Colouring, Centrality and Core-Periphery Structure in Graphs

M. Puck Rombach

Somerville College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2013

Abstract

Colouring, Centrality and Core-Periphery Structure in Graphs
Michaela Puck Rombach
Somerville College
Doctor of Philosophy in Mathematics
Trinity, 2013

Researchers from a wide range of backgrounds contribute to the field of network science. This study aims to add results to the field from both a pure approach, where the main goal is to produce rigorous results on abstract problems, and an applied approach, where the main goal is to study and model empirical data.

We prove a rigorous result about special colourings of random Erdős-Rényi graphs. A proper vertex colouring is a colouring of the vertices of a graph such that no two adjacent vertices receive the same colour. An equitable colouring of a graph is a proper vertex colouring such that the sizes of any two colour classes differ by at most one. The equitable chromatic number $\chi_{=}(G(n,p))$ of G is defined as the least positive integer k for which there exists an equitable colouring of a graph G with k colours. The equitable chromatic threshold $\chi^{*}_{=}(G(n,p))$ of G is the least positive integer k such that for every $k_0 > k$ there exists an equitable colouring of G with k_0 colours. Krivelevich and Patkós conjectured in 2009 that $\chi(G(n,p)) \sim \chi_{=}(G(n,p)) \sim \chi^{*}_{=}(G(n,p))$ for $C/n < p < 1-\epsilon$, where $\epsilon > 0$. We prove this conjecture for $n^{-1+\epsilon_1} < p < 1-\epsilon_2$ where $\epsilon_1, \epsilon_2 > 0$.

Core-periphery structure and centrality are important notions in the study of empirical networks. We investigate several measures that have been proposed to indicate centrality of nodes in networks, and find examples of networks where they fail to distinguish any of the vertices nodes from one another. We develop a new method to investigate core-periphery structure, which entails identifying densely-connected core nodes and sparsely-connected periphery nodes. Nodes in a core are also reasonably well-connected to periphery nodes. Our method takes cores of different size and shapes into account. The idea for this method was partly motivated by the coexistence of community structure and core-periphery structure, and we provide examples of empirical networks that exhibit such a coexistence.

Finally, we present an experiment and an analysis of empirical networks, functional human brain networks, aimed to both introduce a set of new methods and gain an understanding of the networks. We found that reconfiguration patterns of dynamic communities can be used to classify nodes into a stiff core, a flexible periphery, and a bulk. The separation between this stiff core and flexible periphery changes as a person learns a simple motor skill and, importantly, it is a good predictor of how successful the person is at learning the skill. This temporally defined core-periphery organisation corresponds well with the core-periphery detected by the method that we proposed earlier the static networks created by averaging over the subjects dynamic functional brain networks.

We hope our results contribute to the pure side of network science, the applied side of network science, and the bridge between them.

To Mama, Papa and Saskia.

Acknowledgements

I need to acknowledge many organisations and people for their support. I will start with those who helped me professionally and then acknowledge those who gave me personal support, but a hard separation between those two categories does not exist.

I thank my supervisors, Mason Porter and Alex Scott, for their guidance and support during my studies. They are an unlikely combination in terms of their research interests (see the Introduction for some words on marrying the pure and applied sides of networks), but it has been a perfect set-up for me, and I feel privileged to have worked with two experts in such different fields. My discussions with them have been invaluable in my development as a mathematician.

I thank all of my other co-authors Dani Bassett, Mihai Cucuringu, James Fowler, Scott Grafton, Peter Mucha, Nicholas Wymbs. I thank Alex Arenas, Charlie Brummit, Valentin Danchev, Sergey Dorogovtsev, Andrew Elliott, Martin Everett, Des Higham, Brendan McKay, Melissa Lever, Jan Haerter, Sang Hoon Lee, Christian Lohse, José Mendes, Jim Moody, Alex Pothen, Stan Wasserman, and two anonymous referees for helpful comments and discussions on the papers I was involved in writing.

My DPhil was funded by the James S. McDonnell Foundation (# 220020177), without which none of this would have been possible. The core-periphery work in particular (Chapters 7 and 8) and related travels were funded by the James S. McDonnell Foundation (# 220020177) and the NSF (DMS-0645369) and was carried out in part at the Statistical and Applied Mathematical Sciences Institute in Research Triangle Park, North Carolina. I thank both the Mathematical Institute and Somerville College for providing my other travel grants.

I thank Mark Newman for providing the data for the Zachary Karate Club network and the 2006 network of network scientists, Martin Rosvall for the 2010 network of network scientists data, and Keith Poole and Howard Rosenthal for maintaining the Congressional voting data [227].

The fMRI experiments at the University of California, Santa Barbara were supported by the Sage Center for the Study of the Mind, the Errett Fisher Foundation, the Templeton

Foundation, the David and Lucile Packard Foundation, the Public Health Service Grant NS44393, and the Institute for Collaborative Biotechnologies through Contract W911NF-09-D-0001 from the US Army Research Office. My supervisor, Mason Porter, was also supported by the FET-Proactive project PLEXMATH (FP7-ICT-2011-8; grant # 317614) funded by the European Commission.

During the course of my DPhil I have had many opportunities to visit institutes and people abroad for conferences, summer schools and academic visits, some of which spanned several weeks. I especially want to thank the *Statistical and Applied Mathematical Sciences Institute (SAMSI)* and Peter Mucha for inviting and hosting me in the Research Triangle Park, US, I thank José Mendes and Sergey Dorogovtsev for hosting me at Universidade de Aveiro in Portugal, and I thank James Fowler for hosting me at the University of California, San Diego, US, and for many discussions about centrality.

I thank my office mates from DH9: Martin Gould in particular for sharing much needed wine and whine nights with me, Marya Bazzi, Lucas Jeub, Sewook Oh and Marta Sarzynska for forming an OCIAM networks army with me. I thank the other combinatorics DPhil students: Matthew Ashford, Ross Atkins, Annika Heckel, Anne Hillebrand, Kitty Meeks, Max Pitz, Fiona Skerman, Kerstin Weller and Paul Withers. I will miss you all so much.

I thank Catryn Collins, Emily Hicks, Molly Kavanagh, Tiffany Taylor for reminding me to have fun with maths.

There are many more friends to thank for their support and cheer: Brigitte Augustijn, Ben Bertoldi, Nancy Blaker, Maria Dalzell, Eleanor Denham, Andrew McDowell, Sarah de Haas, Charlotte Hoekstra, Will Hobbs, Charlotte Koot, Henry Martin, Martijn Mos, Anne-Mischa van Schouwenburg, Olga Shvarova, Corwin Wright. You have all meant a great deal to me that this space is too short for to describe. I thank Maria, Stephen and Spike Dalzell for being the best house mates ever and creating a home that I am sad to leave behind.

I thank my family, to whom this text is dedicated, my mother, father and sister, whom I love so much and wish I could take with me when I leave the continent. They have supported me (for my entire life) in every possible way they could. I also thank my grandmothers and grandfather for their pride and support.

Last but not least, I thank my dog Susie, my faithful, fluffy companion.

Statement of originality

The material presented in this thesis has, to the best of my knowledge, never been published or written by another person, except where explicit acknowledgement is made within the text or the Acknowledgements page. Whenever results represent the efforts of a collaboration, it is made clear within the text how much and which parts of the work were done by me and my collaborators, respectively. Some figures have been taken from articles in which I am a co-author, but that were not created by me, and this is explicitly stated in the caption whenever that is the case.

Contents

List of Figures

List of Symbols

The following list contains the symbols used throughout this thesis. Unless explicitly stated otherwise, these are the standard meanings of symbols used in the text. The symbol is in the first column, a short definition on the second column, and the page number of the page where the symbol and definition are first introduced in the third column.

Symbol	**Definition**	**Page**
$\mathbf{A}$	Adjacency matrix of a network.	76
$\mathbf{A}_s$	Adjacency matrix of a network layer s.	88
b	$1/(1-p)$	30
$bc(i)$	Betweenness centrality of i.	94
$\mathcal{B}(n_1, n_2, p)$	Random graph distribution.	16
C	Core matrix.	116
$cc(i)$	Closeness centrality of i.	94
$cm(i)$	Communicability of i.	98
C_n	Cycle on n vertices.	7
$c(i)$	Community assignment of node i.	81
$c_s(i)$	Community assignment of node i in layer s.	88
c_i	Capacity of vertex i.	49

Symbol	Definition	Page
d	np	30
$d(G)$	Diameter of G.	15
$\overline{d}(G)$	Average degree of G	14
$d(i)$	Degree of vertex i.	7
$d^{in}(i)$	In-degree of a vertex i.	22
$d^{out}(i)$	Out-degree of a vertex i.	22
$d(i,j)$	Distance between i and j.	15
$E(G)$	Set of edges of a graph G.	6
$ec(i)$	Eigenvector centrality of i.	97
$\mathbb{E}[X]$	Expectation of a r.v. X.	12
F_N	Network flexibility.	148
$f_{(i,j)}$	Flow along edge (i,j).	49
$f(i)$	Flexibility of node i.	148
$G, G(V,E)$	A graph on vertex set V and edge set E.	6
$\overline{G}$	Complement of a graph G.	31
$\mathcal{G}(n,p)$	Random graph distribution.	11
$\mathcal{G}(n,m)$	Random graph distribution.	11
$\mathcal{G}(n_c,n_p,p,k)$	Random graph distribution.	17
$\mathcal{G}(n,p,r)$.	Random graph distribution.	16
$g_s(i)$	Inter-layer strength of node i in layer s.	88
$I(A)$	Indicator variable of event A.	12
$[k]$	Set of integers $1,\ldots,k$.	31
$kc(i)$	Karma centrality, or self-communicability of i.	98
K_n	Complete graph on n vertices.	7

