Figure 7.7, the pdfs of all the six metrics d_j , m_j , $\alpha_{i,j}$, s_k , u_k , and $\beta_{k,l}$ are well fitted by power law functions with exponents γ shown in Table 7.4.2. The SourceForge network also has a high clustering coefficient, a high assortativity coefficient and an small average path length, which are shown in Table 7.4.2.

7.4 Modeling of social networks

7.4.1 Model description

In this section, we use the terms “community” and “group” interchangeably. As stated before, we use the nodes of hypergraph to represent the groups and the hyperedges

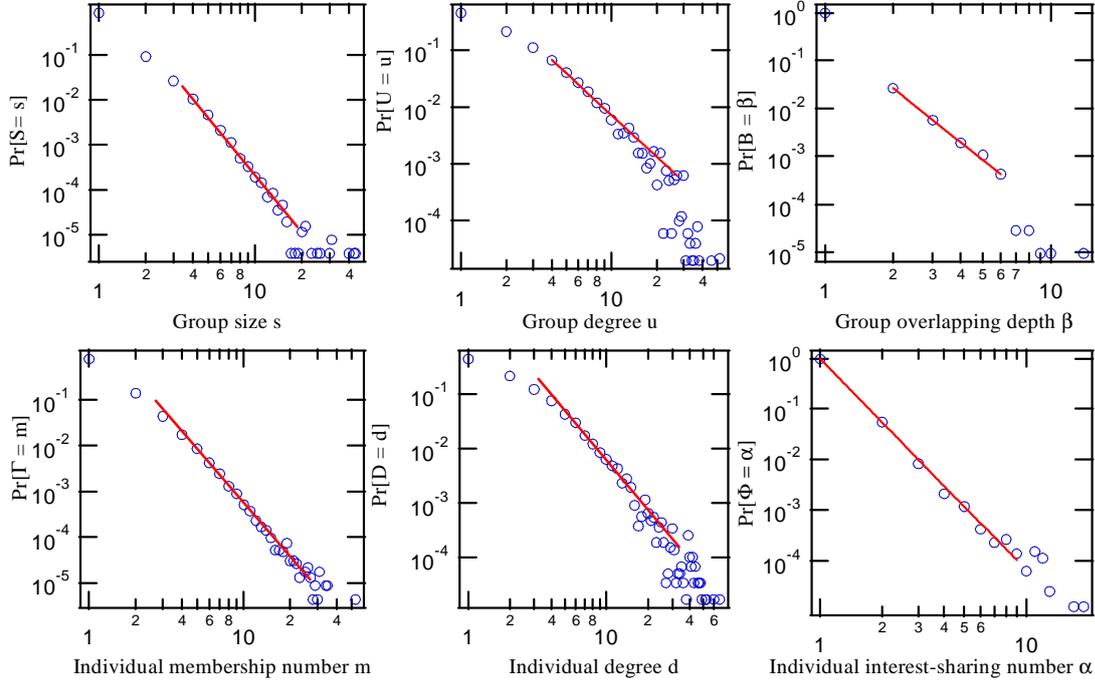


Figure 7.7: The probability density distribution of group size s , group degree u , group overlapping depth β (the first row from left to right), individual membership number m , individual degree d , individual interest-sharing number α (the second row from left to right) of the SourceForge software collaboration network.

to represent the individuals. In the description of our model, the nodes and groups, the hyperedges and individuals are used interchangeably. Our model is a growing hypergraph model, starting with a small hypergraph which represent the initial groups and individuals. Later on, new individuals and new groups are added to the network in the growing process.

We notice that the number of group M is larger than the number of individuals N in ArXiv networks and Sourceforge network, and M is smaller than N in IMDB network. Making a movie needs more efforts and labor force than writing a paper or developing an open-source software. In our model, we take $\frac{M}{N} = 1$, assuming that each coming individual start a new group. Note that the group size of real-world affiliation network follow a power-law distribution. We employ preferential attachment of individual to the existing groups to achieve the power-law distributed group size. The tricky issue is to determine the membership number of each new coming individuals, namely to decide how many nodes that a new hyperedge should connect to. The analysis of real-world affiliation networks tells a power-law distribution of the membership number,

```

 $H(N + M_0, N + N_0) \leftarrow \text{GROWINGHYPERGRAPH}(H_0(M_0, N_0), \Gamma)$ 
Input: A seed hypergraph  $H_0(M_0, N_0)$  with  $M_0$  nodes and  $N_0$  hyperedges
The membership numbers for new hyperedges  $\Gamma = [\bar{m}_1 \bar{m}_2 \cdots \bar{m}_N]$ 
Output: A hypergraph  $H(N + M_0, N + N_0)$ 
1  $H \leftarrow H_0(M_0, N_0)$ 
2 for each  $j \in \{1, 2, 3, \dots, N\}$  do
3 add a new hyperedge  $j$  to  $H$ 
4  $m_j \leftarrow 0$ 
5 add a new node to  $H$  and let it be incident to the hyperedge  $j$ 
6  $m_j \leftarrow m_j + 1$ 
7 while  $m_j < \bar{m}_j$  do
8    $k \leftarrow$  a random natural number between 1 and  $j - 1$ 
9    $r \leftarrow$  a random real number between 0 and 1
10  if  $r < s_k / \sum_{i=1}^{j-1} s_i$  then
11    let the hyperedge  $j$  be incident to the node  $k$ 
12     $m_j \leftarrow m_j + 1$ 

```

Figure 7.8: Meta-code for GROWINGHYPERGRAPH.

hence we pre-produce a power-law distributed sequence of numbers, taking them as the membership numbers of new coming individuals.

Our hypergraph model is described by the following procedure:

1. Start with a seed hypergraph $H_0(M_0, N_0)$ with M_0 groups and N_0 hyperedges.
2. Suppose that the desired number of individuals (hyperedges) of the network to be generated is $N + N_0$. Determine the membership numbers for the N new hyperedges: $\Gamma = [\bar{m}_1 \bar{m}_2 \cdots \bar{m}_N]$. Note that the membership number vector Γ is the input parameter of our hypergraph model.
3. At growing step j , $j = 1, 2, \dots, N$, add a new hyperedge j and a new group to the hypergraph. Make the new hyperedge j and the new group incident, and the membership number of j becomes 1.
 - (a) Connect the new hyperedge j to the existing group k with probability $p_k = s_k / \sum_{i=1}^{j-1} s_i$, where s_k is the group size of group k and $\sum_{i=1}^{j-1} s_i$ is the sum of group sizes of all the existing groups.
 - (b) Repeat 3a) $\bar{m}_j - 1$ times so that the membership number of the hyperedge j increases to the expected membership number \bar{m}_j .
4. Repeat 3) until the number of hyperedges increases to $N + N_0$.

The model is also presented with pseudo-codes in Algorithm 7.8. Compute the metrics d_j , m_j , $\alpha_{i,j}$, s_j, u_j and $\beta_{i,j}$ using the methods given in Section 7.2.1 including the formulas (7.6) to (7.10).

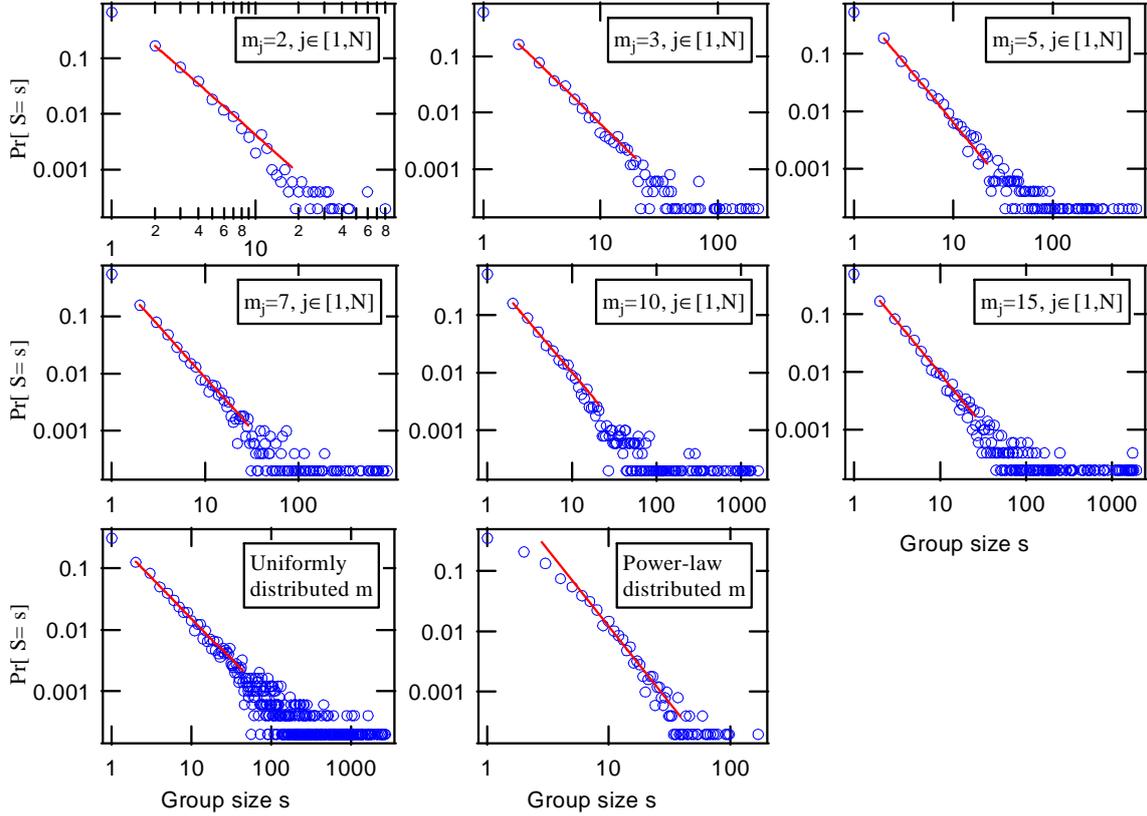


Figure 7.9: The probability density distribution of group size s for H_2 , H_3 , H_5 , H_7 , H_{10} , H_{15} , $H_{U[2,121]}$, and H_{pow} . They all have 5020 groups (nodes) and 5020 hyperedges (individuals).

7.4.2 Properties of the growing hypergraph model

Simulation settings

We use a hypergraph $H(20, 20)$ with the membership number $m_j = 1, j = 1, 2, \dots, 20$, as the starting seed. We add 5000 new hyperedges (individuals) and 5000 new nodes (groups) to the starting seed through 5000 growing steps. Hence, all the hypergraphs we generate have 5020 nodes and 5020 hyperedges.

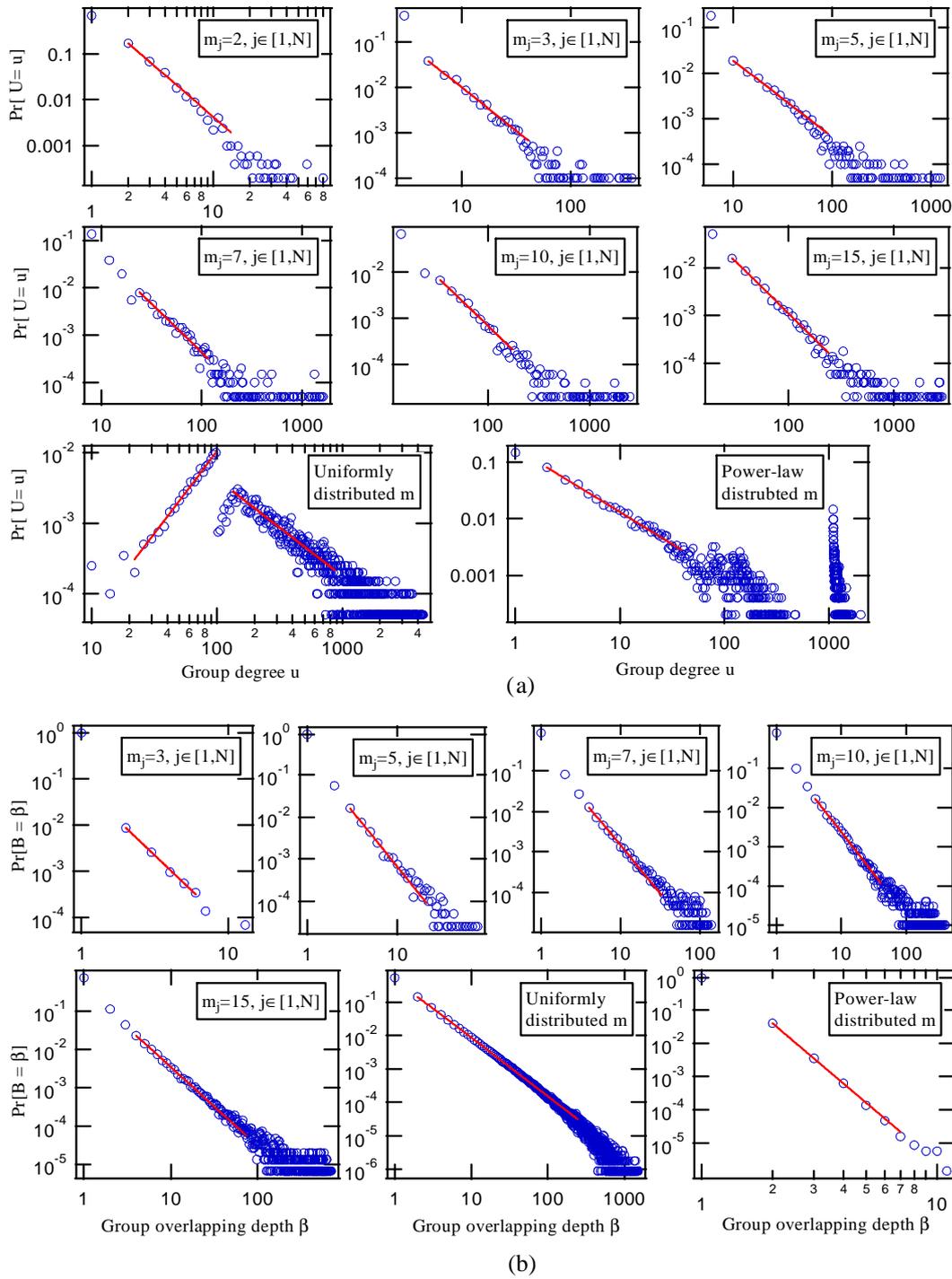


Figure 7.10: The probability density distribution of (a) group degree u for $H_2, H_3, H_5, H_7, H_{10}, H_{15}, H_{U[2,121]}$, and H_{pow} , and (b) group overlapping depth β for $H_3, H_5, H_7, H_{10}, H_{15}, H_{U[2,121]}$, and H_{pow} . They all have 5020 groups (nodes) and 5020 hyperedges (individuals).