Symbol	Definition	Page
K_{n_1,n_2}	Complete bipartite graph on n_1, n_2 vertices.	7
$k_s(i)$	Strength of node i within layer s.	88
$\log n$	$\ln n$, or $\log_e n$.	30
m	Number of edges, $\|E(G)\|$, of a graph G.	7
N	Network.	88
n	Number of vertices, $\|V(G)\|$, of a graph G.	7
$O(f(n))$	Asymptotically bounded above by $f(n)$.	8
$o(f(n))$	Dominated by $f(n)$.	8
$\mathbb{P}[A]$	Probability of event A.	12
p, p_G	Edge probability in $G(n,p) \sim \mathcal{G}(n,p)$.	11
P^*_{ij}	Number of shortest paths from i to j.	95
$P^*_{ij}(k)$	Number of shortest paths from i to j containing k.	95
$pr(i)$	PageRank centrality of i.	97
$q(i)$	Contribution of node i to aggregate modularity.	87
$R_{\underline{\gamma}}$	Core quality: $\sum_{i,j} \mathbf{A}(i,j)\mathbf{C}(i,j)(\underline{\gamma})$.	116
s	Chapter 4: Colour class size.	50,60
s	Chapters 5-8: Network layer.	88
$sc(i)$	Subgraph centrality of i.	99
$SP(x,y)$	Set of shortest paths between x and y.	91
s.t.	"Such that"	8
$st(i)$	Stress centrality of i.	95
t	Chapter 8: constant coupling strength.	148
$T(V,E)$	Tree.	7
$t_{r,s}(i)$	Inter-layer coupling of node i between layers r and s.	88

Symbol	Definition	Page
$var(\underline{v})$	Variance of the entries of a vector $\underline{v}$.	102
$V(G)$	Set of vertices of a graph G.	6
W	Weighted adjacency matrix of a network.	76
$w_{i,j}$	Weight of the link between nodes i and j.	76
w_s	Total strength in network layer s.	88
α	Sharpness parameter in the core function.	117
$\alpha(G)$	Independence number of G.	31
β	Core size parameter in the core function.	117
γ	Constant structural resolution parameter.	148
$\underline{\gamma}$	Vector of parameters.	116
$\Gamma(i)$	Neighbourhood of vertex i.	7
$\overline{\Gamma}(i)$	Non-neighbourhood of vertex i.	7
$\Gamma^k(i)$	Set of vertices that are distance k from vertex i.	104
γ_s	Structural resolution parameter of layer s.	88
$\Delta(G)$	Maximum degree of G.	7
$\delta(G)$	Minimum degree of G.	7
$\delta_{i,j}$	Kronecker delta.	81
$\delta_{j,k}(i)$	Pair dependency of j and k on i.	96
$\Theta(f(n))$	Asymptotically bounded below and above by $f(n)$.	8
$\kappa_s(i)$	Multilayer strength of node i.	88
μ	Expectation or mean.	12
μ_N	Total strength in a multilayer network.	88
$\chi(G)$	Chromatic number of G.	29
$\chi_=(G)$	Equitable chromatic number of G.	43

Symbol	Definition	Page
$\chi^*_=(G)$	Equitable chromatic threshold of G.	44
$\Omega(f(n))$	Asymptotically bounded below by $f(n)$.	8
$\omega(f(n))$	Dominates $f(n)$.	8
$\omega(G)$	Clique number of G.	31

Journal Articles Associated with DPhil

Journal articles associated with this DPhil have currently been submitted:

Core-Periphery Structure in Networks, by M. P. Rombach, M. A. Porter, J. H. Fowler and P. J. Mucha [238]. The contents of this paper are discussed in Chapter 7. A revised version has been submitted, after peer-review, to the *Society for Industrial and Applied Mathematics (SIAM) Journal of Applied Mathematics*. Available online: arXiv:1202.2684.

Task-Based Core-Periphery Organization of Human Brain Dynamics, by D. S. Bassett, N. F. Wymbs, M. P. Rombach, M. A. Porter, P. J. Mucha, and S. T. Grafton [20]. The contents of this paper are discussed in Chapter 8. This paper has been accepted and will appear in *PLOS Computational Biology*. Available online: arXiv:1210.3555.

Teach Network Science to Teenagers, by H. A. Harrington, M. Beguerisse Díaz, M. P. Rombach, L. M. Keating, M. A. Porter [143]. This is a paper detailing our efforts to design a program of outreach workshops to teach school students about network science, and we hope that it will help others to organise similar activities. This article has been accepted and will appear in *Network Science*, and is not discussed further in this dissertation. Available online: arXiv:1302.6567.

Journal articles associated with this DPhil that have currently not yet been submitted:

Equitable Colourings of Random Graphs, by M. P. Rombach and A. D. Scott. This work is discussed in Chapter 4.

Discriminating Power of Centrality Measures, by M. P. Rombach and M. A. Porter [237]. This a working paper of which the contents are discussed in Chapter 6 (in particular, Section 6.6). Available online: arXiv:1305.3146.

Chapter 1

Introduction

1.1 Bridges

Many objects of practical and scientific interest can be described as a set of entities and relationships or interactions between those entities: a *network*. For example, Humans and animals form social ties, cities are connected to each other via roads, proteins in our bodies bind to each other to carry out biological functions, charged particles interact with nearby particles and arrange into crystalline structures, and more. The realization that these situations can be studied using similar tools led to the birth of *network science*. The field on its own is young, with few of its breakthroughs pre-dating the 1980s and most of them firmly in the 21st century [211], but it stands on the shoulders of giants.

Networks have been known to mathematicians as *graphs*, and their study dates back at least as far as Euler's famous *Seven Bridges of Königsberg* problem in 1736 [101]. Where network theory typically refers to the entities as *nodes* and the relationships as *links*, graph theory calls them *vertices* and *edges*. As Figure 1.1 illustrates, they describe the same object.[1] These terms are used almost interchangeably. Most of the time, we will use graph-

[1]Figure 1.1 illustrates the simplest definition of a network or graph. There are more general definitions that include multiple edges between the same vertex pair or that link vertices to themselves. Additionally, networks can be directed, weighted, change over time, can include relationships between more than two vertices, etc.

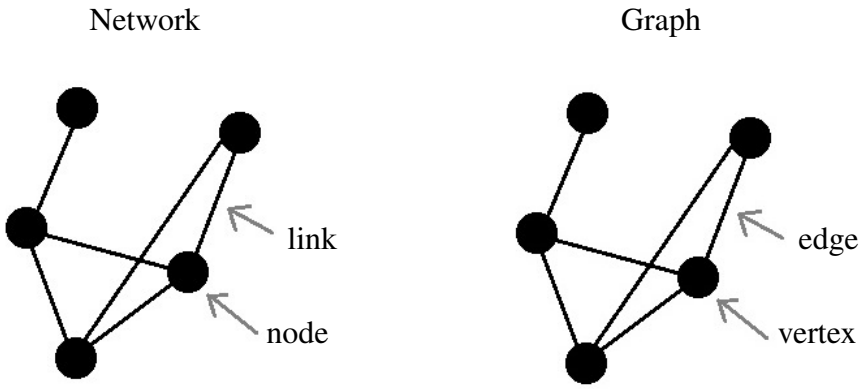


Figure 1.1: Networks and graphs are two words for the same thing.

theoretical terms when we are looking through theoretical glasses, and network-theoretical terms when we are looking through more applied glasses. However, this distinction is not always clear even to the author and, arguably, this is a good thing.

Statistical mechanics, an area of physics that studies interactions between large numbers of particles (which provides tools to derive thermodynamics) kicked off in the 1870s with the work of Bolzmann [47]. Studying interpersonal relationships in social science is certainly not a new concept either, and scientific texts date back to 1887, when the social scientist Tönnies classified different types of social links between individuals and described the way they form communities and societies [264].

Few academics have been trained as "network scientists" *per se*. Conference crowds are predominantly made up of physicists, social scientists, statisticians, mathematicians, computer scientists, and combinations thereof. From personal experience, mathematicians, and in particular graph theorists, participate little in this conference sphere. However, many graph-theoretical journal articles make explicit links to applications and mention the current explosion in interest in networks. At the same time, network scientists are still discovering the vast knowledge base of graph theory (and probabilistic graph theory in particular). This text is meant to be a contribution to bridging the gap between the applied field of

network analysis and the more theoretical field of probabilistic graph theory (or *random graph theory*, as it is more commonly called). It emphasises the differences in motivation between these two different fields, while drawing direct parallels between the tools that they use.

1.2 Contributions

In terms of contributions to network science, Bollóbas summarized them well in [45], by stating that journal articles on the subject usually fall into one the following five categories.

(A) Direct observations by measuring properties of empirical networks.

(B) Suggestions of new (random) models motivated by **(A)**.

(C) Observations of computer simulations of the models proposed in **(B)**.

(D) Heuristic investigation of these new models to predict their properties.

(E) Rigorous theorems and proofs about these models.

I would like to add some more categories to this.

(F) Suggestions of new ways to define, detect, and measure properties of networks.

(G) Calculations of the measures proposed in **(F)** on the models proposed in **(B)**.

(H) Rigorous theorems and proofs about these measures.

This thesis contains components from almost all of the categories A–H listed above. See the summary in Section 9.1.