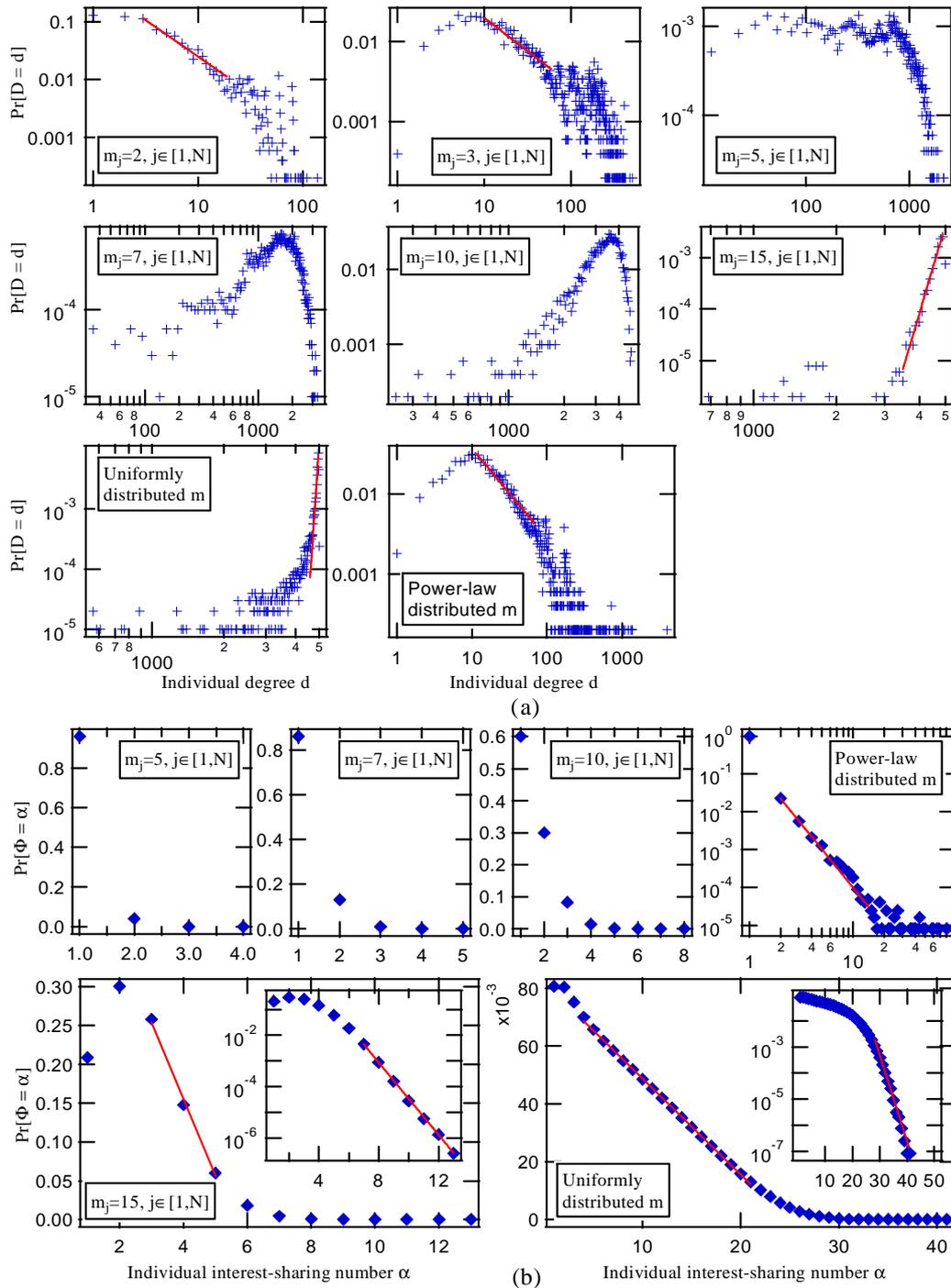


Figure 7.11: The probability density distribution of (a) individual degree d for $H_2, H_3, H_5, H_7, H_{10}, H_{15}, H_{U[2,121]}$, and H_{pow} , and (b) individual interest-sharing number α for $H_5, H_7, H_{10}, H_{15}, H_{U[2,121]}$, and H_{pow} . They all have 5020 groups (nodes) and 5020 hyperedges (individuals).

In the growing process, we first apply the constant membership number $m_j = 2$, $j = 1, 2, \dots, 5000$, obtaining the uniform hypergraph H_2 . In the same way, we construct H_3, H_5, H_7, H_{10} and H_{15} . Then we construct the hypergraph $H_{U[1,100]}$ with a uniformly distributed membership number in the interval $[1, 100]$. We construct these hypergraphs in order to study by comparison the properties of H_{pow} which is obtained by applying the sequence of membership numbers with the pdf $\Pr[\Gamma = m] = m^{-2.02}$. We construct H_{pow} in this way: generating a sequence of natural numbers following a power-law distribution with the pdf $\Pr[\Gamma = m] = m^{-2.02}$, and applying this sequence of natural numbers as the membership numbers in the growing process.

We denote the group size and group degree of a random group by S and U , the group overlapping depth of a random pair of groups by B , the individual degree of a random individual by D , and the interest-sharing number of a random pair of hyperedges by Φ .

Network	$\gamma(s)$	$\gamma(u)$	$\gamma(\beta)$	$\gamma(m)$	$\gamma(d)$	$\gamma(\alpha)$	C	ρ_D	l
arXiv1	5.50	2.14	3.93	1.95	1.84	3.56	0.637	0.584	6.50
arXiv2	6.24	1.63	3.56	1.72	1.68	2.86	0.289	0.382	4.89
IMDB	2.04	0.407	4.80	1.81	1.91	3.62	0.762	0.682	4.29
SF	3.91	2.45	3.76	3.48	2.61	4.60	0.636	0.401	7.06
H_2	2.12	2.39	3.38	n.a.	2.35	n.a.	0.616	0.508	6.13
H_3	2.55	2.46	3.07	n.a.	2.16	n.a.	0.581	0.576	6.71
H_5	2.38	2.09	3.19	n.a.	2.12	n.a.	0.491	0.498	7.85
H_7	3.06	2.81	3.11	n.a.	2.59	n.a.	0.613	0.644	7.62
H_{10}	3.22	2.22	3.53	n.a.	2.38	n.a.	0.686	0.519	6.89
H_{15}	2.90	1.95	3.34	n.a.	2.66	n.a.	0.722	0.478	6.56
$H_{U[1,100]}$	3.66	2.85	3.82	n.a.	3.01	n.a.	0.566	0.422	7.22
H_{pow}	3.91	2.45	3.76	3.48	2.61	4.60	0.636	0.401	7.06

Table 7.2: The exponents γ of power-law fittings $f(x) = x^{-\gamma}$ of s, u, β, m, d and α , and the clustering coefficients C , the assortativity coefficients ρ_D and the average path lengths l of the arXiv GR-QC (arXiv1) and HEP-TH (arXiv2) coauthorship networks, the IMDB actor collaboration network (IMDB), the SourceForge software collaboration network (SF), and the growing hypergraph model with different sequences of membership numbers. For the IMDB network, the exponents γ of s and u for the second region are 5.35 and 3.40.

Results and discussion

Due to the principle of preferential attachment [16], we expect that the group size of all the generated hypergraphs follow power law distributions, which are confirmed by Figure 7.9. The exponents of the power laws are shown in Table 7.4.2. The group

degree of all hypergraphs also follows a power-law distribution, as illustrated in Figure 7.10 (a), where the proper bin size has been used. The exponents are shown in Table 7.4.2. The group degree of H_{pow} always follow a power law distribution. The group overlapping depths of all hypergraphs follow power law distributions with exponents which are relatively larger in absolute values, as depicted in Figure 7.10 (b). All the exponents are reported in Table 7.4.2. Nacher et al. [39] and Manka-Krason et al. [40] showed that the nodal degree of line graphs of simple graphs with power law degree distribution follows a power law distribution. The individual degree distribution of H_2 is just the degree distribution of line graphs of scale-free graphs. The first of plot in Figure 7.11 (a) verifies the conclusion of [39] and [40]. Figure 7.11 (a) shows that the individual degree of H_3 still can be said to follow a power-law and is quite similar to that of H_2 . The distributions of the individual degree of H_5 , H_7 and H_{10} do not follow any power law. The individual degree of H_{15} and $H_{U[1,100]}$ seem to follow power laws $f(x) = x^{-\gamma}$ with negative γ (the exponent $-\gamma$ would be positive). Above all, the individual degree of H_{pow} perfectly follows a power law distribution, as shown in Figure 7.11 (a). The interest-sharing number α of only H_{pow} follows a power-law distribution, as illustrated in the 4th plot on the first row of Figure 7.11 (b). In H_{15} and $H_{U[1,100]}$, the beginning part is linear and the tail is exponential (insets in the two plots on the second row of Figure 7.11 (b)). The clustering coefficients C , the assortativity coefficients ρ_D and the average path lengths l of all the generated hypergraphs H_2 , H_3 , H_5 , H_7 , H_{10} , H_{15} , $H_{U[1,100]}$ and H_{pow} are reported in Table 7.4.2. All the generated hypergraphs exhibit high clustering coefficient, high assortativity coefficient and short average path lengths as what real-world affiliation networks show.

7.5 Chapter conclusion

Many real-world networks, especially social networks, exhibit an overlapping community structure. Affiliation networks are an important type of social networks. We propose a hypergraph representation which reproduces the clique structure of affiliation networks. We give analytically the topological and spectral properties of affiliation networks. We also present formulas which facilitate the computation for characterizing the real-world affiliation networks of ArXiv coauthorship, IMDB actors collaboration and SourceForge collaboration. We propose a preferential attachment based growing hypergraph model for affiliation networks. Numerical analyses show that our hypergraph model with power-law distributed membership numbers reproduces the power-law distributions of group size, group degree, overlapping depth, individual degree and interest-sharing number of real-world affiliation networks, and reproduces the properties of high clustering, assortative mixing and short average path length of real-world affiliation networks.

Chapter 8

Randomness of Brain Networks

Recently, researchers have been curious about the relations between the properties of complex networks and the amount of randomness and structure in complex networks. In this chapter, a spectral randomness metric is proposed to quantitatively measure the randomness of networks. The spectral randomness metric can better capture the randomness of network, compared to assortativity coefficient and average path length. A metric measuring the structure, structure coefficient, is proposed. The randomness and structure of the brain networks of a group of healthy individuals and a groups of patients with Alzheimer's disease have been analysed. We that the brain networks of Alzheimer's disease are statistically more random than the healthy brain networks.

8.1 Spectral randomness metric

The spectra of complex networks have been widely studied [34][24]. Haemers and van Dam [90] conjectured that the spectrum of a graph is the unique fingerprint of that graph, provided its size N is large enough. The adjacency eigenvalues of Erdős-Rényi random graph follow Wigner's semicircle distribution [23]. If the adjacency eigenvalues of a given graph also follow the same Wigner's semicircle distribution of Erdős-Rényi random graph model $G_p(N)$, the graph must be a realization of $G_p(N)$.

Wigner's theorem [75][76][77] states that, the probability distribution of an eigenvalue λ of a random $N \times N$ real symmetric matrix with independently distributed elements a_{ij} with $\sigma^2 = Var[a_{ij}]$, follows a semicircle law,

$$\lim_{N \rightarrow \infty} f_\lambda(x) = \frac{\sqrt{4N\sigma^2 - x^2}}{2\pi N\sigma^2}, \quad |x| \leq 2\sigma\sqrt{N}$$

The adjacency matrices of Erdős-Rényi random graphs satisfy the conditions of Wigner's semicircle law with $\sigma^2 = p(1-p)$. The probability density function of an eigenvalue of $G_p(N)$ has two different features: the largest eigenvalue λ_1 is beyond

the semicircle law and consequently the center of the semicircle shifts leftward to the negative values with different link density p since $\sum_{j=1}^N \lambda_j = 0$.

Theorem 21 *Except the largest eigenvalue λ_1 , the probability density function of a eigenvalue λ , of Erdős-Rényi random graphs with finite N nodes, follows the shifted semicircle law,*

$$f_\lambda(x) = \frac{\sqrt{4Np(1-p) - (x+p)^2}}{2\pi Np(1-p)}, \quad |x| \leq 2\sqrt{Np(1-p)}$$

If we scale the adjacency matrix A of $G_p(N)$ by $\frac{1}{\sqrt{N}}$, $A_N = \frac{A}{\sqrt{N}}$, then the probability density function of a eigenvalue $\lambda^{(A_N)}$ of A_N follows,

$$f_{\lambda^{(A_N)}}(x) = \frac{\sqrt{4p(1-p) - \left(x + \frac{p}{\sqrt{N}}\right)^2}}{2\pi p(1-p)}, \quad |x| \leq 2\sqrt{p(1-p)} \quad (8.1)$$

Proof. See the proof in Appendix D. ■

Denote by $h_G(k)$ the probability density function of a random adjacency eigenvalue of $G(N, L)$, and define ϕ_λ by

$$\phi_\lambda = \int_{x=x_{\min}}^{x_{\max}} |h_G(x) - f_{\lambda^{(A_N)}}(x)|^2 dx \quad (8.2)$$

The eigenvalues of Erdős-Rényi random graphs $G_p(N)$ with infinite N nodes, or of infinite numbers of Erdős-Rényi random graphs $G_p(N)$ with finite N nodes, strictly follows the semicircle distribution, defined by (8.1). The randomness metric $\psi_\lambda(ER)$ of an Erdős-Rényi random graph with finite size, is not zero,

$$\phi_\lambda(ER) = \int_{x=x_{\min}}^{x_{\max}} |h_{ER}(x) - f_{\lambda^{(A_N)}}(x)|^2 dx \neq 0$$

Definition 22 *The spectral randomness ψ_λ is defined by,*

$$\psi_\lambda = \frac{\phi_\lambda}{\mu_{\phi_\lambda(ER)}} \quad (8.3)$$

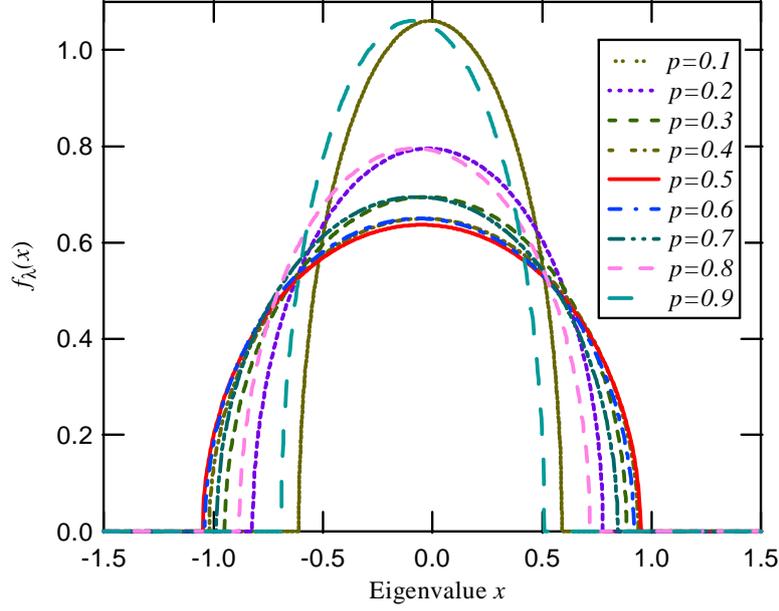


Figure 8.1: The finite variant of semicircle law (8.1) for $G_p(N)$ with link density p .

8.2 Metrics partially indicating randomness

1. Binomial degree distribution. The degree d_j of an arbitrary node j of the class of Erdős-Rényi random graphs $G_p(N)$ obeys the binomial distribution,

$$\Pr[d_j = k] = \binom{N-1}{k} p^k (1-p)^{N-1-k} \quad (8.4)$$

A graph, which contains as much randomness as Erdős-Rényi random graph, must have binomial degree distribution. However, a graph with binomial degree distribution is not necessarily an Erdős-Rényi random graph. For example, the line graph of an Erdős-Rényi random graph has binomial degree distribution, but it has high clustering coefficient and assortativity coefficient [40][41]. Hence, the degree distribution cannot fully capture the randomness of graphs.

2. Assortativity coefficient. If a graph has a positive assortativity coefficient, $\rho_D > 0$, its high-degree nodes preferably connect to other high-degree nodes, and if its assortativity coefficient is negative, $\rho_D < 0$, the high-degree nodes tend to connect to low-degree nodes. The nodes of fully random graphs have no preference in their direct neighbors. Hence, fully random graphs have zero assortativity coefficient. The absolute value of assortativity coefficient $|\rho_D|$ indicates how random a graph

is. The larger $|\rho_D|$ a graph has, the less random the graph is. However, $|\rho_D| = 0$ does not mean a graph is fully random. For example, a random regular graph has $|\rho_D| = 0$, but each node in the random regular graph has the same degree, which is the deterministic feature, reflecting that it is not fully random.

3. Average path length. Average path length describes the average length of shortest path between every pair of nodes in a graph. The small average path length in random graphs and most real-world networks [15], is called the small-world properties. In Watts-Strogatz small-world network model [15], the larger the rewiring probability p_r is, the smaller the average path length is. The average path length in Watts-Strogatz small-world network model indicates how random the graph is. In broader sense, average path length does not always indicate randomness. A counter example is the star graph $K_{1,N-1}$, which has the average path length of 2, and fully deterministic topology.

The spectral randomness ψ_λ can better measure the randomness of graphs, since the spectrum is the fingerprint of a graph. If the adjacency eigenvalues of a graph follows the semicircle distribution (8.1), the graph must be an Erdős-Rényi random graph, and vice versa.

8.3 Randomness of small-world graphs

8.3.1 Non-repetitive rewiring

Given a graph $G(N, L)$, we denote the set of links in G as $\mathcal{L} = \{l_1, l_2, l_3, \dots, l_L\}$, and the set of nodes in G as \mathcal{N} . There are $\binom{N}{2}$ positions where we can place a link, and L positions are occupied by links, which means there are $\binom{N}{2} - L$ free positions left. A link rewiring keeps the number of links unchanged, placing links randomly in the free link positions. After sufficient rewiring, a graph $G(N, L)$ will be so randomized that it approaches the random graph model $G_r(N, L)$.