1.3 Outline

This introductory chapter is purposely brief, as most of the chapters contain comprehensive introductions to the subjects discussed in them, with links to the earliest known studies to highlight the well-established roots on which network science builds. Usually, the research in these early works form good introductory examples, so we often discuss them in detail. We will introduce some standard notation in the next section.

In Chapter 2, we introduce probabilistic methods related to graph theory, as well as a few random-graph models that are studied in both theoretical and applied settings. In Chapter 3, we introduce graph-colouring problems. A *proper graph colouring* is a colouring in which every vertex receives a colour, such that two vertices that share an edge receive different colours. The basic graph-colouring question is the following: what is the minimum number of colours necessary (at least) to colour a certain graph or class of graphs? Colouring problems are well-studied from a theoretical perspective and have many applications. In Chapters 3 and 4, we focus primarily on theory, although we will revisit colourings in an applied setting in Chapter 8. Chapter 3 is an introduction to results in this area; we also present many examples. Chapter 4 consists entirely of a proof of a conjecture posed in the previous chapter; it is concerned with a special type of colouring, called an *equitable colouring*, in which every colour is equally well-represented in terms of number of vertices that receive that colour.

Chapter 5 discusses the applied problem of *community detection*. Intuitively, a community is a subset of the nodes in a network that are relatively densely connected to each other and relatively sparsely connected to the rest of the network. A community structure of a network is an assignment of its nodes to communities. There are many different definitions of a community and of how one should measure its quality. A much-used quantifier for the quality of a community structure is the so-called *modularity function* Q. Most algorithms to find community structure optimize this function (or a closely related one), but a wealth of other notions exist. We present several methods and algorithms –mainly because of its

parallels with *core-periphery structure*, which we discuss in Chapter 7.

In Chapter 6, we present an introduction to *centrality* measures on networks. These are measures that attempt to quantify the importance of a node in a network. This chapter also forms an introduction to Chapter 7, where we revisit many of the same concepts.

In Chapter 7, we discuss *core-periphery structures* and propose a new method to detect them. A core-periphery structure of a network is a division of the nodes into densely connected core nodes and sparsely connected periphery nodes. Core nodes should also be reasonably well-connected to periphery nodes, but periphery nodes should not be well-connected to each other or to the core. Hence, a node belongs to a core if and only if it is well-connected both to other core nodes and to periphery nodes. A core structure in a network is thus not merely densely connected but also tends to be central to the network in some other way. As was the case with community structure in Chapter 5, core-periphery structure is an intuitive concept, but it is hard to pin down precisely. A few definitions and detection methods have been proposed, and we hope to contribute to the understanding of these methods by introducing our own technique and performing a set of comparisons with various existing methods using empirical networks.

In Chapter 8, we apply methods from Chapters 5 and 7 to empirical networks: human brain networks. We introduce methods that create human brain networks from fMRI data, which measures activity levels in different regions of the brain. We present the results of an experiment that was performed and analysed as a collaboration (of which the author was part) between network scientists and neuroscientists. This study was aimed at detecting patterns in brain networks as a result of a person learning a new skill, and uses core-periphery structure in brain networks and the way it changes as a predictor of how fast a person learns. To supplement the above collaboration work, we also detect a community structure in the brain networks by combining core-periphery structures from Chapter 7 and colouring concepts from Chapter 3.

In Chapter 9, we summarise the contributions of this thesis and list some open questions

and possibilities for further work.

1.4 Notation

1.4.1 Symbols

We define symbols where they are used for the first time in the text. The reader should refer to the list of symbols on page xiii, that contains brief definitions and page references to full descriptions for symbols used. Some symbols are used for multiple purposes, and there are simply not enough letters in the Roman and Greek alphabets. The meaning of a symbol, if used to denote a function, may depend on the type of argument used. For example, $\omega(G)$, where G is a graph, denotes the *clique number* of G, or the size of the largest complete subgraph (see page 31). However, $\omega(f(n))$, where $f(n)$ is a function of n, is used to describe the limiting behaviour of a function when $n \to \infty$ (see page 8). At some points in the text, a symbol may be used briefly for a different purpose than its "standard" meaning described in the List of Symbols. Often, the reason for this is that we are describing work by other authors and do not wish to stray from their original notation. For example, we use α (with no arguments) as a parameter for core sharpness in our core-periphery detection method described in Section 7.4, and the character is described that way in the List of Symbols. However, in Section 7.3.5, we briefly use α in the same way as it is used in [82] to describe Della Rossa *et al*'s notion of an α-periphery. When a symbol conflict like this occurs, we strive to provide enough clarity in the context of the text to prevent any confusion.

1.4.2 Graphs

Let $G(V, E)$, or simply G, be a graph: set of *vertices* (sometimes called *nodes*) $V(G)$ and a set of edges (or links) $E(G)$ connecting pairs of those vertices. Unless explicitly stated

otherwise, we assume that G is a binary, undirected graph –*i.e.*, every vertex pair either has an edge or non-edge between them, and the edges do not carry a weight or direction. We will use the notation $n = |V(G)|$ for the number of vertices and $m = |E(G)|$ for the number of edges of a graph G. We denote the *degree* of a vertex i in G by $d(i)$, where the degree is the total number of other vertices that share an edge with vertex i (*i.e.*, the number of *neighbours* of i). Let $\Delta(G)$ denote the maximum degree over all vertices of a graph G, and let $\delta(G)$ denote the minimum degree. The set of neighbours of a vertex i is denoted by $\Gamma(i)$, so $\Gamma(i) = \{j \in V(G) | (i, j) \in E(G)\}$. Analogously, we use $\overline{\Gamma}(i)$ to denote the set of *non-neighbours* of i (*i.e.*, the vertices that do not share an edge with i). This means that $\Gamma(i) \cup \overline{\Gamma}(i) = V(G)$, $\Gamma(i) \cap \overline{\Gamma}(i) = \varnothing$, and $i \in \overline{\Gamma}(i)$ unless i has a self-edge. The neighbourhood of a set of vertices U is denoted by $\Gamma(U)$, and is the union of $\Gamma(i)$ such that $i \in U$. The sets $\Gamma(U)$ and U are not necessarily disjoint.

We use G as a symbol for any graph. We use the notation K_n to denote the *complete* graph on n vertices: the graph consisting of n vertices and an edge on every pair of vertices. A *bipartite graph* G is a graph where the vertices can be split into two sets V_1 and V_2, such that no edge of G has both of its endpoints in the same set. A *k-partite* graph is defined similarly. Let K_{n_1,n_2} denote the *complete bipartite* graph that consists of two vertex sets V_1 and V_2, such that $|V_1| = n_1$, $|V_2| = n_2$, and the edge $(i, j) \in E(K_{n_1,n_2})$ if and only if $i \in V_1$ and $j \in V_2$. Let a cycle on a graph be a set of vertices $\{v_1, \ldots, v_k\}$, such that $(v_i, v_{i+1(\text{modk})}) \in E(G)$ for $1 \leq i \leq k$. We use $T(V, E)$ for a graph that is a tree: a connected graph without any cycles. We use C_n to denote the graph that is *just* a cycle on n vertices with no other edges.

A *hypergraph* is a generalisation of the graph defined above. A hypergraph is defined similarly to a standard graph, except that edges can span any number of vertices rather than just two. The number of vertices an edge spans is called its *edge-degree*.

1.4.3 Vectors and matrices

To distinguish more easily between scalars, vectors, and matrices, we use the following convention. WE usually denote scalar constants by lower case letters (from the Roman or Greek alphabet). We denote sets and graphs by upper case letters. We denote vectors by lower case and underlined letters, and we denote matrices by upper case and bold face letters. For efficiency, we often denote scalar entries of vectors and matrices by lower case, non-bold face letters (corresponding to the letter denoting the relevant vector or matrix) and subscript indices. The number of indices indicates the type of object from which the entry is taken. For example $a_{i,j} = \mathbf{A}(i,j) = [\mathbf{A}]_{i,j}$ is the entry found in the ith row and jth column of the matrix $\mathbf{A}$. We use a similar notation for the vertices of a graph. For example, $V(G) = \{v_1, \ldots, v_n\}$ is the set of vertices of a graph G.

1.4.4 Big O notation

Big O notation describes the behaviour of a function in the limit as its argument goes to infinity. It is often used when dealing with error terms, probabilities, time and space requirements for algorithms, or other situations in which only a bound on the limiting behaviour of a function is of interest. The notation is defined as follows (where c is a positive constant). The abbreviation "s.t." stands for "such that". Let $f(n), g(n) \geq 0$, and let $c, c_1 > 0$.

Notation	**Definition**
$f(n) = O(g(n))$	$\exists c, n_0$ s.t. $\forall n > n_0,\ f(n) \leq cg(n)$
$f(n) = \Omega(g(n))$	$\exists c, n_0$ s.t. $\forall n > n_0,\ f(n) \geq cg(n)$
$f(n) = \Theta(g(n))$	$\exists c_1, c_2, n_0$ s.t. $\forall n > n_0,\ c_1 g(n) \leq f(n) \leq c_2 g(n)$
$f(n) = o(g(n))$	$f(n)/g(n) \to 0$ as $n \to \infty$
$f(n) = \omega(g(n))$	$f(n)/g(n) \to \infty$ as $n \to \infty$

Chapter 2

Random Graphs

This chapter is a literature review and contains no original research.

2.1 Introduction

This chapter is an introduction to the probabilistic method and models of random graphs. This chapter and Chapters 3 and 4 will focus almost exclusively on results on properties of random graphs and results obtained by the probabilistic method. If a property occurs with a non-zero probability on random graphs, then that is a proof of the existence of at least one graph with that property. Hence, the probabilistic method is useful for existence proofs and extremal combinatorics. Explicit construction of large graphs or even an exhaustive computer search becomes hard very quickly: even searching all graphs with $n > 10$ is non-trivial. The probabilistic method makes it possible to consider the space of arbitrarily large graphs and the first or second moments of the desired properties with ease and without the need of construction.

As we shall see in Chapters 5 – 8, random graphs are also useful as *null models* [214]. When studying a real-world, or *empirical*, network, we are interested in certain properties and summary statistics of that network. In order to decide whether such properties are unexpected, we clearly must first determine the expected properties.

Deciding what family of randomized graphs to which to compare a real-world network is not an easy task, and although this chapter and Chapters 3 and 4 will focus mostly on so-called Erdős-Rényi random graphs, we will also briefly discuss two other random-graph distributions: the *configuration model* (in Section 2.7) and the *Barabási-Albert model* (in Section 2.8).

The first approach to a combinatorial problem using probabilistic methods was by Kendall and Babington-Smith in 1940 [158] and Szele in 1943 [261][1], and the latter result is a good introduction to the subject. A *complete tournament* is a complete graph, where every edge has a direction from one endpoint to the other endpoint. Thus, an edge $(i, j) \in E(G)$ points from vertex i to vertex j. One can think of this as a sports tournament in which every player plays against every other player, such that each game has exactly one winner and one loser (there are no ties). Additionally, winning is not *transitive*; if person A beats person B, and person B beats person C, it is still possible for person C to beat person A. If there are n players, then there are $\binom{n}{2}$ pairs of players and $2^{\binom{n}{2}}$ possible resulting directed graphs. Let a *Hamilton path* be an ordering of the vertices $v_1, \ldots, v_n$, such that $(v_i, v_{i+1}) \in E(G)$ for all $i = 1, \ldots, n-1$. One then asks whether there exists an integer $P(n)$ such that one can always find a complete directed graph on n vertices containing $P(n)$ Hamiltonian Paths. The answer is yes. Consider a tournament in the sport of coin flipping, where a fair coin is flipped to determine the winner. What is the expected number of Hamiltonian Paths in the complete coin-flipping tournament graph? There are $n!$ possible orderings; for each ordering, the probability that it fits the criterion of a Hamiltonian Path is $\left(\frac{1}{2}\right)^{n-1}$, because all of the $n-1$ edges must point in the right direction. Then, the expected number of Hamiltonian Paths is $\frac{n!}{2^{n-1}}$. If all possible outcomes had less than this number of paths, it could not possibly be the expected number. Therefore, there exists at least one graph on n vertices that contains at least $\frac{n!}{2^{n-1}}$ Hamiltonian Paths.

Shortly after that, Erdős and Rényi founded what is now called the field of "probabilistic

[1]The paper by Szele in [261] is written in Hungarian, so we are relying on the exposition on page xii of [41], and [4].

combinatorics". In 1947, Erdős used a similar argument to the one above to show that $R(k) \geq 2^{k/2}$ [95], where $R(k)$ is the Ramsey number of k, which is the smallest number of vertices such that any graph on that number of vertices contains either a clique of size k or an empty subgraph of size k. Colloquially, it is the minimum number of people to invite to a party if ones goal is to have either a group of k people present who are all mutual friends, or a group of size k who are all mutual strangers. Erdős proved additional probabilistic results on Ramsey numbers in [92] and [96].

The study of random graphs took off when Erdős and Rényi introduced their uniform random graph models around 1960 [93, 94, 96]. They introduced the model $\mathcal{G}(n, m)$, which describes the distribution of labelled graphs on n vertices and m edges. A closely related model (which is also often referred to as the Erdős-Rényi model) $\mathcal{G}(n, p)$ was introduced independently by Solomonoff and Rapaport [255] and by Gilbert [125] slightly earlier. The distribution $\mathcal{G}(n, p)$ describes labelled graphs on n vertices where each edge occurs independently at random with uniform probability p. Both of these distributions have two parameters: one that controls the size (the number of vertices) of the graph and one that controls the the density (the number of edges). Given that the edges are distributed homogeneously in both cases, one could expect them to behave rather similarly. The distribution $\mathcal{G}(n, m)$ has a fixed number of edges, whereas $\mathcal{G}(n, p)$ has an expected number of edges. They do behave similarly asymptotically (as $n \to \infty$) [41] for most properties in which we are interested – such as giant components, connectedness, and chromatic number[2] –, so we will focus our attention on $\mathcal{G}(n, p)$. We will denote instances of graphs taken from these distributions by $G(n, p)$ and $G(n, m)$, so $G(n, p) \sim \mathcal{G}(n, p)$ and $G(n, m) \sim \mathcal{G}(n, m)$.

The rest of this chapter is organised as follows. In Section 2.2, we introduce some general notation related to random graphs that will be used in this chapter and Chapter 3–4. In Section 2.4, we will look at some basic properties of $G(n, p)$, the model that we will study in Chapter 4. In Section 2.5 we introduce a simple variation of this model by allowing

[2]For other properties, such as "having an even number of edges", the distinction between $\mathcal{G}(n, p)$ and $\mathcal{G}(n, m)$ is very important.

p to be non-uniform over the graph. In Sections 2.6 – 2.9, we look at several other proposed random graph models. In particular, these are models that have been proposed mainly to mimick the heavy-tailed degree distributions of many empirical networks. Finally, in Section 2.11, we introduce a set of probabilistic tools that we will use throughout the rest of this thesis.

2.2 Some notation

Let X denote a random variable, and let $\mu = \mathbb{E}[X]$ and $\sigma = \sqrt{\mathbb{E}[X^2] - \mu^2}$ be the mean and standard deviation of X, respectively. The variance is denoted by $\mathrm{var}(X) = \sigma^2$. Let $\mathbb{P}[A]$ denote the probability of event A, and let $I(A)$ be the indicator variable of event A, so

$$I(A) = \begin{cases} 1, & \text{if } A, \\ 0, & \text{if not } A. \end{cases} \tag{2.1}$$

When we write that P, where P is a statement about a random graph $G(n,p)$ or $G(n,m)$, is true *asymptotically almost surely*, or *with high probability*, we mean that P holds with probability $\mathbb{P} \to 1$ as $n \to \infty$, which is the same as saying that P holds with probability at least $1 - o(1)$. We are almost exclusively interested in the case where $n \to \infty$, as most rigorous results presented in this and the next two chapters only hold in this limit. This is comparable to the *thermodynamic limit* in physics. As a system of particles grows towards infinity, due to the central limit theorem, fluctuations in global properties become negligible.

2.3 A word on density

Strictly speaking, the number of edges in a graph with n vertices can range between 0 and $\binom{n}{2}$. Graphs are called *sparse* when they have relatively few edges and *dense* when they

have many edges. Almost all graphs of interest, both as theoretical objects and empirical networks, will have densities that range between $m = \Theta(n)$ and $m = \Theta(n^2)$.

In random-graph theory, graphs exhibit interesting properties suddenly when $m = \Theta(n)$. Erdős-Rényi random graphs, $G(n, p)$, consist of tiny components when $m = o(n)$. However, when $np \to c$, one observes behaviour that is extremely sensitive to the constant c [41, 96, 148]. When $c < 1$, all components are of size $O(\log n)$ with high probability. When $c = 1$, the maximum size of a component will almost surely be of order $n^{2/3}$. When $c > 1$, there will almost surely be a *giant* component: a component that contains a constant fraction of all vertices. Also, this giant component contains a number of cycles that is linear in n, while the other components contain no more than one cycle each.

Most empirical networks of current scientific interest are sparse, in the sense that the number of edges scales with the number of vertices [76,104,207]. One important exception is the class of *similarity networks*: weighted networks that are usually almost complete. We will see examples of these in Chapter 7 and 8. When it comes to algorithms, this implies that there is a special interest in algorithms of which the running time relies on the number of edges as much as possible, such that they are fast for sparse networks. We will discuss algorithms and running times for most methods that we discuss in later Chapters 5–8.

The behaviour of sparse and dense graphs is sufficiently different that we need two separate proofs for our theorems in Chapter 4. In that case, sparse means $p = n^{-1+\epsilon}$, $0 < \epsilon < 1$.

2.4 Basic properties of $G(n, p)$

We have mentioned the emergence of a giant component in Section 2.3 and will discuss many properties of $G(n, p)$ in Chapter 4, which is dedicated completely to this model. For a comprehensive text on the properties of Erdős-Rényi random graphs and $\mathcal{G}(n, p)$, see [41]. In this section, we mention three properties that are of interest to both pure and

applied mathematicians: the degree sequence, the appearance of small subgraphs, and the graph diameter. As examples, we mention some exact results on the asymptotic behaviour of these properties to give a flavour of the investigations devoted to this model.