In non-repetitive rewiring, each link in \mathcal{L} is only rewired once. The set of unrewired links is denoted by \mathcal{L}' . Before the first step of rewiring, $\mathcal{L}' = \mathcal{L} = \{l_1, l_2, l_3, \dots, l_L\}$. In each rewiring step, we choose uniformly at random one link l_r from \mathcal{L}' , then place it uniformly at random in one of the $\binom{N}{2} - L$ free link positions. The set of unrewired links is updated, $\mathcal{L}' = \mathcal{L}' \setminus \{l_r\}$.

8.3.2 Randomness of small-world graphs

By non-repetitive rewiring, we randomize the Watts-Strogatz small-world graphs with 200 nodes, 800 links, and the rewiring probability $p_r = 0.05, 0.1, 0.2, 0.4$. The spectral randomness ψ_λ appears to decrease exponentially with the percentage of rewired links,

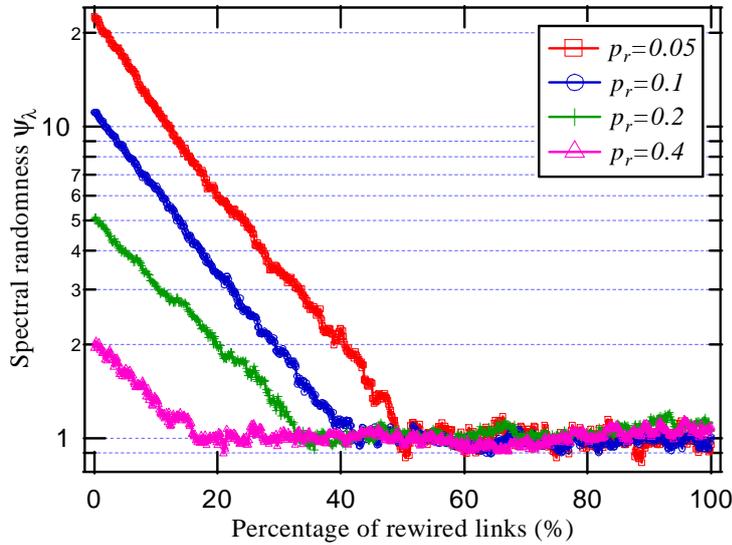


Figure 8.2: The spectral randomness ψ_λ of the Watts-Strogatz small-world graphs under non-repetitive rewiring. The small-world graphs have 200 nodes and 800 links, and the rewiring probability $p_r = 0.05, 0.1, 0.2, 0.4$.

as shown in Figure 8.2. The rewiring probability p_r of the Watts-Strogatz small-world graph indicates the amount of randomness in the graph. As shown in Figure 8.2, the spectral randomness ψ_λ reflects the fact that the small-world graph is more random when p_r is becomes larger. The graph is equivalent to Erdős-Rényi random graph when the spectral randomness $\psi_\lambda \approx 1$, since ψ_λ has been normalized by definition (8.3).

Definition 23 *The spectral randomness ψ_λ of a graph G decreases when G is randomized by non-repetitive rewiring, We define the **structure coefficient** ρ by the minimum percentage of rewired links to randomize G sufficiently that $\psi_\lambda \approx 1$.*

As illustrated in Figure 8.2, the structure coefficient ρ of the Watts-Strogatz small-world graphs with different rewiring probability p_r varies: The less random the graph is (the smaller p_r is), the larger the structure coefficient ρ is.

8.4 Randomness of brain networks

8.4.1 Description of brain networks

The brain network data involved 22 patients with recently diagnosed mild to moderate Alzheimer's disease and 22 healthy individuals who were recruited from the Alzheimer

Group	Healthy	Alzheimer
Group size	22	22
Age (\pm SD)	66(\pm 9)	66(\pm 9)
Gender (M/F)	7/11	12/6

Table 8.1: The subject characteristics of brain network data.

Center of the VU University Medical Centre. Healthy subjects were often spouses of the patients. Alzheimer patients were assessed according to a standard clinical protocol, which involved history taking, physical and neurological examination, an interview with a spouse or close family member, blood tests, magnetic resonance imaging (MRI) scan of the brain according to a standard protocol, routine electroencephalography (EEG), and a thorough neuropsychological assessment. The diagnosis was made in a consensus meeting in which all the available clinical data were considered by a multidisciplinary team. Controls were screened by a neurologist and underwent the same neuropsychological tests as the patients. Exclusion criteria for this study were active psychiatric or neurologic disease. The Local Research Ethics Committee approved the study and all participants gave written informed consent. Main subject characteristics are summarized in Table 8.1.

Cerebral electromagnetic fields were recorded while subjects were seated inside a magnetically shielded room (Vacuumschmelze GmbH, Hanau, Germany) using a 151-channel whole-head MEG system (CTF Systems Inc., Port Coquitlam, BC, Canada). For technical reasons two channels had to be omitted, yielding 149 channels or sensors for analyses. Fields were measured during a no-task, eyes-closed condition. During the MEG recording, patients were instructed to close their eyes, stay awake, and reduce eye movements. For each subject care was taken to select four artifact-free data segments of about 6.5 seconds by two of the investigators (WDH and CS), who were blinded to the diagnosis. Typical artifacts were due to (eye) movements, swallowing, dental prosthetics, or drowsiness.

Since slower and faster frequency ranges have different neuropsychological meanings, it is common practice to filter MEG data in different frequency bands prior to performing further analysis. We used all commonly used relevant frequency bands: delta (0.5-4 Hz), theta (4-8 Hz), lower alpha (8-10 Hz), higher alpha (10-13 Hz), beta (13-30 Hz) and gamma (30-45 Hz). Thus, all graph analyses were performed for these frequency bands separately.

Correlations between all pair-wise combinations of MEG channels were computed with the Synchronization Likelihood (SL, mathematical details can be found in [96]). The end result of computing the SL for all channel combinations is a square matrix (with 149 rows and columns, equal to the number of MEG channels), where each entry contains the resulting SL value of the sensor pair. This connectivity matrix is then

freq. (Hz)	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
0.5-4	0.17	0.24	0.29	0.22	0.22	0.21	0.30	0.31	0.32	0.22	0.17
4-8	0.49	0.55	0.55	0.58	0.48	0.65	0.50	0.54	0.55	0.54	0.50
8-10	0.36	0.43	0.39	0.29	0.39	0.39	0.33	0.40	0.38	0.40	0.34
10-13	0.43	0.33	0.31	0.34	0.39	0.38	0.37	0.39	0.42	0.34	0.38
13-30	0.58	0.57	0.50	0.51	0.52	0.45	0.55	0.51	0.56	0.51	0.48
30-45	0.81	0.78	0.66	0.81	0.76	0.73	0.67	0.75	0.67	0.73	0.71
freq. (Hz)	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22
0.5-4	0.26	0.18	0.17	0.16	0.15	0.16	0.22	0.22	0.17	0.14	0.20
4-8	0.51	0.48	0.43	0.52	0.45	0.65	0.52	0.47	0.49	0.50	0.47
8-10	0.34	0.41	0.32	0.38	0.34	0.26	0.45	0.41	0.37	0.35	0.25
10-13	0.39	0.35	0.31	0.32	0.40	0.38	0.37	0.33	0.37	0.34	0.35
13-30	0.57	0.50	0.59	0.51	0.50	0.45	0.51	0.55	0.53	0.50	0.49
30-45	0.70	0.82	0.77	0.84	0.82	0.73	0.81	0.95	0.80	0.67	0.79

Table 8.2: ρ for patients with Alzheimer.

used for further graph spectral analysis. The SL values are a measure of the amount of interaction or communication between two brain regions, and are taken as the edge weights. However, there are some bad channels which cause disjoint nodes in the brain networks. We rule out all the disjoint nodes. There are 44 patients half of whom have Alzheimer. For each band, four measurements have been done and the averaged network has been taken. We take $T = 0.019$ to threshold the weighted brain networks into unweighted networks.

8.4.2 The structure coefficient of brain networks

The brain networks of both groups in Table 8.1 are randomized by non-repetitive rewiring, and the spectral randomness ψ_λ is computed. Figure 8.3 shows the spectral randomness ψ_λ of the brain networks of an individual from the group Alzheimer decrease exponentially with the percentage of rewired links. The structure coefficient ρ of all brain networks are computed by the definition, as shown in Table 8.2 and 8.3.

Figure 8.4 shows that the structure coefficient ρ of individuals in both groups fluctuates. There is no threshold of ρ which we can use to distinguish two groups of brain networks. Hence, the structure coefficient ρ has been averaged over the group Alzheimer and the group Healthy, as shown in Figure 8.5. For five out of six frequency bands, 0.5-4 Hz, 4-8 Hz, 8-10 Hz, 10-13 Hz, and 30-45 Hz, the healthy brain networks have higher average structure coefficient than the brain networks with Alzheimer's disease. In other words, all frequency bands except the beta band 13-30 Hz, the healthy brains have more structure (less randomness) than the brains with Alzheimer's disease.

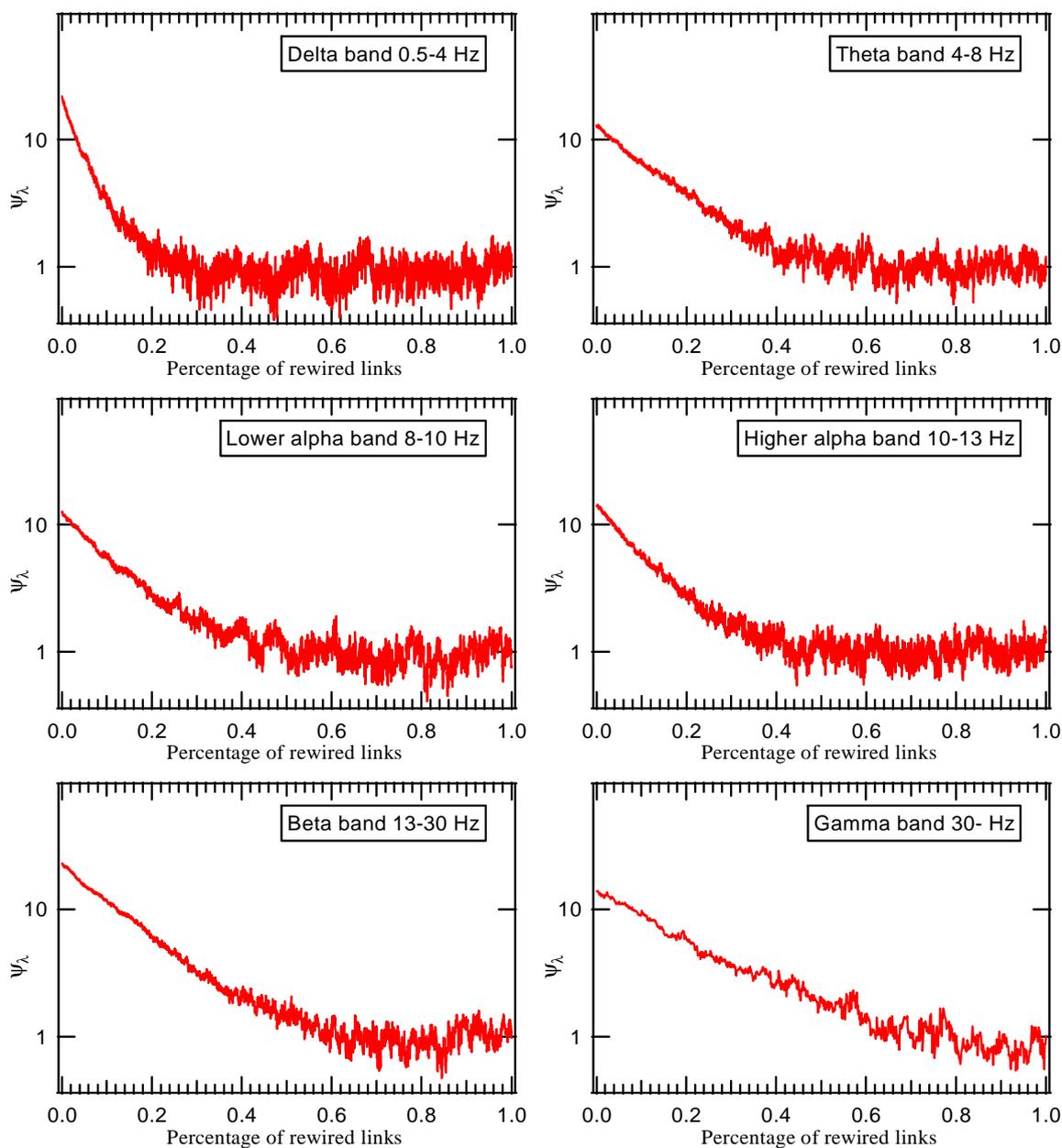


Figure 8.3: ψ_λ decreases with the percentage of rewired links. The Non-Repetitive Rewiring has been performed. The simulated brain networks are from one patient of the group Alzheimer.

freq. (Hz)	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
0.5-4	0.31	0.17	0.18	0.23	0.21	0.23	0.22	0.22	0.27	0.32	0.27
4-8	0.48	0.46	0.49	0.46	0.61	0.52	0.53	0.51	0.58	0.53	0.50
8-10	0.38	0.38	0.37	0.37	0.34	0.36	0.39	0.45	0.38	0.27	0.38
10-13	0.36	0.36	0.38	0.36	0.31	0.40	0.37	0.40	0.35	0.39	0.35
13-30	0.48	0.50	0.58	0.51	0.53	0.56	0.45	0.53	0.54	0.58	0.48
30-45	0.76	0.77	0.77	0.76	0.68	0.79	0.68	0.71	0.72	0.71	0.83
freq. (Hz)	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22
0.5-4	0.41	0.21	0.27	0.22	0.20	0.19	0.23	0.26	0.24	0.22	0.29
4-8	0.51	0.51	0.53	0.57	0.45	0.51	0.53	0.58	0.51	0.52	0.67
8-10	0.36	0.38	0.37	0.38	0.41	0.38	0.34	0.37	0.41	0.36	0.42
10-13	0.36	0.46	0.37	0.43	0.38	0.37	0.40	0.36	0.36	0.40	0.36
13-30	0.52	0.40	0.53	0.53	0.51	0.51	0.47	0.56	0.48	0.50	0.52
30-45	0.87	0.84	0.99	0.79	0.72	0.78	0.73	0.81	0.72	0.79	0.75

Table 8.3: The structure coefficient ρ of brain networks of the group Healthy.

In a study using electroencephalography (EEG) recordings, the brain networks of Alzheimer patients have more random network topology [97]. A functional MRI study using graph theory showed lower clustering coefficients, suggesting an increase of network randomness in the brain networks of Alzheimer patients [98]. The higher randomness in the brain networks of Alzheimer patients we found in this study is in line with previous researches, and might be interpreted as reflecting the loss of structure and organization in the brain. The gradually spreading damage in dementia disrupts the balanced brain network structure, impairing optimal information processing, and leading to progressive cognitive symptoms. For diagnostic purposes, it would be very useful if brain networks of individuals could be classified with a high level of accuracy. At present, there is no single test that achieves this, and although the current clinical diagnostic work-up allows for a diagnosis with a fairly high degree of certainty, the golden criteria to diagnose Alzheimer is still post-mortem tissue examination.