One of the simplest invariants of a graph is its degree sequence. In $\mathcal{G}(n,p)$, the degree of a vertex has a binomial distribution with parameters $n-1$ and p. (The mean degree is $p(n-1)$.) Much attention has been given to studying the degree sequence of $G(n,p)$ [33, 36, 38, 92, 147, 195]. For example, if p is not too close to 0 or 1, it is known that there is a unique vertex of minimum and maximum degree, respectively, and that the exact maximum and minimum degree of $G(n,p)$ is determined if and only if p or $1-p$ is $o(\log n/n)$ [33]. For large enough n, one would expect the degrees of two vertices in $G(n,p)$ to be almost independent. McKay and Wormald showed that the distribution of the degree sequence for a wide range of Erdős-Rényi random graphs is, indeed, well approximated by the conditional distribution $\{(X_1,\ldots,X_n)|\sum_{i=1}^n X_i = 2m\}$, where X_i are independent random variables with the same distribution as the degree of one vertex [195]. Thus, if $N=\binom{n}{2}$, it follows that

$$\mathbb{P}[\{X_i = d(i)\} = k] = \frac{\binom{n-1}{k}\binom{N-n+1}{M-k}}{\binom{N}{M}}.$$

Another property of a graph G is the number of times that it contains a certain subgraph H, and this has been studied in considerable detail [15, 35, 92, 93, 244]. The class of *strictly balanced graphs* H is convenient for studying the asymptotic distribution of the number of subgraphs of $G(n,p)$ that are isomorphic to H. Let $\overline{d}(G)$ be the mean degree of a graph G. Let the *maximum average degree* of a graph G be $m(G) = \max\{\overline{d}(F)|F \subset G\}$. A graph G is *balanced* if $\overline{d}(G) = m(G)$, and it is *strictly balanced* if $\overline{d}(F) = m(G) \Leftrightarrow F = G$. The class of strictly balanced graphs includes complete graphs, cycles, and trees. The following theorem by Bollobás guarantees that the distribution of the number of subgraphs in $G(n,p)$ that are isomorphic to a constant-sized graph H is Poisson.

Theorem 2.4.1. *[35][see also [41], page 78] Let H be a fixed, strictly balanced graph with k vertices and $l \geq 2$ edges, and suppose that its automorphism group has a elements. Let $c > 0$ and $p = cn^{-k/l}$. Let $G \in \mathcal{G}(n,p)$, and let $X = X(G)$ be the number of H-subgraphs of G. It follows that*

$$\lim_{n\to\infty} \mathbb{P}[X = r] = e^{-\lambda}\lambda^r/r!, \quad r = 0, 1, \ldots.$$

The problem is much harder for subgraphs that are not strictly balanced. In [35], as a corollary of Theorem 2.4.1, Bollobás showed that there is a threshold function for the existence of a general subgraph F, and that this threshold function depends only on the maximum of the mean degrees of the subgraphs of F, *i.e.* $m(F)$.

The last property that we discuss is that of the graph *diameter*. A *path* in G between two vertices x and y is an ordered set of vertices $x = x_1, x_2, \ldots, x_l = y$, such that $x_i x_{i+1} \in E(G)$ for $1 \leq i \leq l-1$, and a *shortest path* is a path that minimizes l over all such sets. Let $d(i,j)$ denote the (integer) distance between vertex i and j (the length of a shortest path between them), and let $d(G)$ denote the diameter of a graph G. The *diameter* is the greatest distance between a vertex pair of G, taken over all pairs. The graph G is disconnected if and only if $d(G) = \infty$, so we only consider connected graphs. Again, this property has been studied extensively for $G(n,p)$ [37, 63, 71, 164]. For a wide range of p, almost all graphs $G(n,p)$ have the same diameter (as a function of n) [37]. Additionally, if $np \to \infty$, then the diameter of the giant component of G is close to $\frac{\log n}{\log np}$ [71].

2.5 Stochastic block model variations of $G(n,p)$

Sometimes probabilistic proofs or models of empirical networks call for non-uniform variations of $G(n,p)$. In its general form, a *stochastic block model*, the vertex set of a graph is partitioned into "blocks" $1, \ldots, q$ and if vertices i and j fall into block r and s respectively, then the edge probability $p(i,j) = p_{rs}$. Hence, the edge probability between two vertices depends only on the blocks to which the two vertices belong.

An important example of a stochastic block model is the random bipartite graph: $\mathcal{B}(n_1, n_2, p)$. This is the distribution of random graphs on the vertex set $V(G) = V_1 \cup V_2$, such that V_1 and V_2 are disjoint, $|V_1| = n_1$, $|V_2| = n_2$, and edges occur with probability p between V_1 and V_2, and with probability 0 within one of V_1 or V_2. This creates a random bipartite graph. We will use this in Chapter 4.

In applied network theory, non-uniform versions of $G(n, p)$ have proven useful as benchmarks for many properties. When testing a method designed to detect a certain property in a graph or a real-world network, it is useful to have a graph model that has the desired property in such a way that it can be tuned in order to test the sensitivity of the method. One can then prove rigorous results on the performance of the method, making use of the tractability of this random model, or one can generate many instances of it to obtain numerical results.

For example, in [151], Jerrum and Sorkin introduce the model $G(n, p, r)$: a graph with a "planted" bisection of (low) density r, that separates two halves of the graph with internal (high) density p. This model was introduced to prove results on the performance of a certain algorithm designed to detect the bisection that minimizes the edge density across the bisection. We will discuss this example further in Section 5.6.2, which describes the algorithm in question, *simulated annealing*.

Chapter 5 deals with so-called *community structure* in graphs. Community structure refers to the idea that empirical networks tend to form communities: sets of nodes that are more densely connected to each other than to the rest of the network [108, 211, 229]. One of the first methods of detecting such a structure was proposed by Girvan and Newman in [126], where the authors also propose a computer-generated model for testing such methods. In this model, the vertices are first partitioned into communities, and then the edges occur with p_{in} between two vertices that lie in the same community, and p_{out} between two vertices that lie in different communities, with $0 \leq p_{out} \leq p_{in} \leq 1$. This allows one to tune the random graph between the uniform model (when $p_{out} = p_{in}$) where the pre-determined

communities are indistinguishable and an extreme community structure of disconnected cliques (when $p_{out} = 0$ and $p_{in} = 0$).

We will employ a similar non-uniform variation of the $G(n, p)$ model ourselves in Chapter 7. In this chapter, we propose a new method of detecting so-called *core-periphery structure* in networks, and we compare existing methods to each other on both computer-generated graphs and empirical networks. Core-periphery structure is the organisation of a network into a densely connected core and a sparsely connected periphery, such that the core is also densely connected to the periphery [50, 211, 238]. To test different methods for detecting such a structure, we introduce a random model $\mathcal{G}(n_c, n_p, p, k)$. A graph G with this distribution has $n_c + n_p$ vertices in total, of which n_c are in the core and n_p are in the periphery. The edges occur independently at random. The edge probabilities for periphery-periphery, core-periphery, and core-core pairs are p, kp, and k^2p, respectively, with $p \in [0, 1]$ and $k \in [1, (1/p)^{1/2}]$. See Section 7.5 for a discussion of this model.

2.6 Watts-Strogatz model

The *Watts-Strogatz model* [272] was proposed as an attempt to give an extremely simple model of networks that exhibit the *small-world phenomenon* [166]. This characteristic of real-world networks was noticed and made famous in the 60s by Milgram after a social experiment carried out by Gurevich [137, 197], and it refers to the observation that social networks have a surprisingly small diameter: there are short paths between all pairs of nodes, even when a network is very large. In [272], Watts and Strogatz studied different types of empirical networks - including a power grid of the United States, a movie actor collaboration network and a neural network. They observed that these networks showed clustering (similar to lattices) and small diameter (similar to Erdős-Rényi graphs). Therefore, they proposed a model that is a tunable interpolation between these two (highly ordered and highly disordered) tyes of network.

Let a *ring lattice* of n vertices and constant degree $2k$ be a set of n vertices arranged on a circle, where each vertex is joined by an edge to its k nearest neighbours (on the circle) to the left and to the right. Highly ordered structures, such as lattices, are well-studied due to their mathematical tractability. For many networks in physics, such as interactions between particles in crystal-like structures, they are appropriate models. For social networks, however, they fail to capture important properties. Starting from a ring lattice, Watts and Strogatz proposed a random rewiring procedure that adds this property to the lattice. Each edge is rewired with probability $0 \leq p \leq 1$, so that one of its endpoints is replaced by a different one chosen uniformly at random. This allows one to tune the level of "disorder" in the network by letting p range from 0 to 1.

One important aspect of the Watts-Strogatz paper lies in the fact that it opened up the field of finding random graph models that behave similarly to real-world networks, but are built by simple rules that make them easy to both generate and study mathematically.

2.7 Configuration model

The *configuration model* is an important random-graph model in applied network science. The Erdős-Rényi model is the easiest random graph model to study theoretically due to its uniformity, but it falls short for investigating empirical networks, which rarely possess uniform degree distributions [211]. The configuration model plays an important role as a point of comparison, or a *null model*, in algorithms that, for example, detect communities in networks (see Chapter 5).

The *configuration model* was first introduced in its explicit form by Bollobás in 1980 [34]. Graphs with given degree sequences had already been studied in the context of enumerating them in [22], where Bender and Canfield studied random $0-1$ matrices with given row sums[3]. Bollobas found an asymptotic formula, without reference to an exact formula, using

[3]These matrices are slightly different from the configuration model. For example, a loop in the configuration model should correspond to a 2 on the diagonal of an adjacency matrix. Another, earlier, example of

probabilistic methods. To do this, he introduced the uniform probability space on all labelled graphs with a given degree sequence (allowing self-loops): the configuration model. We will discuss a few properties of this model in this section. Note that the special case of random regular graphs (where all degrees are equal), is a much more heavily studied object in graph theory. See, for example, Chapter 9 of [149].