8.5 Chapter conclusion

In this chapter, we propose a spectral randomness metric to quantitatively measure the randomness of networks. The spectral randomness metric can better capture the randomness of network, compared to assortativity coefficient and average path length. We introduce non-repetitive rewiring to randomize networks, and investigate how the spectral randomness changes when networks are gradually randomized. We propose a metric, structure coefficient, to measure the structure of networks. The randomness

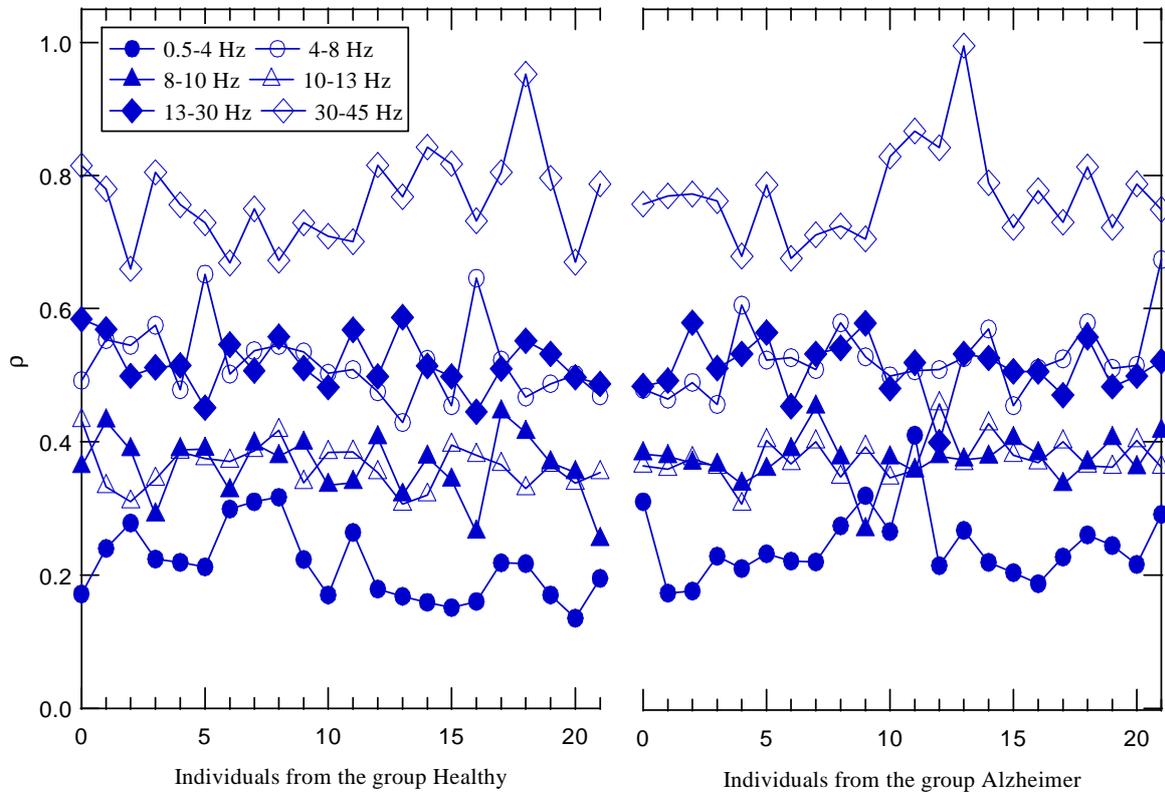


Figure 8.4: The structure coefficient ρ of the brain networks of both groups of Healthy and Alzheimer.

and structure of the brain networks of a group of healthy individuals and a groups of patients with Alzheimer's disease have been analysed. We show that the brain networks of Alzheimer's disease are statistically more random than the healthy brain networks. The higher randomness in the brain networks of Alzheimer patients might be interpreted as reflecting the loss of structure and organization in the brain. For diagnostic purposes, it would be very useful if brain networks of individuals could be classified with a high level of accuracy. At present, there is no single test that achieves this, and although the current clinical diagnostic work-up allows for a diagnosis with a fairly high degree of certainty, the golden criteria to diagnose Alzheimer is still post-mortem tissue examination. The ability of graph spectral analysis to describe unique network features is promising in this regard, although in our study the group differences were not strong enough to use the present measure as disease marker. One reason for this might be that biological data contains much variability (also within individuals) and noise. However, by adjusting methodological choices like recording conditions or

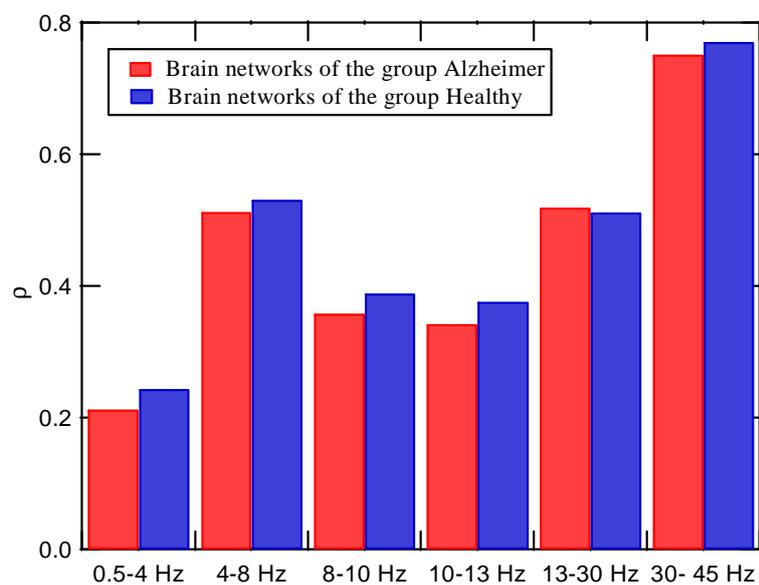


Figure 8.5: The average structure coefficient ρ of brain networks of group Alzheimer and Healthy. Except for the frequency band 13-30 Hz, healthy brain networks have higher average structure coefficient ρ than brain networks with Alzheimer.

selection of connectivity measure and network analysis parameters, it is conceivable that better results can be achieved.

Chapter 9

Conclusions

Various types of complex networks, including man-designed infrastructure networks (Internet, telecommunication networks, electrical power grids, transportation networks, gas network and water networks) and natural networks (social networks, ecological networks, and biological networks), shape our daily life. We depend on diverse complex networks more and more. We become aware that the complex infrastructure networks are of crucial importance to our society when the power grid is down, the financial system fails, and communication networks break down. Since the great importance of robustness of complex networks to us, attentions of many researchers has been drawn to make our complex infrastructure networks more robust and optimize the performance of them. This thesis addresses some topics among many aspects of robustness and optimization of complex networks.

Chapter 2 studies the reconstruction of networks from the spectral domain. We define the reconstructability coefficient θ of a network as the maximum number of eigenvalues that can be set to zero, given that the adjacency matrix can be reconstructed exactly. We studied relation between the reconstructability coefficient θ and the network size N for various types of networks, including Erdős-Rényi random graphs, Barabási-Albert scale-free networks and Watts-Strogatz small-world networks, and some deterministic graphs. We found that the reconstructability coefficient θ obeys a general linear scaling law (1.1), $E[\theta] = aN$. For sufficiently large N , a portion a of the smallest eigenvalues in absolute value can be removed from the spectrum and the adjacency matrix is still reconstructable with its original eigenvectors. We also studied the properties of the mean of the reconstructability coefficient $E[\theta]$ under eigenvector perturbation for Erdős-Rényi random graphs. The normalized Gaussian distributed noise matrix, scaled by the perturbation factor ε , was added to the eigenvector matrix X . We show that the linear scaling law $E[\theta] = aN$ still holds for Erdős-Rényi random graphs until the perturbation factor ε exceeds 0.2. The basic eigenvalue relation (2.1) shows that the set of orthogonal eigenvectors are weighted by their corresponding eigenvalues. Any eigenvector specifies an orthogonal direction in the N -dimensional space. The eigenvector

with an eigenvalue in absolute value close to 0 contains redundant information about the topology of the graph, in the sense that after the removal of this eigenvalue the network can still be reconstructed from the remaining spectrum. We observe that when the graphs have more constraints to be generated, the parameter a is higher. Those deterministic graphs, like path, ring and grid graphs, have more constraints to be generated, than Erdős-Rényi random graphs, power law graphs and small-world graphs. In the spectral domain, the more generating constraints the graphs have, the less that N -dimensional space is “sampled”, or in other words, the less spectral bases (eigenvectors) we need to reconstruct the graphs.

Chapter 3 studies some properties of the energy of networks, especially, the relation between the energy and the assortativity for different type of network. The relations between the energy E_G and three other metrics, the multiplicity m_0 of zero adjacency eigenvalues, the multiplicity m_{-1} of -1 adjacency eigenvalues, and the assortativity coefficient ρ_D of graph G with constant number of nodes and constant number of links have been studied. We show that the energy E_G has a linear relation with m_0 , not only for the molecular graphs coined by Ivan Gutman, but also for quite general graphs: deterministic graphs and random graphs. We also show that the energy E_G decreases linearly with m_{-1} for the graphs constructed by combining Erdős-Rényi random graphs and triangles. We use the degree-preserving rewiring to increase/decrease the assortativity of an Erdős-Rényi random graph step by step towards $1/-1$, and show that the energy decreases roughly linearly with the rewiring steps, and the energy is at its maximum when the graph is non-assortative, and the energy decreases when the graph becomes assortative/disassortative. The claims of linearity for graph energy are supported by extensive simulations.

Chapter 4 and 5 presents two algorithms, MARINLINGA and ILIGRA, which reconstructs networks from the line graph domain. For the algorithm MARINLINGA, we transformed the problem of graph reconstruction from the line graph domain into the problem of constructing a graph from the link adjacency matrix (LAM). MARINLINGA consists of two sub-algorithms: the matrix relabeling algorithm and the construction algorithm. The matrix relabeling algorithm preprocesses the LAM into the special order by which we can determine the neighboring links of the first link and the endnodes of the first link incident to the neighboring links. The construction algorithm makes the first two nodes be the endnodes of the first link by default, and thereafter, determines the endnodes of the remaining links. MARINLINGA has a worst case complexity of $O(N_{l(G)}^2)$, where $N_{l(G)}$ denotes the number of nodes of the line graph. We have demonstrated that MARINLINGA is more time-efficient compared to Roussopoulos’ algorithm for connected line graphs. The second algorithm, ILIGRA, constructs its root graph from a given line graph, and check whether the given line graph is a line graph during the construction. ILIGRA also works for disconnected line graphs and finds the connected components of input line graphs during their root graph constructions. The time complexity of ILIGRA is linear in the number of nodes in the given line graphs without checking if the given

graph is a line graph. The time complexity of ILIGRA with full functionality is linear in the number of links in the given line graphs. We showed numerical comparisons with the algorithms of Lehot, Roussopoulos, and Degiorgi and Simon. Given line graphs with small link density (i.e. sparse graphs), ILIGRA is the fastest algorithm in root graph construction.

Chapter 6 gives some properties of line graphs and proposes a random line graph model. We found that there are multiple bands of consecutive positive integers, which can never appear as the number of links L in $H(N, L)$. The exact expressions of bands and bandgaps of L have been derived. We proposed a model which can randomly generate simple graphs which are line graphs of other simple graphs. The model constructs line graphs by merging step by step a pair of nodes of the cliques, which we use to construct line graphs. Obeying necessary rules to ensure that the resulting graphs are line graphs, two nodes to be merged are randomly chosen at each step. If the cliques are all of the same size, the assortativity of the line graphs are each step are close to 0, and the assortativity of the corresponding root graphs increases linearly from -1 to 0 with the steps of merging nodes. The linear law for the assortativity provides a new method to tune the assortativity besides the method of degree-preserving rewiring. We showed that, when we “dope” the constructing elements of the line graphs - the cliques of the same size- with a relatively smaller number of cliques of different size, the characteristics of the assortativity of the line graphs is changed.

Chapter 7 investigates the overlapping community structure of social networks and propose a growing hypergraph model for social networks which reproduces many properties of real-world social networks. Social networks exhibit an overlapping community structure. Affiliation networks are an important type of social networks. A hypergraph representation which reproduces the clique structure of affiliation networks was proposed. We gave analytically the topological and spectral properties of affiliation networks. The real-world affiliation networks of ArXiv coauthorship, IMDB actors collaboration and SourceForge collaboration were characterized. We proposed a preferential attachment based growing hypergraph model for affiliation networks. Numerical analyses showed that our hypergraph model with power-law distributed membership numbers reproduces the power-law distributions of group size, group degree, overlapping depth, individual degree and interest-sharing number of real-world affiliation networks, and reproduces the properties of high clustering, assortative mixing and short average path length of real-world affiliation networks.

Chapter 8 proposes a spectral randomness metric and studies the randomness differences between the brain networks of Alzheimer’s disease and the healthy brain networks. In order to quantitatively measure the randomness of networks, we proposed a spectral randomness metric. Compared to assortativity coefficient and average path length, the spectral randomness metric can better capture the randomness of network. The non-repetitive rewiring was introduced to randomize networks, and investigate how the spectral randomness changes when networks are gradually randomized. We proposed

a metric, structure coefficient, to measure the structure of networks. The randomness and structure of the brain networks of a group of healthy individuals and a groups of patients with Alzheimer's disease have been analysed. We show that the brain networks of Alzheimer's disease are statistically more random than the healthy brain networks. The higher randomness in the brain networks of Alzheimer patients might be interpreted as reflecting the loss of structure and organization in the brain.

9.1 Contribution summary

The contributions of this thesis are summarized as follows:

- The reconstructability of complex networks from the spectral domain has been studied and a general linear law of reconstructability coefficient has been found.
- The energy of networks has found to have linear relationships with the multiplicity of 0 eigenvalues and the multiplicity of -1 eigenvalues for different types of networks, and the relation between energy and assortativity has been studied for many types of networks.
- The reconstruction of complex networks from the line graph domain has been studied, and two efficient algorithms, MARINLINGA and ILIGRA,, have been presented.
- A random line graph model has been proposed. The bandgaps for the number of links in line graphs have been found. A linear law for the assortativity of the root graphs of the proposed random line graph model has been found.
- The overlapping community structure of social network has been characterized and a hypergraph model which reproduces the overlapping community structure and many other properties of social networks has been proposed.
- A spectral randomness measure of networks has been proposed, and by this measure, it is found that the brain networks with Alzheimer's disease are statistically more random than the healthy brain networks.

9.2 Future work

The research field of robustness and optimization of complex networks is much broader than the topics discussed in this thesis. Due to the time limit of the Ph.D. work and the complexity of the problems, some aspects of this thesis have not yet been understood thoroughly or are not solved analytically. In the following, we list some problems we consider for future work.

1. Chapter 2 studied the reconstructability of complex networks from the adjacency spectral domain. It would be interesting to study the reconstructability of complex networks from the Laplacian spectral domain and the property of the corresponding reconstructability coefficient.
2. Chapter 2 defined the reconstructability coefficient by the maximum number of adjacency eigenvalues that can be removed, subject to the condition that the adjacency matrix can be reconstructed exactly. A new metric, reducible energy, can be defined by the sum of the absolute values of adjacency eigenvalues that can be removed, subject to the same condition. We would like to study the property of the reducible energy.
3. Chapter 3 found some interesting properties of energy, but with numerical methods. We would like explain the properties of energy in an analytical way, especially for the relations between energy and assortativity for some typical network models.
4. Chapter 7 proposed a hypergraph model for social networks. The simulation results show that the hypergraph model reproduces the properties of real-world social networks. To analytically solve the model and predicts the properties of the model with different parameters, is a challenging and exciting task.
5. The largest eigenvalue λ_1 of the adjacency matrix of a graph G is a crucial indicator in the dynamic process on the graph, such as virus spread. Besides, λ_1 also plays an important role in the phase-transition threshold of a network of coupled oscillators. The largest eigenvalue λ_1 is much larger than the second largest eigenvalue λ_2 for many graphs. For example, the largest eigenvalue λ_1 of an Erdős-Rényi random graph, is much larger than the rest of the eigenvalues which follow a Wigner's semicircle distribution. There is a clear gap between the largest eigenvalue λ_1 and the other eigenvalues. We define a new adjacency matrix $A(\alpha, \beta)$ by assigning the value α to the existence of a link and β otherwise, where α and $\beta \neq \alpha$ can be any real number. When $\alpha = 1 - p$ and $\beta = -p$, where $p = L/\binom{N}{2}$ is the link density, all the eigenvalues of $A(1 - p, -p)$ of an Erdős-Rényi random graph follow a Wigner's semicircle distribution, according to the Wigner's Theorem [75][76][77]. There is no clear spectral gap. We would like to find the important eigenvalue of $A(1 - p, -p)$ which corresponds to the largest eigenvalue λ_1 of $A(1, 0)$. We also would like to answer a more general question: Is it possible to map the existing theory for the spectrum of $A(1, 0)$ to the spectrum of $A(\alpha, \beta)$?

Appendix A

Properties of the matrix E_k

The matrix E_k is introduced in (2.1). From the definition $E_k = x_k x_k^T$, we deduce that $E_k = E_k^T$, thus, symmetric. The explicit form of the matrix E_k is

$$E_k = x_k x_k^T = \begin{bmatrix} (x_{k1})^2 & x_{k1}x_{k2} & x_{k1}x_{k3} & \cdots & x_{k1}x_{kn} \\ x_{k2}x_{k1} & (x_{k2})^2 & x_{k2}x_{k3} & \cdots & x_{k2}x_{kn} \\ x_{k3}x_{k1} & x_{k3}x_{k2} & (x_{k3})^2 & \cdots & x_{k3}x_{kn} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{kn}x_{k1} & x_{kn}x_{k2} & x_{kn}x_{k3} & \cdots & (x_{kn})^2 \end{bmatrix}$$

which shows that the diagonal element $(E_k)_{ii} = (x_{ki})^2$ equals the square of the i -th vector component of the eigenvector x_k . Hence,

$$\text{trace}(E_k) = \sum_{i=1}^n (x_{ki})^2 = x_k^T x_k = 1 \quad (\text{A.1})$$

It follows from the orthogonality property (A.21) in [23, p. 444] of eigenvectors x_k of a symmetric matrix that $E_k^2 = E_k$ and $E_k E_m = 0$ for $k \neq m$. Let us denote the eigenvalue equation $E_k y_j = \xi_j y_j$ of the symmetric matrix E_k . After left-multiplication by E_k , we obtain $E_k^2 y_j = \xi_j^2 y_j$ and, since $E_k^2 = E_k$, we arrive at $E_k y_j = \xi_j^2 y_j$. Hence, for any eigenvalue ξ_j and corresponding eigenvector y_j , we have that $\xi_j y_j = \xi_j^2 y_j$, which implies that ξ_j is either zero or 1. The trace-relation (A.7) in [23, p. 436] and (A.1) indicates that $\sum_{j=1}^n \xi_j = 1$. The eigenvalues of E_k directly follow from the rank-one update formula because

$$\det(x_k x_k^T - \lambda I) = (-\lambda)^n \det\left(I - \frac{1}{\lambda} x_k x_k^T\right) = (-\lambda)^{n-1} (\lambda - 1) \quad (\text{A.2})$$

and are precisely the same as those of the adjacency matrix of the complete graph K_n . Consequently, we conclude that $n - 1$ eigenvalues are zero and one eigenvalue equals

1, such that $\|E_k\|_2 = 1$ that follows from (A.33) in [23, p. 448]. The zero eigenvalues imply that $\det(E_k) = 0$ and that the inverse of E_k does not exist. Geometrically, this is understood because, by projecting, information is lost and the inverse cannot create information.

Appendix B

Quality of the bound of Ξ

The bound of Ξ is introduced in (2.4). When the graph belongs to the class of Erdős-Rényi random graph $G_p(N)$, the spectrum rapidly tends to Wigner's Semicircle Law [23, p. 489]. We confine ourselves to the class of Erdős-Rényi random graph $G_p(N)$ to estimate the quality of the bound (2.4). In particular, we compute the bound (2.4) of θ probabilistically as

$$\Pr \left[\sum_{k=1}^{\theta} |\lambda_{(k)}| < \frac{1}{2} \right] = 1 - \epsilon$$

meaning that the probability that $\sum_{k=1}^{\theta} |\lambda_{(k)}| < \frac{1}{2}$ is almost sure, when $\epsilon > 0$ is chosen arbitrarily small. However, the distribution of the θ smallest order statistics is difficult and we content ourselves to compute the average of the sum of order statistics

$$r = \sum_{k=1}^{\theta} E [|\lambda_{(k)}|]$$

First, we compute the absolute value Y of a random variable X . The event that $\{Y \leq y\}$ is equivalent to $\{|X| \leq y\} = \{-y \leq X \leq y\}$ and nonexistent for $y < 0$. Hence

$$\Pr [Y \leq y] = \Pr [-y \leq X \leq y] = F_X(y) - F_X(-y)$$

and, after differentiation with respect to y , we find the relation for the probability density function as

$$f_{|X|}(y) = f_X(y) + f_X(-y)$$

Applied to Wigner's Semicircle Law,

$$f_{\left| \lambda \left(\frac{A}{\sqrt{N}} \right) \right|}(x) = \frac{1}{\pi \sigma^2} \sqrt{4\sigma^2 - x^2} 1_{|x| \leq 2\sigma}$$

and we find the distribution $F_{|\lambda(\frac{A}{\sqrt{N}})|}(t) = \frac{1}{\pi\sigma^2} \int_0^t \sqrt{4\sigma^2 - x^2} 1_{|x| \leq 2\sigma} dx$ for $t \leq 2\sigma$ as

$$F_{|\lambda(\frac{A}{\sqrt{N}})|}(t) = \Pr \left[\left| \lambda \left(\frac{A}{\sqrt{N}} \right) \right| \leq t \right] = \frac{2}{\pi} \arcsin \frac{t}{2\sigma} + \frac{2}{\pi} \frac{t}{2\sigma} \sqrt{1 - \left(\frac{t}{2\sigma} \right)^2}$$

while $F_{|\lambda(\frac{A}{\sqrt{N}})|}(t) = 1$ for $t > 2\sigma$. The Taylor expansion is

$$F_{|\lambda(\frac{A}{\sqrt{N}})|}(2\sigma x) = \frac{4}{\pi} x \left(1 - \frac{1}{6} x^2 + O(x^4) \right)$$

which shows that, for small x ,

$$F_{|\lambda(\frac{A}{\sqrt{N}})|}(2\sigma x) \simeq \frac{4}{\pi} x \tag{B.1}$$

Second, using a similar argument as in [23, p. 377] that approximates the exact distribution of the k -th order statistics as a Gaussian with mean

$$\mu = N \cdot F_{\lambda(\frac{A}{\sqrt{N}})}(x)$$

and variance

$$\sigma^2 = N \cdot F_{\lambda(\frac{A}{\sqrt{N}})}(x) \left(1 - F_{\lambda(\frac{A}{\sqrt{N}})}(x) \right)$$

that tends to a delta function for large N , we can approximate,

$$E [|\lambda_{(k)}|] \simeq 2\sigma\sqrt{N} F_{\lambda(\frac{A}{\sqrt{N}}}^{-1} \left(\frac{k}{N} \right)$$

Using (B.1) yields

$$E [|\lambda_{(k)}|] \simeq \sqrt{N}\pi\sigma^2 \left(\frac{k}{N} \right) = \frac{\pi\sigma^2}{\sqrt{N}} k$$

such that

$$\begin{aligned} r &= \sum_{k=1}^{\theta} E [|\lambda_{(k)}|] \simeq \frac{\pi\sigma^2}{\sqrt{N}} \sum_{k=1}^{\theta} k \\ &= \frac{\pi\sigma^2}{2\sqrt{N}} (\theta)(\theta+1) \simeq \frac{\pi\sigma^2}{2\sqrt{N}} \theta^2 \end{aligned}$$

The requirement that $r < \frac{1}{2}$, then implies approximately that

$$\theta < \frac{N^{1/4}}{\sigma\sqrt{\pi}}$$

The derivation shows that the conservative bound is inappropriate because simulations show that $\theta = O(N)$, while the conservative bound points to $\theta = O(N^{1/4})$.

Appendix C

The Initialization of MARINLINGA When $s_3 = 0$

Theorem 6 cannot be used when $s_3 = 0$. Since there exists limited number of cases of $s_3 = 0$, we can still accomplish the initialization.

C.1 When $s_1 = 1$

Link 1 has only one right neighboring link: link 2. Link 1 does not have left neighboring links. The initial state of E is \mathcal{E}_1 . Lines 3-4 of Algorithm 4.18 initialize E by \mathcal{E}_1 .

$$\mathcal{E}_1 = \begin{bmatrix} 1 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

C.2 When $s_1 = 2$

There are different adjacency patterns. The submatrix of C in Figure C.1 (a) implies that, links 2 and 3 are adjacent to link 1, and link 2 is not adjacent to link 3. Links 2 and 3 must be incident to two different endnodes of link 1. The pattern in Figure C.1 (b) has two possible configurations K_3 and $K_{1,3}$. If $s_1 = 2$ and $s_2 = 0$, the initial state is $\mathcal{E}_{2,a}$, as shown in lines 1-2 of Algorithm C.2. When $s_1 = 2$ and $s_2 = 1$, because the graph is connected, either $c_{2,4} = c_{3,4} = 1$ or $c_{2,4} = 0, c_{3,4} = 1$ or $c_{2,4} = 1, c_{3,4} = 0$. If $c_{2,4} = c_{3,4} = 1$, the initial state is $\mathcal{E}_{2,b,1}$, which is K_3 , otherwise the initial state is $\mathcal{E}_{2,b,2}$, which is $K_{1,3}$, as shown in lines 8-12 of Algorithm C.2.

$$\mathcal{E}_{2,a} = \begin{bmatrix} 1 & 1 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$\mathcal{E}_{2,b,1} = \begin{bmatrix} 1 & 1 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \mathcal{E}_{2,b,2} = \begin{bmatrix} 1 & 1 & 1 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

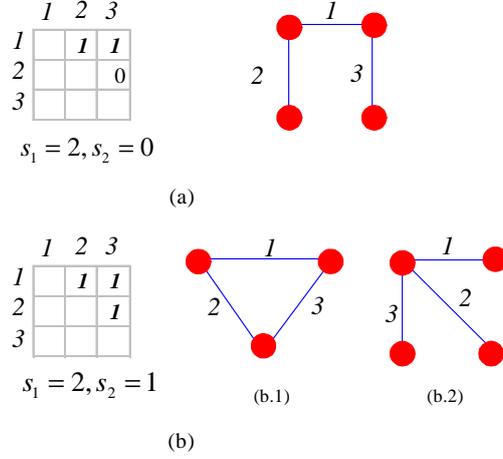


Figure C.1: The adjacency patterns of link 1 and its neighboring links when $s_1 = 2$. The graphs on the right are the possible configurations correspondingly.

```

 $E_{2 \times L_G} \leftarrow \text{INITIALIZATION2}(C, s_2)$ 
1  if  $s_2 = 0$  then
2     $E \leftarrow \mathcal{E}_{2,a}$ 
3  else
4    if  $c_{2,4} = 1$  and  $c_{3,4} = 1$  then
5       $E \leftarrow \mathcal{E}_{2,b,2}$ 
6    else
7       $E \leftarrow \mathcal{E}_{2,b,1}$ 

```

Figure C.2: Meta-code for INITIALIZATION2.

C.3 When $s_1 = 3$

There are two recognizable adjacency patterns as described in Figure C.3 (b), and (c). Taking pattern (c) as an example, links 1, 2 and 3 are pairwise adjacent, then the configuration of them is K_3 or $K_{1,3}$, as shown in Figure C.1 (b). Link 4 is also adjacent to link 1, but not adjacent to links 2 and 3, suggesting that the configuration of links 1, 2 and 3 must be K_3 , and link 4 is incident to the other endnode of link 1. Figure C.3 (a) depicts the smallest forbidden link adjacency pattern in a LAM. The adjacency relation of links 1, 2 and 3 is recognizable, and the configuration is a path on four nodes, as shown in Figure C.1 (a). Link 4 is adjacent to link 1, then link 4 must be also adjacent to links 2 or 3. Hence the pattern is forbidden. If $s_1 = 3$ and $s_2 = 0$, the initial state is

$\mathcal{E}_{3,b}$ (lines 1-2 of Algorithm C.4). If $s_1 = 3$, $s_2 = 1$ and $c_{3,4} = 0$, the initial state is $\mathcal{E}_{3,c}$ (lines 3-5 of Algorithm C.4). When $s_1 = 3$, $s_2 = 1$ and $c_{3,4} = 1$, due to the connectivity of the concerned graph, either $c_{2,5} = c_{3,5} = c_{4,5} = 1$ or $c_{2,5} = c_{3,5} = 1, c_{4,5} = 0$ or $c_{2,5} = c_{3,5} = 0, c_{4,5} = 1$ or $c_{2,5} = 1, c_{3,5} = c_{4,5} = 0$ or $c_{2,5} = 0, c_{3,5} = c_{4,5} = 1$. If $c_{2,5} \neq c_{3,5}$, the initial state is $\mathcal{E}_{3,d,2}$, else if $c_{2,5} = c_{3,5} \neq c_{4,5}$, the initial state is $\mathcal{E}_{3,d,1}$, else if $c_{2,5} = c_{3,5} = c_{4,5} = 1$, we need to look further at the relation of $c_{2,6}$ and $c_{3,6}$: if $c_{2,6} \neq c_{3,6}$, the initial state is $\mathcal{E}_{3,d,2}$, else the initial state is $\mathcal{E}_{3,d,1}$ (lines 11-15 of Algorithm C.4). If there are only 5 links in total and $c_{2,5} = c_{3,5} = c_{4,5} = 1$, one can choose any of $\mathcal{E}_{3,d,1}$ and $\mathcal{E}_{3,d,2}$ as the initial state, and get isomorphic configurations. If $s_1 = 3$, $s_2 = 2$ and $s_3 = 0$, the same method is employed (lines 21-26 of Algorithm C.4).

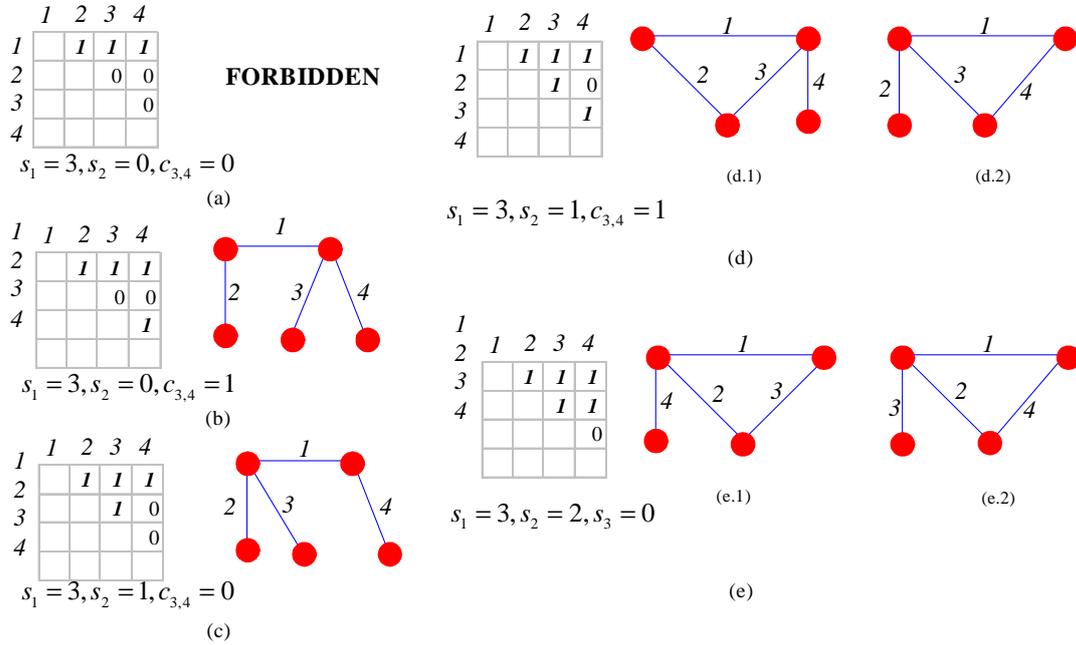


Figure C.3: The adjacency patterns of link 1 and its neighboring links when $s_1 = 3$. Pattern (a) is forbidden, and patterns (b), (c) and (f) correspond to only one configuration respectively. Patterns (d) and (e) both have two possible configurations.

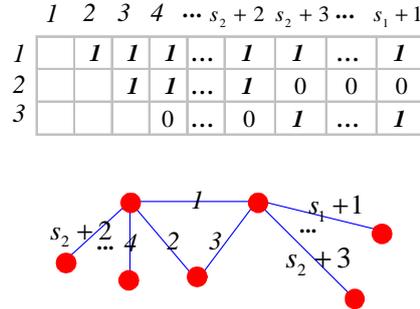
$$\begin{aligned} \mathcal{E}_{3,b} &= \begin{bmatrix} 1 & 1 & 2 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \mathcal{E}_{3,c} = \begin{bmatrix} 1 & 1 & 1 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\ \mathcal{E}_{3,d,1} &= \begin{bmatrix} 1 & 1 & 2 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \mathcal{E}_{3,d,2} = \begin{bmatrix} 1 & 1 & 1 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\ \mathcal{E}_{3,e,1} &= \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \mathcal{E}_{3,e,2} = \begin{bmatrix} 1 & 1 & 1 & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \end{aligned}$$

```

 $E_{2 \times L_G} \leftarrow \text{INITIALIZATION3}(C, s_2, s_3)$ 
1 if  $s_2 = 0$  then
2    $E \leftarrow \mathcal{E}_{3,b}$ 
3 else if  $s_2 = 1$  and  $c_{3,4} = 0$  then
4    $E \leftarrow \mathcal{E}_{3,c}$ 
5 else if  $s_2 = 1$  and  $c_{3,4} = 1$  then
6   if  $c_{2,5} \neq c_{3,5}$  or  $(c_{2,5} = c_{3,5}$  and  $c_{2,5} = c_{4,5}$  and  $L_G = 5)$  then
7      $E \leftarrow \mathcal{E}_{3,d.2}$ 
8   else if  $c_{2,5} = c_{3,5}$  and  $c_{2,5} \neq c_{4,5}$  then
9      $E \leftarrow \mathcal{E}_{3,d.1}$ 
10  else if  $c_{2,5} = c_{3,5}$  and  $c_{2,5} = c_{4,5}$  and  $c_{2,6} = c_{3,6}$  then
11     $E \leftarrow \mathcal{E}_{3,d.1}$ 
12  else if  $c_{2,5} = c_{3,5}$  and  $c_{2,5} = c_{4,5}$  and  $c_{2,6} \neq c_{3,6}$  then
13     $E \leftarrow \mathcal{E}_{3,d.2}$ 
14 else if  $s_2 = 2$  and  $s_3 = 0$  then
15  if  $c_{2,5} \neq c_{3,5}$  or  $(c_{2,5} = c_{3,5}$  and  $c_{2,5} = c_{4,5}$  and  $L_G = 5)$  then
16     $E \leftarrow \mathcal{E}_{3,e.2}$ 
17  else if  $c_{2,5} = c_{3,5}$  and  $c_{2,5} \neq c_{4,5}$  then
18     $E \leftarrow \mathcal{E}_{3,e.1}$ 
19  else if  $c_{2,5} = c_{3,5}$  and  $c_{2,5} = c_{4,5}$  and  $c_{2,6} = c_{3,6}$  then
20     $E \leftarrow \mathcal{E}_{3,e.1}$ 
21  else if  $c_{2,5} = c_{3,5}$  and  $c_{2,5} = c_{4,5}$  and  $c_{2,6} \neq c_{3,6}$  then
22     $E \leftarrow \mathcal{E}_{3,e.2}$ 

```

Figure C.4: Meta-code for INITIALIZATION3.