In Section 2.3 we mentioned the sharp threshold for the existence of a giant component in Erdős-Rényi random graphs. Analogous results exist for random graphs with given degree sequences. In [181], Łuczak characterized the connectedness of a graph in the configuration model as a function of its minimum degree. When studying random graphs, one is usually interested in asymptotic results (*i.e.*, where $n \to \infty$). Using a given degree sequence is not always practical, as it would have to be an infinite sequence. Accordingly, in [199] and [201], Molloy and Reed defined an asymptotic degree sequence as a sequence of integers $\mathcal{K} = k_1(n), k_2(n), \ldots$, such that $k_i(n) = 0$ for $i \geq n$ and $\sum_{i \geq 0} k_i(n) = n$. Here $k_i(n)$ signifies the number of vertices of degree i in a graph G on n vertices. They called a degree sequence $\mathcal{K}$ *smooth* if there exist constants ρ_i such that $\lim_{n \to \infty} k_i(n)/n = \rho_i$. Let $Q(\mathcal{K}) = \sum_{i \geq 1} i(i-2)\rho_i$. Molloy and Reed showed that, if $Q(\mathcal{K}) > 0$, then a random graph with given degree sequence $\mathcal{K}$ has a giant component with many cycles; otherwise, it does not. See [199, 201] for a more nuanced description of the giant component in this model.

In [3], Aiello, Chung, and Lu defined the following power-law random graph model $P(\alpha, \beta)$. Let k_i be the number of nodes with degree i. Then $P(\alpha, \beta)$ assigns uniform probability to all graphs with $k_i = e^{\alpha}/i^{\beta}$. Thus, one considers a degree distribution that follows a power law, and then applies the configuration model: a uniform distribution over all graphs with that degree distribution. Note that α is the intercept and β is the slope when the degree sequence is plotted on a log-log scale. In [3], the authors proved some results on the behaviour of the giant component as a function of β: when $\beta < 1$, the graph is connected with high probability. When $1 < \beta < 3.4785\ldots$, there is a giant component.

random matrices with given row and column sums was studied in [21].

2.8 The BA and LCD models

The *Barabási-Albert model*, or the *preferential attachment model*, was originally introduced in 1976 by de Solla Price [254], but did not gain popularity until a version of it was proposed in 1999 by Barabási and Albert [14], as an attempt to model a heavy-tailed degree distribution often found in real-world networks as a consequence of the way in which networks grow. The term *preferential attachment* refers to the tendency of new nodes in a network to connect to *hubs* (*i.e.*, nodes of high degree that already exist). In a model in which high-degree nodes have a higher probability of increasing their degree as the network grows, the degree distribution becomes heavy-tailed.

The model is described as follows in the original paper [14]. One starts with a small number of vertices n_0. At every time step, one adds a new vertex with $m \leq n_0$ edges that connect to already-existing vertices. The probability that a new vertex j connects to an existing vertex i is proportional to $d(i)$. Then, after t time steps one has a network with $t + n_0$ vertices and mt edges. If $m = 1$, this yields a random non-uniform tree process, which had already been discussed earlier [262]. Barabási and Albert suggested that the proportion of ρ_k vertices in this model with degree k should obey a power law $\rho_k \sim \alpha k^{\gamma}$. They deduced by a mean-field argument that $\gamma = 3$. The intuition for this model is elegant, but there are some problems with its description in [14]. The reader might have noticed that the above description is a mixture of a model with a precise density parameter (m) and an expected density parameter ($\mathcal{P}(ij \in E(G)) = d(i)/\sum_k d(k)$), which is analogous to the difference between $G(n, m)$ and $G(n, p)$. This created the question of exactly how new edges are chosen. Additionally, we have not specified a starting graph, only the number of vertices in the starting graph. For the model to work, there must also already be edges present.

Details such as the seed graph and the exact way in which one chooses new edges can make a significant difference to the asymptotic behaviour of the model. This might seem surprising, because the seed graph is only a constant-sized subgraph of the final graph, and distinguishing the exact ways of choosing the edges may seem similar to distinguishing

between $G(n,p)$ and $G(n,m)$. The introduction of this model prompted a series of papers by Bollobás, Riordan and others, that give a more formal definition [43, 45, 46]. In [45], Bollobás proves that tweaking the exact way of choosing new edges and the seed model - subject to the condition $\mathcal{P}(ij \in E(G)) = md(i)/\sum_k d(k)$ is satisfied- gives almost complete control over the exact number of triangles in the final graph. This is a rather important property. In [46], Bollobás and Riordan proposed a model that is precise and has convenient properties in terms of rigorous analysis. As in the configuration model in Section 2.7, it is convenient to allow loops and multiple edges. Let $(G_m^{(t)})_{t\geq 0}$ be a graph process in which one adds one vertex and m edges at each time step t. We define $(G_1^{(t)})_{t\geq 0}$ such that $G_1^{(t)}$ is a graph on $\{v_i | 1 \leq i \leq t\}$ as follows. Start with $G_1^{(1)}$, a graph that consists of one vertex and one loop. Create $G_1^{(t)}$ from $G_1^{(t-1)}$ by adding a single vertex v_t and an edge between v_t and v_i, where v_i is a vertex chosen at random with probability

$$\mathbb{P}[i = j] = \begin{cases} d_{G_1^{(t-1)}}(v_j)/(2t-1), \ 1 \leq j \leq t-1, \\ 1/(2t-1), \ j = t. \end{cases}$$

We then define $(G_m^{(t)})_{t\geq 0}$ by running $(G_1^{(t)})_{t\geq 0}$ on vertices $v_1', v_2', \ldots$ and creating $G_m^{(t)}$ from $G_1^{(t)}$ by merging $v_1', v_2', \ldots, v_m'$ into v_1, $v_{(i-1)m+1}', v_{(i-1)m+2}', \ldots, v_{(i-1)m+m}'$ into v_i. In this case, merging a set of vertices entails replacing them by a single vertex that keeps all of the neighbours of the original set (allowing for multiple edges where multiple vertices in the original set were joined to the same neighbour). They named this model the *LCD model*, because it has a static description that is related to *linear chord diagrams* [44].

In [43], Bollobás and Riordan proved that the asymptotic diameter (the greatest distance between any two vertices of a graph; see page 15) of this graph process is $\log n$ for $m = 1$ and $\log n / \log\log n$ for $m \geq 2$. In [43], Bollobás, Riordan, Spencer, and Tusnády confirmed the value conjectured by Barabási and Albert of $\gamma = 3$ for the power-law degree distribution of this process.

2.9 Other models

2.9.1 The Buckley-Osthus model

The *Buckley-Osthus model* was introduced independently in [87] and [88] and was formalized by Buckley and Osthus [58]. Their model $H_{a,1}^{(t)}$ is defined for directed graphs and is defined similarly to $G_1^{(t)}$ in the LCD model described in Section 2.8, except that

$$\mathbb{P}[i = j] = \begin{cases} \frac{d^{in}_{H_1^{(t-1)}}(v_j)+a}{(a+1)t-1}, & 1 \le j \le t-1, \\ \\ \frac{a}{(a+1)t-1}, & j = t, \end{cases}$$

where $d_G^{in}(i)$ denotes the in-degree (number of directed edges pointing towards a vertex) of vertex i in graph G. The out-degree is denoted similarly as $d_G^{out}(i)$. The model $H_{a,m}^{(t)}$ is created from $H_{a,1}^{(t)}$ in the same way that $G_m^{(t)}$ is created from $G_1^{(t)}$ in Section 2.8. Edge endpoints are chosen with a probability proportional to their in-degree plus an initial attractiveness am, where a is a constant. Buckley and Osthus proved that this process results in a power-law degree distribution.

2.9.2 The copying model

The *copying model* was introduced in [175] as an alternative way to explain the heavy tail degree distribution found in networks such as the World Wide Web. The model has two parameters: a *copying constant*, $0 \le \alpha \le 1$, and a constant out-degree d^{out}. We start with a seed graph $G_{\alpha,d^{out}}^{(1)}$ of constant out-degree d^{out}. At each time step, t, we add one vertex i to the graph. This new vertex receives d^{out} out-edges. To pick the endpoints of these edges, i first picks a vertex j from $V(G_{\alpha,d^{out}}^{(t-1)})$ uniformly at random. Then, for $k = 1, \ldots, d^{out}$, the vertex i either sends an edge to the kth neighbour of j, or it picks a neighbour uniformly at random from $V(G_{\alpha,d^{out}}^{(t-1)})$ and sends its kth edge towards it. The former (copying) occurs with probability α, and the latter occurs with probability $1 - \alpha$. This model also has a power-law

degree distribution [175]. A difference with, for example, the Barabási-Albert model is that the copying model creates many dense bipartite subgraphs (see page 7), a phenomenon that has been observed in the world wide web [176].

2.9.3 The Cooper-Frieze model

The *Cooper-Frieze model* is a very general model with many parameters that includes the LCD model, the Buckley-Osthus model, and the copying model as special cases [78]. See reference [78] for the exact definition of the model and a general proof of its asymptotic power-law degree distribution.

2.10 Entropy of networks

From a physics perspective, the uncertainty about a network topology, provided that we have knowledge about some properties of the network: number of edges, degree distribution, community structure, *etc.*, can be expressed as the *entropy* of the system. It is a measure of the number of ways in which a system can be arranged, measuring its disorder. Looking at a network through this lens provides with a new way of 'fitting' a network to an appropriate ensemble of random networks [27]. In [28], Bianconi *et al* propose entropy as an indicator of the role of certain features such as degree distribution, degree correlations and community structure.