Figure C.5: The adjacency pattern and the corresponding configuration when $s_3 = 0$, $s_2 \geq 3$ and $s_1 \geq 4$.

C.4 When $s_1 \geq 4$

C.4.1 When $s_2 \geq 3$

The configuration is unique. The initial state of E is \mathcal{E}_4 .

$$\mathcal{E}_4 = \left[\begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & \cdots & s_2 + 2 & s_2 + 3 & \cdots & s_1 + 1 & s_1 + 2 & \cdots & L_G \\ \hline 1 & 1 & 2 & 1 & \cdots & 1 & 2 & \cdots & 2 & 0 & \cdots & 0 \\ 2 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right]$$

```

 $E_{2 \times L_G} \leftarrow \text{INITIALIZATION4}(C, s_1, s_2, s_3)$ 
1  if  $s_2 \geq 3$  then
2     $E \leftarrow \mathcal{E}_4$ 
3  else if  $s_2 = 0$  or  $(s_2 = 1$  and  $c_{3,4} = 1$  and  $c_{3,5} = 1$  and  $c_{4,5} = 1)$  then
4     $E \leftarrow \mathcal{E}_{4,a.4}$ 
5  else if  $s_2 = 1$  and  $c_{3,4} = 0$  and  $c_{4,5} = 1$  then
6     $E \leftarrow \mathcal{E}_{4,b.2}$ 
7  else if  $s_2 = 1$  and  $c_{3,4} = 1$  and  $c_{3,5} = 0$  and  $c_{4,5} = 1$  then
8     $E \leftarrow \mathcal{E}_{4,c.2}$ 
9  else if  $s_2 = 2$  and  $c_{4,5} = 1$  then
10    $E \leftarrow \mathcal{E}_{4,d.2}$ 
11 else if  $s_2 = 2$  and  $c_{4,5} = 0$  then
12    $E \leftarrow \mathcal{E}_{4,d.3}$ 

```

Figure C.6: Meta-code for INITIALIZATION4.

C.4.2 When $s_2 \leq 2$

There are 13 forbidden patterns, as shown in Figure C.7, where the links with labels larger than 5 are not displayed. The pattern in Figure C.3 (a) is forbidden, hence the 4 patterns in Figure C.7 (a.1) are also forbidden, where x can be 1 or 0. The pattern of links 1 – 4 in Figure C.7 (a.2-3) is the same as the pattern in Figure C.3 (b), which has a specific configuration. In Figure C.7 (a.2), link 5 is adjacent to link 1 but not 2, then link 5 must be adjacent to link 3, which is not true, hence the 2 patterns in Figure C.7 (a.2) are forbidden. In Figure C.7 (a.3), link 5 is adjacent to link 1 and 3, then link 5 must be adjacent to link 4, which is not true, hence the pattern in Figure C.7 (a.3) is also forbidden. Similarly, based on the patterns in Figure C.3, we can conclude that patterns in Figure C.7 (b.1), (b.3), (c.1), (c.3), (d.1) and (d.4) are also forbidden. Based on the values of entries s_2 , $c_{3,4}$, $c_{3,5}$ and $c_{4,5}$, Algorithm C.6 decides the initial state of E .

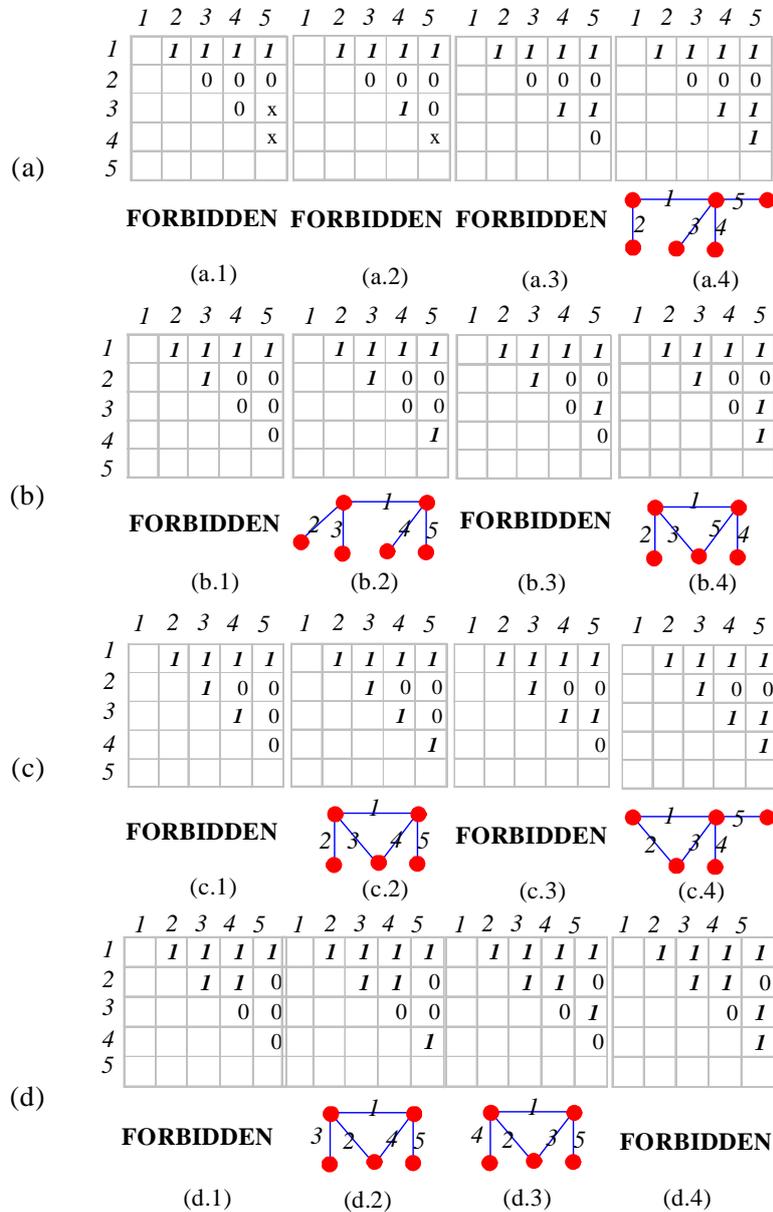


Figure C.7: The adjacency patterns of link 1 and its neighboring links when $s_1 = 4$. There are 16 forbidden patterns. The other 12 possible patterns correspond to only one configuration respectively. The entry x can be 1 or 0.

$$\mathcal{E}_{4,a.4} = \mathcal{E}_{4,c.4} = \left[\begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & \cdots & s_1+1 & s_1+2 & \cdots & L_G \\ \hline 1 & 1 & 2 & 2 & 2 & \cdots & 2 & 0 & \cdots & 0 \\ \hline 2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right]$$

$$\mathcal{E}_{4,b.2} = \mathcal{E}_{4,b.4} = \mathcal{E}_{4,c.2} = \mathcal{E}_{4,d.2} = \left[\begin{array}{ccccccccccc} 1 & 2 & 3 & 4 & 5 & \cdots & s_1+1 & s_1+2 & \cdots & L_G \\ \hline 1 & 1 & 1 & 2 & 2 & \cdots & 2 & 0 & \cdots & 0 \\ \hline 2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right]$$

$$\mathcal{E}_{4,d.3} = \left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & \cdots & s_1+1 & s_1+2 & \cdots & L_G \\ \hline 1 & 1 & 2 & 1 & 2 & \cdots & 2 & 0 & \cdots & 0 \\ \hline 2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right]$$

Appendix D

Proof of the adapted semicircle law

The adapted semicircle law has been presented in Theorem 21.

Proof. For all the graphs,

$$\sum_{j=1}^N \lambda_j = 0$$

which probabilistically means,

$$E[\lambda] = \sum_{k=-\infty}^{\infty} k \Pr[\lambda = k] = 0 \quad (D.1)$$

and

$$\sum_{k=-\infty}^{\infty} \Pr[\lambda = k] = 1 \quad (D.2)$$

according to Perron-Frobenius Theorem, any connected graph has one eigenvalue λ_1 that is the largest, and the multiplicity of λ_1 is 1. Hence, we have,

$$\Pr[\lambda = \lambda_1] = \frac{1}{N} \quad (D.3)$$

and (D.1) can be extended to,

$$E[\lambda] = \lambda_1 \frac{1}{N} + \sum_{\text{All others}} k \Pr[\lambda = k] \quad (D.4)$$

due to (D.2) and (D.3),

$$\sum_{\text{All others}} \Pr[\lambda = k] = 1 - \frac{1}{N}$$

Let us write the semicircle law for graphs with finite N as,

$$f_\lambda(x) = \frac{\sqrt{4Np(1-p) - (x + \varepsilon)^2}}{2\pi Np(1-p)}, \quad |x| \leq 2p(1-p)\sqrt{N} \quad (D.5)$$

given $R = 2p(1-p)\sqrt{N}$,

$$\int_{-R-\varepsilon}^{R-\varepsilon} \frac{\sqrt{4Np(1-p) - (x+\varepsilon)^2}}{2\pi Np(1-p)} dx = 1 - \frac{1}{N} \quad (D.6)$$

Thus (D.4) is written as,

$$\lambda_1 \frac{1}{N} + \int_{\substack{-R-\varepsilon, \\ x \neq \lambda_1}}^{R-\varepsilon} x \frac{\sqrt{4Np(1-p) - (x+\varepsilon)^2}}{2\pi Np(1-p)} dx = 0$$

where the second item on the left is reduced, by using (D.6),

$$\begin{aligned} & \int_{\substack{-R-\varepsilon, \\ x \neq \lambda_1}}^{R-\varepsilon} (x+\varepsilon) \frac{\sqrt{4Np(1-p) - (x+\varepsilon)^2}}{2\pi Np(1-p)} dx - \varepsilon \int_{\substack{-R-\varepsilon, \\ x \neq \lambda_1}}^{R-\varepsilon} \frac{\sqrt{4Np(1-p) - (x+\varepsilon)^2}}{2\pi Np(1-p)} dx \\ &= 0 - \varepsilon \left(1 - \frac{1}{N}\right) \end{aligned}$$

Hence we get, for large N ,

$$\lambda_1 \frac{1}{N} - \varepsilon \left(1 - \frac{1}{N}\right) = 0$$

since $\lambda_1 \approx (N-2)p + 1$,

$$\varepsilon = \frac{\lambda_1}{N-1} = \frac{(N-2)p+1}{N-1} \approx p \quad (D.7)$$

Substituting (D.7) into (D.5), the theorem is proved. ■

Appendix E

Notation

Except that the notation of ILIGRA has been described in Table 5.1, this thesis follows the notation outlined below.

$G(N, L)$	A simple graph with N nodes and L links
\mathcal{N}	The set of nodes in a graph
\mathcal{L}	The set of links in a graph
A	The adjacency matrix of a graph
J	All-one matrix
u	All-one vector
λ	A eigenvalue of adjacency matrix A
Λ	$= \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$: Diagonal matrix of the adjacency eigenvalues
d_j	Degree of node j
Δ	$= \text{diag}(d_1, d_2, \dots, d_N)$: Diagonal matrix of the nodal degrees
$h(x)$	Heavyside's step function
θ	Reconstructability coefficient
$G_p(N)$	Erdős-Rényi random graph with N nodes and link probability p
$G_{ER}(N, L)$	Erdős-Rényi random graph with N nodes and L links
K_N	The complete graph with N nodes
$K_{n,m}$	The complete bipartite graph with $N = n + m$
$\Pr[X]$	Probability of the event X
$E[X]$	Expectation of the random variable X
$f_X(x)$	Probability density function of X
E_G	$= \sum_{j=1}^N \lambda_j $: The energy of a graph G with N nodes

m_0	The multiplicity of adjacency eigenvalue 0
m_{-1}	The multiplicity of adjacency eigenvalue -1
C	The link adjacency matrix (LAM) of a graph
$l(G)$	The line graph of graph G
$H(N, M)$	A Hypergraph H with N nodes and M hyperedges

Bibliography

- [1] P. J. Flory, “Molecular size distribution in three-dimensional polymers. i. gelation.” *Journal of the American Chemical Society*, vol. 63, pp. 3083–3090, 1941.
- [2] A. Rapoport, “Nets with distance bias,” *Bulletin of Mathematical Biophysics*, vol. 13, pp. 85–91, 1951.
- [3] —, “Spread of information through a population with sociostructural bias: I. assumption of transitivity.” *Bulletin of Mathematical Biophysics*, vol. 15, pp. 523–533, 1953.
- [4] —, “Contribution to the theory of random and biased nets,” *Bulletin of Mathematical Biophysics*, vol. 19, pp. 257–277, 1957.
- [5] P. Erdős and A. Rényi, “On random graphs,” *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.
- [6] —, “On the evolution of random graphs,” *Publ. Math. Inst. Hungar. Acad. Sci.*, vol. 5, pp. 17–61, 1960.
- [7] —, “On the strenght of connectedness of a random graph,” *Acta Mathematica Scientia Hungary*, vol. 12, pp. 261–267, 1961.
- [8] M. Chavez, M. Valencia, V. Latora, and J. Martinerie, “Complex networks: new trends for the analysis of brain connectivity,” *arXiv*, Feb. 2010.
- [9] M. Boersma, D. J. Smit, H. M. de Bie, and et al., “Network analysis of resting state eeg in the developing young brain: Structure comes with maturation,” *Human Brain Mapping*, 2010.
- [10] D. S. Bassett and E. Bullmore, “Small-world brain networks,” *The Neuroscientist*, vol. 12, no. 6, pp. 512–523, 2006.
- [11] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, “Random graph models of social networks,” *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 2566–2572, 2002.

- [12] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On powerlaw relationships of the internet topology,” in *in Proc. ACM SIGCOMM, August/September, 1999*, pp. 251–262.
- [13] L. A. Adamic and B. A. Huberman, “Power-law distribution of the world wide web,” *SCIENCE*, vol. 287, p. 2115, 2000.
- [14] R. Albert and A. L. Barabási, “Statistical mechanics of complex networks,” *Reviews of Modern Physics*, vol. 74, pp. 47–97, 2002.
- [15] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small -world’ networks,” *Nature*, vol. 393, pp. 440–442, 1998.
- [16] A. L. Barabási and R. Albert, “Emergence of scaling in the random networks,” *Science*, vol. 286, pp. 509–512, 1999.
- [17] M. E. J. Newman, “Assortative mixing in networks,” *Phys. Rev. Lett.*, vol. 89, p. 208701, Oct 2002.
- [18] ———, “Mixing patterns in networks,” *Phys. Rev. E*, vol. 67, p. 026126, 2003.
- [19] M. Girvan and M. Newman, “Community structure in social and biological networks,” *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [20] W. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, no. 2, p. 026113, 2004.
- [21] P. Van Mieghem, “Robustness of large networks,” *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, 2005.
- [22] A. H. Dekker and B. D. Colbert, “Network robustness and graph topology,” *The 27th Australasian Computer Science Conference*, 2004.
- [23] P. Van Mieghem, *Performance Analysis of Communications Networks and Systems*. Cambridge University Press, 2006.
- [24] S. N. Dorogovtsev, A. V. Goltsev, J. F. F. Mendes, and A. N. Samukhin, “Spectra of complex networks,” *Phys. Rev. E*, vol. 68, no. 4, p. 046109, 2003.
- [25] B. Simon, “Fifty years of eigenvalue perturbation theory,” *Bulletin (New Series) of The American Mathematical Society*, vol. 24, no. 2, pp. 303–319, 1991.
- [26] K. V. Ngo, “An approach of eigenvalue perturbation theory,” *Appl. Num. Anal. Math.*, vol. 2, no. 1, pp. 108–125, 2005.