2.11 Some probabilistic tools

In this section we give some probabilistic tools that we will use in the rest of this thesis (especially, in Chapter 4). In statistics, *moments* are quantities that describe the shape of the distribution of a variable. The *first moment* of a random variable is the expected value μ. In a binary random variable, which is either true (1) or not true (0), the expectation is

equal to the probability that the variable is true. The *second moment* of a random variable describes the "width" of the distribution (*i.e.*, how far the variable tends to stray from the mean). We will discuss *third* and *fourth* moments of a distribution in an applied setting in Chapter 8.

2.11.1 First-moment methods

We present Markov's Inequality and the Union Bound, which are lemmas that indicate as much as possible about bounds on a random variable X from only the first moment of a distribution.

2.11.1.1 Markov's Inequality

Markov's Inequality is a simple bound for the probability that a random variable exceeds its expectation by some constant factor. Much tighter bounds exist once the variance of X is known, such as Chebychev's Inequality in Section 2.11.2.1. Markov's Inequality is a useful bound when nothing is known about X other than the expectation μ.

Lemma 2.11.1. *[Markov, see [7], p. 310] Let $X \geq 0$. For any positive α,*

$$\mathbb{P}\left[X \geq \alpha\mu\right] \leq \frac{1}{\alpha}.$$

2.11.1.2 Union bound

The union bound, which is also known as Boole's Inequality, states that the probability that at least one event in a set of events occurs is bounded by the sum of the probabilities of the individual events:

Lemma 2.11.2.

$$\mathbb{P}\left[\bigcup_i A_i\right] \leq \sum_i \mathbb{P}\left[A_i\right]. \tag{2.2}$$

2.11.2 Second-moment methods

First moment methods can be used to show that at least one instance exists "on either side" of the expected value. This is very useful for the purpose of existence proofs, for example, but in many cases we might be interested in the concentration of the values of X around their mean. The second moment gives us information by estimating $|\mathbb{E}[X] - X|$.

2.11.2.1 Chebychev's Inequality

Chebychev's Inequality uses the second moment to bound the probability that a random variable deviates far from its mean.

Lemma 2.11.3. *[Chebychev; see page 43 of [7], and [167]] For any positive α,*

$$\mathbb{P}[|X - \mu| \geq \alpha\sigma] \leq \frac{1}{\alpha^2}.$$

Proof.

$$\sigma^2 = \mathbb{E}\left[(X - \mu)^2\right] \geq \alpha^2\sigma^2\mathbb{P}[|X - \mu| \geq \alpha\sigma].$$

□

2.11.2.2 Hoeffding's inequality

In [145], Hoeffding provided bounds for the probability that the sum of instances of a random variable deviates from its expected value. Let $X_1, \ldots, X_n$ be random independent variables and let

$$S = \sum_{i=1}^{n} X_i, \qquad \overline{X} = \frac{S}{n},$$
$$\mu = \mathbb{E}[\overline{X}], \qquad \sigma^2 = n\mathrm{var}(\overline{X}) = \frac{\mathrm{var}(S)}{n}.$$

If $X_1, \ldots, X_n$ have a common mean and variance, then these are μ and σ^2.

Lemma 2.11.4. *[145] If $X_1, \ldots, X_n$ are random independent variables, such that $0 \leq X_i \leq 1$, where $i = 1, \ldots, n$, and α is any positive constant, then*

$$\mathbb{P}[|\overline{X} - \mu| \geq \alpha] \leq 2e^{-2n\alpha^2}.$$

A more general version of the lemma states that if $a_i \leq X_i \leq b_i$, where $i = 1, \ldots, n$, then

$$\mathbb{P}[|\overline{X} - \mu| \geq \alpha] \leq 2\exp\left[-\frac{2n^2\alpha^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right].$$

2.11.2.3 Chernoff bounds

In 1952, Herman Chernoff wrote a seminal paper that gives a set of exponentially decreasing bounds on the sums of independent random variables [70]. This work was motivated by the following setting. Suppose that $\mathbb{P}[X = 1] = p_i$ and $\mathbb{P}[X = 0] = 1 - p_i$, where $i \in \{0, 1\}$ and $p_1 > p_0$. We must decide whether to accept or reject a null hypothesis based on a knowledge of $S_n = \sum_{i=1}^{n} X_i$. When the difference between the means of the distribution of S_n under H_0 and its distribution under H_1 (where H_0 is the null hypothesis that $i = 0$) is small compared to the standard deviations of S_n under the respective hypotheses, one can compute the error probabilities using a normal approximation. However, when the difference in means is large compared to the standard deviations, we are dealing with the behaviour of S_n in the tails of the distribution. A version of this problem on continuous distributions was studied by Cramér in [80].

The following version of the Chernoff bound is an extremely useful probabilistic tool that we will employ many times in the proofs in Chapter 4.

Lemma 2.11.5. *[Chernoff, see Section 2.1 in [149], and [70]] If $X = \mathrm{Bi}(n, p)$, $\mu = np$,*

$\lambda > 0$ *then the following holds:*

$$\mathbb{P}[X < \mu - \lambda] \leq e^{-\frac{\lambda^2}{2\mu}},$$
$$\mathbb{P}[X > \mu + \lambda] \leq e^{-\frac{\lambda^2}{2\mu+\lambda}}.$$

2.11.3 Lovász Local Lemma

When using probabilistic methods, one cares only about whether the probability of success is nonzero, rather than precisely how high it is. A nonzero probability already proves the existence of at least one instance of the object one seeks. The Lovász Local Lemma can be used for such situations. It shows that there is a nonzero probability of none of the events in a set of (bad) events occurring, as long as the probability for each of them is bounded (as is the case with the Union Bound). However, the Lovász Local Lemma can also be used when there is some mutual dependence between events, as long as there is a bound on the number of other events on which each event depends.

Lemma 2.11.6 ([91], see also [257] and page 39 of [202])**.** *Consider a set $\mathcal{A}$ of events, such that $\forall A \in \mathcal{A}$, $\mathbb{P}[A] \leq p$, for some $0 \leq p < 1$. Additionally, A is mutually independent from all but at most d other events in $\mathcal{A}$.*

Then, if $4pd \leq 1$, it follows that $\mathbb{P}[\text{none of the events in } \mathcal{A} \text{ occur}] > 0$.

2.11.4 Martingales and Azuma's Inequality

In combinatorics, martingales were first used to obtain concentration results for the chromatic number of random graphs [245]. So, they will come back briefly in Chapter 3, although we do not go into much detail of those proofs. For a survey, see [74]. When independence assumptions do not hold on a set of random variables, it becomes hard to find concentration results such as those by Hoeffding and Chernoff. Martingale inequalities provide very good bounds when a random variable can be expressed as a chain of

events. A martingale is a sequence of random variables $X_0, X_1, X_1, \ldots$ with finite means, such that [106]

$$\mathbb{E}[X_{t+1}|X_0, X_1, \ldots, X_t] = X_t.$$

A simple example of a martingale is a one-dimensional random walk. A random walker starts at position $X_0 = 0$, and at every time step t, to position $\mathbb{P}[X_{t+1} = X_t + 1] = \mathbb{P}[X_{t+1} = X_t - 1] = 1/2$. At any time t, we have that

$$\mathbb{E}[X_{t+1}|X_0, X_1, \ldots, X_t] = \mathbb{E}[X_{t+1}|X_t] = X_t.$$

A martingale is said to be $\underline{c}$-*Lipschitz* for $\underline{c} = (c_1, \ldots, c_n)$, if $|X_i - X_{i-1}| \leq c_i$. Azuma's Inequality provides a powerful tool for controlling martingales:

Lemma 2.11.7. *[Azuma's Inequality; [11, 145, 260]] If a martingale X is $\underline{c}$-Lipschitz, for $\underline{c} = (c_1, \ldots, c_n)$, $\lambda \geq 0$, then*

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq \lambda] \leq 2\exp\left[-\frac{\lambda^2}{2\sum_{i=1}^{n} c_i^2}\right].$$

Chapter 3

Graph Colouring and the Probabilistic Method

This chapter is a literature review and contains no original research.

3.1 Introduction

A *proper vertex colouring* is a labelling of the vertices of a graph such that no edge connects two identically coloured vertices. The *chromatic number* $\chi(G)$ of a graph is the smallest number of colours needed for such a colouring. The problem of colouring graphs started with the (in)famous *Four Colour Theorem*, which states that all planar graphs can be coloured with 4 colours. The problem was first posed by Francis Guthrie in 1831 [174], but was not solved until 1977, and it was proved by extensive computer search [9]. No short proof is known to exist. This shows how even a seemingly simple version of the colouring problem can prove extremely hard to solve. Graph colouring is a huge field that spans many types of problems, approaches, and practical applications [141, 278]. For an overview of deterministic problems, for example, see reference [150]. Techniques for proofs on random graphs are very different from deterministic ones, and (as we will see) there exist accurate results with elegant proofs for $G(n, p)$. For overviews on colouring and

probabilistic methods, see, for example, references [41] and [202].

The chromatic number $\chi(G)$ of a graph is the smallest number of colours needed for any proper colouring. An *equitable colouring* of a graph is a proper vertex colouring such that the sizes of any two colour classes differ by at most one. The *equitable chromatic number* $\chi_{=}(G)$ of G is defined as the least positive integer k for which there exists an equitable colouring of a graph G with k colours. The equitable chromatic threshold $\chi^{*}_{=}(G(n,p))$ of G is defined by the least positive integer k such that for every $k_0 > k$ there exists an equitable colouring of G with k_0 colours. We use the notation "log" to denote the natural logarithm ln or $\log_e$. We set $b = \frac{1}{1-p}$, and $d = np$.