- [27] N. Kambhatla, S. Haykin, and R. D. Dony, "Image compression using klt, wavelets and an adaptive mixture of principal components model," *Journal of VLSI Signal Processing Systems*, vol. 18, no. 3, pp. 287–296, 1998.
- [28] L. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, pp. 167–242, 2007.
- [29] I. Gutman, G. Stevanovic, S. Radenkovic, S. Milosavljevic, and N. Cmiljanovic, "Dependence of the π -electron energy on a large number of non-bonding molecular orbitals," *J. Serb. Chem. Soc.*, vol. 69, no. 10, pp. 777–782, 2004.
- [30] J. Day and W. So, "Graph energy change due to edge deletion," *Linear Algebra and its Applications*, vol. 428, no. 8-9, pp. 2070–2078, 2008.
- [31] R. Balakrishnan, "The energy of a graph," *Linear Algebra and its Applications*, vol. 387, pp. 287–295, 2004.
- [32] M. Lazić and Kragujevac, "On the laplacian energy of a graph," *Czechoslovak Mathematical Journal*, vol. 56, no. 131, pp. 1207–1213, 2006.
- [33] I. Gutman, "The energy of a graph: old and new results," in *Algebraic Combinatorics and Applications*, A. Betten, A. Kohner, R. Laue, and A. Wassermann, eds., pp. 196–211, 2001.
- [34] P. Van Mieghem, *Graph Spectra for Complex Networks*. Cambridge University Press (Cambridge, U.K.), 2011.
- [35] H. Whitney, "Congruent graphs and the connectivity of graphs," *American Journal of Mathematics*, vol. 54, pp. 150–168, 1932.
- [36] F. Harary, *Graph Theory*. Addison-Wesley, Reading, 1969.
- [37] D. Cvetković, P. Rowlinson, and S. Simić, *Spectral Generalizations of Line Graphs*. Cambridge, UK.: Cambridge University Press, 2004.
- [38] J. C. Nacher, U. Ueda, T. Yamada, M. Kanehisa, and T. Akutsu, "Line graphs as social networks," *BMC Bioinformatics*, vol. 24, no. 207, pp. 2611–2618, 2004.
- [39] J. C. Nacher, T. Yamada, S. Goto, M. Kanehisa, and T. Akutsu, "Two complementary representations of a scale-free network," *Physica A*, vol. 349, pp. 349–363, 2005.
- [40] A. Manka-Krason, A. Mwijage, and K. Kulakowski, "Clustering in random line graphs," *Computer Physics Communications*, vol. 181, no. 1, pp. 118–121, 2010.

- [41] A. Manka-Krason and K. Kulakowski, “Assortativity in random line graphs,” *Acta Physica Polonica B Proceedings Supplement*, vol. 3, no. 2, pp. 259–266, 2010.
- [42] M. J. Krawczyk, L. Muchnik, A. Manka-Krason, and K. Kulakowski, “Line graphs as social networks,” *Physica A*, vol. 390, pp. 2611–2618, 2011.
- [43] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multiscale complexity in networks,” *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.
- [44] T. S. Evans and R. Lambiotte, “Line graphs, link partitions, and overlapping communities,” *Phys. Rev. E*, vol. 80, no. 1, p. 016105, 2009.
- [45] T. Evans and R. Lambiotte, “Overlapping communities, link partitions and line graphs,” *Proceedings of the European Conference on Complex Systems*, 2009.
- [46] J. C. Wierman, D. P. Naor, and J. Smalley, “Incorporating variability into an approximation formula for bond percolation thresholds of planar periodic lattices,” *Phys. Rev. E*, vol. 75, no. 1, p. 011114, 2007.
- [47] A. van Rooij and H. Wilf, “The interchange graphs of a finite graph,” *Acta Math. Acad. Sci. Hungar.*, vol. 16, pp. 263–269, 1965.
- [48] L. W. Beineke, “Characterizations of derived graphs,” *Journal of Combinatorial Theory*, vol. 9, pp. 129–135, 1970.
- [49] N. D. Roussopoulos, “A max {m,n} algorithm for determining the graph h from its line graph g,” *Information Processing Letters*, vol. 2, no. 4, pp. 108–112, 1973.
- [50] P. G. H. Lehot, “An optimal algorithm to detect a line graph and output its root graph,” *Journal of ACM*, vol. 21, no. 4, pp. 569–575, 1974.
- [51] J. Krausz, “Démonstration nouvelle d’une théorème de Whitney sur les réseaux,” *Mat. Fiz. Lapok*, vol. 50, pp. 75–85, 1943.
- [52] Vismara, “Finding maximum common connected subgraphs using clique detection or constraint satisfaction algorithms,” in *MCO’08: Modelling, Computation and Optimization in Information Systems and Management Sciences*, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2008, pp. 364–374.
- [53] J. Naor and M. B. Novick, “An efficient reconstruction of a graph from its line graph in parallel,” *Journal of Algorithms*, vol. 11, no. 1, pp. 132–143, 1990.
- [54] S. Simić, “An algorithm to recognize a generalized line graphs and output its root graph,” *Publ. Math. Inst. (Beograd)*, vol. 49, no. 63, pp. 21–26, 1990.

- [55] D. G. Degiorgi and K. Simon, “A dynamic algorithm for line graph recognition,” in *WG '95: Proceedings of the 21st International Workshop on Graph-Theoretic Concepts in Computer Science*, 1995, pp. 37–48.
- [56] O. Ore, *Theory of Graphs*. American Mathematical Society Colloquium Publications, 1962, vol. 21.
- [57] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, “Complex networks: Structure and dynamics,” *Physics Reports*, vol. 424, pp. 175–308, 2006.
- [58] P. Van Mieghem, H. Wang, X. Ge, S. Tang, and F. A. Kuipers, “Influence of assortativity and degree-preserving rewiring on the spectra of networks,” *The European Physical Journal B - Condensed Matter and Complex Systems*, pp. 643–652, July 2010.
- [59] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, July 2010.
- [60] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Phys. Rev. E*, vol. 435, no. 7043, pp. 814–818, 2005.
- [61] A. Mcdaid and N. Hurley, “Detecting highly overlapping communities with model-based overlapping seed expansion,” *ASONAM 2010*, 2010.
- [62] P. Poller, G. Palla, and T. Vicsek, “Preferential attachment of communities: the same principle, but a higher level,” *Europhys. Lett.*, vol. 73, no. 3, pp. 478–484, 2006.
- [63] B. Skyrms and R. Pemantle, “A dynamic model of social network formation,” *Proc. Natl. Acad. Sci.*, vol. 97, pp. 9340–9346, 2000.
- [64] R. Toivonen, J.-P. Onnela, J. Saramki, J. Hyvnen, and K. Kaski, “A model for social networks,” *Physica A: Statistical and Theoretical Physics*, vol. 371, no. 2, pp. 851–860, 2006.
- [65] B. Bollobás, *Random graphs*, 2nd ed. Cambridge University Press, 2001.
- [66] F. Chung and L. Lu, “Connected components in random graphs with given expected degree sequences,” *Annals of Combinatorics*, vol. 6, pp. 125–145, 2002.
- [67] R. van der Hofstad, “Random graphs and complex networks,” available at <http://www.win.tue.nl/~rhofstad/NotesRGCN2009.pdf>.

- [68] B. Bollobás and Y. Leong, “Graphs extremal and random,” *Newsletter of Institute for Mathematical Sciences*, vol. 11, pp. 14–21, 2007.
- [69] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabasi, “Hierarchical Organization of Modularity in Metabolic Networks,” *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.
- [70] S. Sinha, “Complexity vs. stability in small-world networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 346, no. 1-2, pp. 147–153, 2005.
- [71] J. D. Cohen and F. Tong, “Neuroscience: The face of controversy,” *Science*, vol. 293, no. 5539, pp. 2405–2407, 2001.
- [72] S. Jalan and J. N. Bandyopadhyay, “Randomness of random networks: A random matrix analysis,” *EPL (Europhysics Letters)*, vol. 87, no. 4, p. 48010, 2009.
- [73] X. Ying and X. Wu, “On randomness measures for social networks,” in *SDM*. SIAM, 2009, pp. 709–720.
- [74] W. de Haan, Y. A. Pijnenburg, R. L. Strijers, Y. van der Made, W. M. van der Flier, P. Scheltens, and C. J. Stam, “Functional neural network analysis in frontotemporal dementia and alzheimer’s disease using eeg and graph theory,” *BMC Neuroscience*, 2009.
- [75] E. P. Wigner, “Characteristic vectors of bordered matrices with infinite dimensions,” *Annals of Mathematics*, vol. 62, no. 3, pp. 548–564, November 1955.
- [76] —, “Characteristic vectors of bordered matrices with infinite dimensions ii,” *Annals of Mathematics*, vol. 65, no. 2, pp. 203–207, March 1957.
- [77] —, “On the distribution of the roots of certain symmetric matrices,” *Annals of Mathematics*, vol. 67, no. 2, pp. 325–327, March 1958.
- [78] W. Weibull, “A statistical distribution function of wide applicability,” *J. Appl. Mech.-Trans. ASME*, vol. 18, no. 3, pp. 293–297, 1951.
- [79] M. Deijfen, H. van den Esker, R. van der Hofstad, and G. Hooghiemstra, “A preferential attachment model with random initial degrees,” *Arkiv för Matematik*, vol. 47, pp. 41–72, 2009.
- [80] M. E. J. Newman, “The structure and function of complex networks,” *SIAM Review*, vol. 45, pp. 167–256, 2003.
- [81] P. Van Mieghem, H. Wang, X. Ge, S. Tang, and F. A. Kuipers, “Influence of assortativity and degree-preserving rewiring on the spectra of networks,” *The European Physical Journal B - Condensed Matter and Complex Systems*, pp. 643–652, 2010.

- [82] K. Mehlhorn and S. Näher, *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.
- [83] A. L. Cauchy, *Cours d'analyse de l'Ecole Royale Polytechnique Vol. 3 (1821)*. Cambridge University Press, 2009.
- [84] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*. Cambridge University Press, Feb. 1988.
- [85] P. Van Mieghem, J. S. Omic, and R. E. Kooij, "Virus spread in networks," *IEEE/ACM Transaction on Networking*, vol. 17, no. 1, pp. 1–14, 2009.
- [86] P. Van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. van de Bovenkamp, D. Liu, and H. Wang, "Decreasing the spectral radius of a graph by link removals," *Phys. Rev. E*, vol. 84, p. 016101, Jul 2011.
- [87] J. G. Restrepo, E. Ott, and B. R. Hunt, "Onset of synchronization in large networks of coupled oscillators," *Phys. Rev. E*, vol. 71, p. 036151, Mar 2005.
- [88] H. Wang, W. Winterbach, and P. Van Mieghem, "Assortativity of complementary graphs," *The European Physical Journal B*, vol. 83, no. 2, pp. 203–214, 2011.
- [89] E. F. Schubert, *Doping in III-V Semiconductors*. Cambridge, UK.: Cambridge University Press, 2005.
- [90] E. R. van Dam and W. H. Haemers, "Which graphs are determined by their spectra?" *Linear Algebra and its Applications*, vol. 373, pp. 241–272, 2003.
- [91] J. Scott, *Social Network Analysis: A Handbook*. London, UK.: SAGE Publications Ltd, 1991.
- [92] M. Newman, S. Strogatz, and D. Watts, "Random graph with arbitrary degree distribution and their applications," *Phys. Rev. E*, vol. 64, p. 026118, 2001.
- [93] S. Lattanzi and D. Sivakumar, "Affiliation networks," in *Proceedings of the 41st annual ACM symposium on Theory of computing*, 2009, pp. 427–434.
- [94] D. Cvetković, P. Rowlinson, and Simić, "Eigenvalue bounds for the signless laplacians," *Publ. Inst. Math. (Beograd)*, vol. 81, no. 95, pp. 11–17, 2007.
- [95] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)*, vol. 1, no. 1, pp. 35–49, 2007.

- [96] C. J. Stam and B. van Dijk, “Synchronization likelihood: an unbiased measure of generalized synchronization in multivariate data sets,” *Physica D*, vol. 163, pp. 236–251, 2002.
- [97] C. J. Stam, B. F. Jones, G. Nolte, M. Breakspear, and P. Scheltens, “Small-world networks and functional connectivity in alzheimer’s disease,” *Cerebral Cortex*, vol. 17, no. 1, pp. 92–99, 2007.
- [98] K. Supekar, V. Menon, D. Rubin, M. Musen, and M. D. Greicius, “Network analysis of intrinsic functional brain connectivity in alzheimer’s disease,” *PLoS Computational Biology*, vol. 4, no. 6, p. e1000100, 2008.

Samenvatting (Summary in Dutch)

Infrastructuurnetwerken, zoals het internet, telecommunicatienetwerken, energienetwerken, transportnetwerken (spoor, weg, waterwegen en luchtvaartnetwerken), gasttransportnetwerken en waternetwerken worden steeds complexer. Deze complexe infrastructuurnetwerken zijn cruciaal voor onze maatschappij. Het robuuster maken en optimaliseren van de prestaties van complexe infrastructuurnetwerken is een actief onderzoeksgebied. Naast de door de mens ontworpen infrastructuurnetwerken komen complexe netwerken ook veel voor in de natuur, voorbeelden hiervan zijn sociale-netwerken, ecologische netwerken en biologische netwerken. Om vat te krijgen op sociale en ecologische problemen en onopgeloste medische vragen moeten we begrijpen hoe complexe netwerken georganiseerd zijn, werken en functioneren.

Complexe netwerken kunnen gerepresenteerd worden als grafen. Een graaf bestaat uit een verzameling knopen en zijden die de knopen verbinden. Een graaf wordt uniek beschreven door zijn verbindingsmatrix, waarvan een element in rij i en kolom j een is als er een zijde bestaat tussen knoop i en j en anders nul. Elke verbindingsmatrix heeft een unieke set eigenwaarden en bijbehorende eigenvectoren. De eigenwaarden en eigenvectoren, ook wel het spectrum van een graaf genoemd, bevat alle informatie over een graaf. De topologische/fysische betekenis van enkele eigenwaarden en eigenvectoren is al bekend. Kennis over het spectrum van grafen is van cruciaal belang in de vele facetten van onderzoek naar complexe netwerken, zoals connectiviteit en de verspreiding van virussen in een netwerk. De lijngraaf $l(G)$ van graaf G bestaat uit een verzameling knopen die corresponderen met de zijden in G . Twee knopen in $l(G)$ zijn verbonden als de corresponderende zijden in G een knoop gemeen hebben. Sommige problemen op grafen kunnen getransformeerd worden in eenvoudigere problemen in het domein van lijngraphen. Zo kunnen, bijvoorbeeld, knopen gescheiden worden om overlappende gemeenschappen te vinden door zijden te scheiden in de lijngraaf van de betreffende graaf. Bovendien hebben lijngraphen vaak eigenschappen gemeen met wereldecchte complexe netwerken, zoals een sterke groepering en assortative mixing. Lijngraphen worden daarom door vele beschouwd als modellen van wereldecchte complexe netwerken

Robuustheid en optimalisatie van complexe netwerken is een tamelijke breed onderzoeksgebied. Wij beperken ons daarom tot het reconstrueren van complexe netwerken vanuit het spectrale domein en het domein van lijngraphen. Dit proefschrift is als volgt

georganiseerd. We onderzoeken eerst het reconstrueren van netwerken van eigenwaarden en eigenvectoren en de spectrale eigenschappen van netwerken. In het tweede deel van dit proefschrift introduceren we twee algoritmes om een graaf te reconstrueren vanuit het lijngraaf domein, de eigenschappen van de lijngraaf, en een random lijngraaf model. Tenslotte presenteren we onze onderzoeksresultaten van twee wereldecchte netwerken.

Een verbindingsmatrix kan uit de bijbehorende eigenwaarden en eigenvectoren berekend worden. Ook als sommige van de eigenwaarden nul gemaakt worden, kan de verbindingsmatrix nog correct bekerend worden. We introduceren de reconstructie coëfficiënt, gedefinieerd als het maximum aantal eigenwaarden die verwijderd kunnen worden. Wij hebben ontdekt dat de reconstructie coëfficiënt een lineaire functie is van de netwerk grootte. Ook onderzoeken we de verhouding tussen de graaf energie en de assortativiteit in verschillende netwerk types.

Wij introduceren twee algoritmes om een netwerk te reconstrueren van een lijngraaf: MARINLINGA en ILIGRA. In tegenstelling tot eerdere algoritmes, maakt MARINLINGA geen gebruik van Whitney's theorema, maar van het hernoemen van zijden en 'endnode recognition'. ILIGRA reconstrueert grafen met een lineaire tijd complexiteit. Dit proefschrift breidt onderzoek in het lijngraaf domein uit. We hebben ontdekt dat het aantal zijden in een lijngraaf met een vast aantal knopen niet in een bepaald interval kan liggen. Dit interval is het bandgap van de lijngraaf. We geven de exacte uitdrukking voor de banden en bandgaps in het aantal zijden in lijngrafen. Om onderzoek in lijngrafen te bevorderen stellen we een model voor om willekeurige lijngrafen te genereren. De kern van ons model is om een paar knopen stap voor stap tot klieken samen te voegen met inachtneming van enkele regels die ervoor zorgen dat de resulterende graaf een lijngraaf is. Dankzij dit model hebben we een methode ontdekt om een reeks lijngrafen te genereren waarvan de assortativiteit lineair toeneemt.

In dit proefschrift worden twee soorten wereldecchte grafen onderzocht: sociale netwerken en menselijke hersen netwerken. We karakteriseren de overlappende gemeenschapsstructuur in sociale netwerken gevormd door ArXiv coauteurs, medespelers in IMDB en SourceForge samenwerking, en stellen een op preferential attachment gebaseerd groeiend hypergraaf model voor. Dit hypergraaf model beschrijft de fundamentele eigenschappen van wereldecchte verwantschapsnetwerken waaronder de power-law verdeling van groeps-grootte, groepsgraad, overlappingsdiepte, individuele graad en het aantal gedeelde interesses, en reproduceert eigenschappen van sociale netwerken zoals sterke groepering, assortative mixing en een korte gemiddelde pad lengte. We stellen een spectrale willekeurighedsmaat voor om uitdrukking te geven aan de mate van willekeur in netwerken in het onderzoek naar hersen netwerken. We hebben door middel van het meten van de mate van willekeur ontdekt dat de hersen netwerken van mensen die aan Alzheimer lijden statistisch een grotere mate van willekeur vertonen dan de hersen netwerken van gezonde mensen.

Acknowledgements

I owe my deepest and sincere gratitude to Prof. Piet Van Mieghem. He accepted me as a student to pursue Ph.D. degree in the group of Network Architectures and Services (NAS). Prof. Piet Van Mieghem has always been an inspiring and encouraging supervisor and promoter. During the first year of my Ph.D., he kept me on the right track when the researches were going towards an unexpected direction. Without his effective guidance and word-to-punctuation correction, it would not be possible for me to publish my first paper. Prof. Piet Van Mieghem impressed me with his passion and an serious attitude on science. Although he sets a high-standard for Ph.D. students and has been very strict with it, Prof. Piet Van Mieghem has tolerated me at the points when I appeared stressed out and not very productive. He also taught me the oral and writing communication skills. I thank him particularly for his help in structuring this Ph.D. thesis.

I would like to thank Dr. Huijuan Wang for her advice and collaboration on the reconstructability and randomness of networks, Stojan Trajanovski for discussion and collaboration on line graphs, and Norbert Blenn for “gold of research”-measurements of real-world networks and collaboration on social networks. My thanks go to Prof. C. J. Stam and Dr. Willem de Haan from VU Medical Center for the collaboration on brain networks. I am grateful to Prof. Rob Kooij, Dr. Fernando Kuipers, Dr. Christian Doerr, Wynand Winterbach, and Javier Martín-Hernández for many valuable discussions. Ruud van de bovenkamp translated the summary of this thesis into Dutch, and Niels van Adrichem translated the propositions coming along with the thesis into Dutch. Thank you all! I thank all the kind NAS members for the pleasant working atmosphere, and I will remember each smiling face and each joyful laughter. My gratitude also goes to the European project, Next Generation Infrastructure, which funded my four-year researches.

I thank my parents for their unconditional love, the love I feel most, the love that cares me most and comforts me most, and the love I cannot never equally give back to them.

Finally, I give all my love to Qin and Kaesy.

Curriculum Vitae

Dajie Liu was born in Suqian, Jiangsu Province, China, on March 21st, 1985. He obtained a B.Sc. degree (cum laude) in Electronic Science and Engineering at Southeast University, Nanjing, China, in 2006. He received a M.Sc. (cum laude) degree in Optical Communications and Photonic Technologies at Polytechnic University of Turin, Turin, Italy, in 2008, and another M.Sc. degree in Optical Access Network Management at Southeast University, Nanjing, China, in 2009. He became a Ph.D. student in the group of Network Architectures and Services, faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands, in 2009. Under the supervision of Prof. Piet Van Mieghem, he performed researches in the spectral analyses of complex networks, and the analysis and modeling of social networks and biological networks.

During his Ph.D. period, he assisted in lecturing the course “Performance Analysis” and he served as a reviewer for the conferences, IFIP Networking 2011 and International Teletraffic Congress 2011, and the journal, IEEE/ACM Transactions on Networking. He won the travelling and lodging grant for the Conference on Applications of Graph Spectra in Computer Science (CAGSCS 2012), July 16-20, 2012.

Publications by the author

Refereed Papers

- Dajie Liu, Stojan Tranovski, and Piet Van Mieghem, “ILIGRA: An Efficient Inverse Line Graph Algorithm”, submitted to Journal of Mathematical Modeling and Applications.
- Dajie Liu, Norbert Blenn, and Piet Van Mieghem, “Characterizing and Modeling Social Networks with Overlapping Communities”, International Journal of Web Based Communities (IJWBC), Vol. 9, No. 3, to appear, 2013.
- Dajie Liu, Stojan Tranovski, and Piet Van Mieghem, “Random Line Graphs and A Linear Law for The Assortativity”, Physical Review E, Vol. 87, No. 1, January, p. 012816, 2013.
- Dajie Liu, Norbert Blenn, and Piet Van Mieghem, “A Social Network Model Exhibiting Tunable Overlapping Community Structure”, Procedia Computer Science, Vol. 9, pp. 1400-1409, 1st International Workshop on Advances in Computational Social Science, June 4-6, Omaha, Nebraska, USA, 2012.
- Dajie Liu, Norbert Blenn, and Piet Van Mieghem, “Characterizing the Structure of Affiliation Networks”, Procedia Computer Science, Vol. 9, pp. 567-576, 12th International Conference on Computational Science (ICCS), June 4-6, Omaha, Nebraska, USA, 2012.
- Piet Van Mieghem, Dragan Stevanović, Fernando Kuipers, Cong Li, Ruud van de Bovenkamp, Dajie Liu and Huijuan Wang, “Decreasing the spectral radius of a graph by link removals”, Physical Review E, Vol. 84, No. 1, July, p. 016101, 2011.
- Dajie Liu, Huijuan Wang, and Piet Van Mieghem, “Spectral perturbation and reconstructability of complex networks”, Physical Review E, Vol. 81, No. 1, January, p. 016101, 2010.

- Dajie Liu, Ningfeng Bai, and Xiaohan Sun, “Tolerance Analyses for Metal EBG Waveguides”, *Journal of Southeast University for Graduate Students*, Vol. 5, No.1, pp.138-143, 2008. [Not related to this thesis]
- Dajie Liu, Jinpeng Liu, and Xiaohan Sun, “Seat Management System of Library Based on MCU and Body-infrared Switch”, *Engineers of Electronics*, Vol. 32, No.5, pp.73-75, 2006. [Not related to this thesis]

Presentations and Posters

- Dajie Liu, Willem de Haan, Cornelis J. Stam, and Piet Van Mieghem, “Graph Spectra Based Method For Measuring the Network Randomness”, Oral Presentation, Conference on Applications of Graph Spectra in Computer Science (CAGSCS), July 16-20, Barcelona, Spain, 2012.
- Dajie Liu, Norbert Blenn, and Piet Van Mieghem, “Properties and Modeling of Affiliation Networks”, Poster, ICT.OPEN, November 14-15, Veldhoven, The Netherlands, 2011.
- Dajie Liu, Huijuan Wang, and Piet Van Mieghem, “Spectral Perturbation and Reconstructability of Complex Networks”, Poster, Scientific ICT-Research Event Netherlands (SIREN), November 5, Twente, The Netherlands, 2009.

NGInfra PhD Thesis Series

1. Strategic behavior and regulatory styles in the Netherlands energy industry, Martijn Kuit, 2002, Delft University of Technology, the Netherlands.
2. Securing the public interest in electricity generation markets, The myths of the invisible hand and the copper plate, Laurens de Vries, 2004, Delft University of Technology, the Netherlands.
3. Quality of service routing in the internet: theory, complexity and algorithms, Fernando Kuipers, 2004, Delft University of Technology, the Netherlands.
4. The role of power exchanges for the creation of a single European electricity market: market design and market regulation, François Boisseleau, 2004, Delft University of Technology, the Netherlands, and University of Paris IX Dauphine, France.
5. The ecology of metals, Ewoud Verhoef, 2004, Delft University of Technology, the Netherlands.
6. MEDUSA, Survivable information security in critical infrastructures, Semir Daskapan, 2005, Delft University of Technology, the Netherlands.
7. Transport infrastructure slot allocation, Kaspar Koolstra, 2005, Delft University of Technology, the Netherlands.
8. Understanding open source communities: an organizational perspective, Ruben van Wendel de Joode, 2005, Delft University of Technology, the Netherlands.
9. Regulating beyond price, integrated price-quality regulation for electricity distribution networks, Viren Ajodhia, 2006, Delft University of Technology, the Netherlands.
10. Networked Reliability, Institutional fragmentation and the reliability of service provision in critical infrastructures, Mark de Bruijne, 2006, Delft University of Technology, the Netherlands.

11. Regional regulation as a new form of telecom sector governance: the interactions with technological socio-economic systems and market performance, Andrew Barendse, 2006, Delft University of Technology, the Netherlands.
12. The Internet bubble - the impact on the development path of the telecommunications sector, Wolter Lemstra, 2006, Delft University of Technology, the Netherlands.
13. Multi-agent model predictive control with applications to power networks, Rudy Negenborn, 2007, Delft University of Technology, the Netherlands.
14. Dynamic bi-level optimal toll design approach for dynamic traffic networks, Dusica Joksimovic, 2007, Delft University of Technology, the Netherlands.
15. Intertwining uncertainty analysis and decision-making about drinking water infrastructure, Machtelt Meijer, 2007, Delft University of Technology, the Netherlands.
16. The new EU approach to sector regulation in the network infrastructure industries, Richard Cawley, 2007, Delft University of Technology, the Netherlands.
17. A functional legal design for reliable electricity supply, How technology affects law, Hamilcar Knops, 2008, Delft University of Technology, the Netherlands and Leiden University, the Netherlands.
18. Improving real-time train dispatching: models, algorithms and applications, Andrea D'Ariano, 2008, Delft University of Technology, the Netherlands.
19. Exploratory modeling and analysis: A promising method to deal with deep uncertainty, Datu Buyung Agusdinata, 2008, Delft University of Technology, the Netherlands.
20. Characterization of complex networks: application to robustness analysis, Almerima Jamaković, 2008, Delft University of Technology, Delft, the Netherlands.
21. Shedding light on the black hole, The roll-out of broadband access networks by private operators, Marieke Fijnvandraat, 2008, Delft University of Technology, Delft, the Netherlands.
22. On stackelberg and inverse stackelberg games and their applications in the optimal toll design problem, the energy markets liberalization problem, and in the theory of incentives, Kateřina Staňková, 2009, Delft University of Technology, Delft, the Netherlands.

23. On the conceptual design of large-scale process and energy infrastructure systems: integrating flexibility, reliability, availability, maintainability and economics (FRAME) performance metrics, Austine Ajah, 2009, Delft University of Technology, Delft, the Netherlands.
24. Comprehensive models for security analysis of critical infrastructure as complex systems, Fei Xue, 2009, Politecnico di Torino, Torino, Italy.
25. Towards a single European electricity market, A structured approach for regulatory mode decision-making, Hanneke de Jong, 2009, Delft University of Technology, the Netherlands.
26. Co-evolutionary process for modeling large scale socio-technical systems evolution, Igor Nikolić, 2009, Delft University of Technology, the Netherlands.
27. Regulation in splendid isolation: A framework to promote effective and efficient performance of the electricity industry in small isolated monopoly systems, Steven Martina, 2009, Delft University of Technology, the Netherlands.
28. Reliability-based dynamic network design with stochastic networks, Hao Li, 2009, Delft University of Technology, the Netherlands.
29. Competing public values, Bauke Steenhuisen, 2009, Delft University of Technology, the Netherlands.
30. Innovative contracting practices in the road sector: cross-national lessons in dealing with opportunistic behaviour, Mónica Altamirano, 2009, Delft University of Technology, the Netherlands.
31. Reliability in urban public transport network assessment and design, Shahram Tahmasseby, 2009, Delft University of Technology, the Netherlands.
32. Capturing socio-technical systems with agent-based modelling, Koen van Dam, 2009, Delft University of Technology, the Netherlands.
33. Road incidents and network dynamics, Effects on driving behaviour and traffic congestion, Victor Knoop, 2009, Delft University of Technology, the Netherlands.
34. Governing mobile service innovation in co-evolving value networks, Mark de Reuver, 2009, Delft University of Technology, the Netherlands.
35. Modelling risk control measures in railways, Jaap van den Top, 2009, Delft University of Technology, the Netherlands.

36. Smart heat and power: Utilizing the flexibility of micro cogeneration, Michiel Houwing, 2010, Delft University of Technology, the Netherlands.
37. Architecture-driven integration of modeling languages for the design of software-intensive systems, Michel dos Santos Soares, 2010, Delft University of Technology, the Netherlands.
38. Modernization of electricity networks: Exploring the interrelations between institutions and technology, Martijn Jonker, 2010, Delft University of Technology, the Netherlands.
39. Experiencing complexity: A gaming approach for understanding infrastructure, Geertje Bekebrede, 2010, Delft University of Technology, the Netherlands.
40. Epidemics in Networks: Modeling, Optimization and Security Games. Technology, Jasmina Omić, 2010, Delft University of Technology, the Netherlands.
41. Designing Robust Road Networks: A general method applied to the Netherlands, Maaïke Snelder, 2010, Delft University of Technology, the Netherlands.
42. Simulating Energy Transitions, Emile Chappin, 2011, Delft University of Technology, the Netherlands.
43. De ingeslagen weg. Een dynamisch onderzoek naar de dynamiek van de uitbesteding van onderhoud in de civiele infrastructuur, Rob Schoenmaker, 2011, Delft University of Technology, the Netherlands
44. Safety Management and Risk Modelling in Aviation: the challenge of quantifying management influences, Pei-Hui Lin, 2011, Delft University of Technology, the Netherlands.
45. Transportation modelling for large-scale evacuations, Adam J. Pel, 2011 Delft University of Technology, Delft
46. Clearing the road for ISA Implementation? Applying Adaptive Policymaking for the Implementation of Intelligent Speed Adaptation, Jan-Willem van der Pas, 2011, Delft University of Technology, the Netherlands
47. Design and decision-making for multinational electricity balancing markets, Reinier van der Veen, 2012, Delft University of Technology, The Netherlands
48. Understanding socio-technical change. A system-network-agent approach, Catherine Chiong Meza, 2012, Delft University of Technology, the Netherlands

49. National design and multi-national integration of balancing markets, Alireza Ab-basy, 2012, Delft University of Technology, the Netherlands
50. Regulation of Gas Infrastructure Expansion, Jeroen de Joode, 2012, Delft Uni-versity of Technology, The Netherlands
51. Governance Structures of Free/Open Source Software Development. Examining the role of modular product design as a governance mechanism in the FreeBSD Project, George Dafermos, 2012, Delft University of Technology, the Netherlands
52. Making Sense of Open Data - From Raw Data to Actionable Insight, Chris Davis, 2012, Delft University of Technology, the Netherlands
53. Intermodal Barge Transport: Network Design, Nodes and Competitiveness, Rob Konings, 2009, Delft University of Technology, Trail Research School, the Nether-lands
54. Handling Disruptions in Supply Chains: An integrated Framework and an Agent-based Model, Behzad Behdani, 2013, Delft University of Technology, the Nether-lands
55. Images of Cooperation: A Methodological Exploration in Energy Networks, An-dreas Ligtvoet, 2013, Delft University of Technology, the Netherlands
56. Robustness and Optimization of Complex Networks, Reconstructability, Algo-rithms and Modeling, Dajie Liu, 2013, Delft University of Technology, the Nether-lands

Order information: info@nextgenerationinfrastructures.eu