The rest of this chapter is organised as follows. In Section 3.2, we briefly discuss results on cliques and independent sets in $G(n,p)$, given their close relation to the chromatic number. In Section 3.3, we give an overview of some of the known results of the original chromatic number on $G(n,p)$. In Section 3.4, we highlight the *expose-and-merge* technique, which was developed by Matula to obtain the chromatic number of a random graph. (We will use this in our own proofs in Chapter 4.) In Section 3.5, we highlight a few practical colouring algorithms that relate to random graphs and probabilistic methods. In Section 3.6, we discuss the equitable chromatic number and some known results on it, and we introduce a conjecture by Krivelevich and Patkós that we will resolve in Chapter 4.

3.2 Cliques and independent sets in $G(n,p)$

Let a *clique* of a graph G be a subset of vertices of G such that there exists an edge between every pair of vertices in the subset[1]. Similarly, let an *independent set* of G be a subset of the vertices of G such tha there is no edge between any pair of vertices in the subset. Independent sets are sometimes referred to as *stable sets* of G.

[1]Sometimes the condition that a clique is a *maximal* complete subgraph is added to the definition, so that no more vertices of G are fully connected to it and could be added to increase its size. (Note that this is not the same thing as a clique of maximal size in G.) We do not use this condition.

Let $\omega(G)$ be the size of a largest clique in G, and let $\alpha(G)$ be the *independence number* of a graph G. The independence number is the size of a largest independent set of G. We use the notation $[k]$ to denote the set of integers $1, \ldots, k$. If $C : V(G) \to [k]$ is a proper colouring of G using k colours, then the set of vertices that receive the same colour i must form an independent set, because, in a proper colouring, no two vertices of the same colour can have an edge between them. Let $\overline{G}$ denote the *complement* of a graph G: a graph such that $V(\overline{G}) = V(G)$ and $(i, j) \in E(\overline{G})$ if and only if $(i, j) \notin E(G)$. An independent set in G is a clique in $\overline{G}$. Because $\alpha(G) = \omega(\overline{G})$, $\alpha(G)$ and $\omega(G)$ are very closely related. For random graphs $\overline{G(n, p)} \sim G(n, 1 - p)$.

The independence number of a graph is related to the chromatic number. If there are no independent sets larger than $\alpha(G)$, then clearly $\chi(G)$ is bounded below by $|V(G)|/\alpha(G)$. For random graphs $G(n, p)$, we shall see that this is a tight bound.

Cliques in $G(n, p)$ were first studied in [188], where Matula found expressions for the expectation and the standard deviation of the number of cliques and maximal cliques, respectively, as a function of their size. Grimmett and McDiarmid showed that, with high probability and for fixed p, the largest clique size $\omega(G(n, p)) \to 2 \log_{1/p} n$ [134]. Thus, $\alpha(G(n, p)) \to 2 \log_{1/(1-p)} n$. Independently, Matula showed that, for fixed p, the largest clique size $\omega(G(n, p))$ is highly concentrated: it takes one out of two values with high probability [189, 190].

For sparse graphs, Frieze proved the following theorem in 1990.

Theorem 3.2.1. *[117] For $G(n, p)$ with $p = p(n)$ and $d_\epsilon < d = o(n)$, for a sufficiently large constant d_ϵ, one obtains, with high probability,*

$$\left| \alpha(G(n, p)) - \frac{2}{p} (\log d - \log \log d - \log 2 + 1) \leq \frac{\epsilon}{p} \right|.$$

For the case $d > n^{2/3}$, this was already shown in [39].

3.3 The chromatic number of $G(n,p)$

The expected chromatic number of $G(n,p)$ is described in the following theorem.

Theorem 3.3.1. *[39, 134, 182, 193] For* $\log^{-8} n < p < 1$*, it follows that, with high probability,*

$$(1-o(1))\frac{n}{2\log_b n} \leq \chi(G) \leq (1+o(1))\frac{n}{2\log_b n}.$$

For $c/n < p < \log^{-8} n$*, where* $c > 0$*, with high probability,*

$$(1-o(1))\frac{d}{2\log d} \leq \chi(G) \leq (1+o(1))\frac{d}{2\log d}.$$

Note that $\log_b d \sim \frac{\log d}{p}$ when $p \to 0$. In 1975, Grimmett and McDiarmid established that $(1-o(1))\frac{n}{2\log_b n} \leq \chi(G) \leq (1+o(1))\frac{n}{\log_b n}$ for constant p [134]. Their lower bound comes from establishing the independence number of $G(n,p)$ (see Section 3.2), and the upper bound was established by construction of an explicit algorithm known as the *greedy colouring algorithm*. We will discuss this algorithm at length in Section 3.5.1. In 1988, Bollobás improved their result by showing that the lower bound is tight [39]. We briefly outline Bollobás' proof, because it is similar to the so-called *expose-and-merge* technique that we will present in Section 3.4 and for our original work in Chapter 4). This proof works for constant p and for $p \to 0$ slowly enough (*i.e.*, $p(n) > n^{-1/3+\epsilon}$, where $\epsilon > 0$). For simplicity, let $p = 1/2$.

Theorem 3.3.2 (Bollobás [39]; see also [248])**.** *For* $G \sim \mathcal{G}(n,1/2)$*, we have*

$$\chi(G) \leq (1+o(1))\frac{n}{2\log_b n}.$$

Proof. We start with the following lemma (3.3.3) about the independence number of $G(n,1/2)$. One defines a graph process that consists of revealing one edge of a random graph $G(n,p)$ at the time. This affects only one of a maximal set of edge-disjoint cliques at a time, so

one can use a martingale approach to proving this lemma (see Section 2.11.4). With this lemma, the result on the chromatic number of $G(n, 1/2)$ follows easily. We will discuss and use sparse versions of this lemma in Lemma 3.3.5. Let $k(n) = 2\log_2 n - 3$.

Lemma 3.3.3. *[39] For $G \sim \mathcal{G}(n, 1/2)$, we have*

$$\mathbb{P}[\alpha(G) < k(n)] < e^{-n^{2+o(1)}}.$$

Let $S \in V(G)$ be an arbitrary subset of the vertices of G, such that $|S| = z = \lfloor \frac{n}{\log_2^2 n} \rfloor$. Let G_S be the graph induced on G by the vertices of S. We now colour G using the following algorithm.

Algorithm 3.1 COLOUR-BOLLOBÁS

```
1: while |V(G) − {v ∈ V(G)|v is coloured}| > z, do
2:     pick z vertices from {v ∈ V(G)|v is coloured} and call this set S,
3:     if S has an independent set of size k(z), then
4:         colour this set with a new colour,
5:     else
6:         FAIL
7:     end if
8: end while
9: colour each uncoloured vertex of V(G) with a new colour.
```

Lemma 3.3.4. *With high probability, Algorithm 3.1 finishes and does not fail.*

Proof. According to Lemma 3.3.3, G_S fails to contain an independent set of size $k(z) \sim k(n)$ with probability at most $e^{-z^{2+o(1)}} = e^{-n^{2+o(1)}}$. Applying the Union Bound (see page 24) gives us

$$\mathbb{P}\left[\exists S \text{ s.t. } \alpha(G_S) < k(z)\right] < \binom{n}{z} e^{-n^{2+o(1)}} = o(1).$$

Therefore, with high probability, Algorithm 3.1 does not fail in line 6 of the algorithm. □

Algorithm 3.1 creates a colour class of size $k(z) \sim k(n)$ during every WHILE loop. The only vertices not in colour classes of size $k(n)$ are those coloured in line 9. There are

no more than z of these. Therefore, Algorithm 3.1 colours G using at most $\frac{n}{k(n)} + z = (1 + o(1))\frac{n}{2\log_2 n}$ colours.

□

We note that although this is an existence proof without an explicit construction of a colouring, it can be translated to a practical $n^{\theta(\log n)}$-time algorithm [39]. We will discuss colouring algorithms in Section 3.5. The result by Bollobás was improved in 1990 by McDiarmid to obtain $\chi(G(n,p)) = \frac{n}{2\log_b n - \log_b \log_b n + O(1)}$ with high probability [193].

In 1987, Shamir and Spencer published an important result on the concentration of the chromatic number. They used martingales to show that, for fixed p, the chromatic number $\chi(G(n,p))$ is concentrated on $\sqrt{(}n)\beta(n)$ consecutive integers, where $\beta(n) \to \infty$ arbitrarily slowly. For $d < n^{1/6}$, they showed that $\chi(G(n,p))$ is concentrated on five consecutive integers with high probability [245].

Now let us look at sparse graphs. In 1991, Łuczak proved the following related two theorems about sparse graphs.

Theorem 3.3.5. *[182] With a probability of at least $1 - n^{-2}$, the graph $G \in G(n,p)$ with $C/n < p < \log^{-7} n$ contains $n\log^{-5} d/k_0$ independent sets of size*

$$k_0 = \left\lfloor \frac{2n}{d}(\log d - \log\log d + 1 - \log 2 - \varepsilon) \right\rfloor, \quad \varepsilon > 0.$$

Theorem 3.3.6. *[182] There exists a constant d_0, such that for $d > d_0$ and $p(n) \to 0$, it follows that*

$$\frac{d}{2\log d}\left(1 + \frac{\log\log d - 1}{\log d}\right) < \chi(G) < \frac{d}{2\log d}\left(1 + \frac{30\log\log d}{\log d}\right)$$

with high probability.

Theorem 3.3.5 was proved using martingales, and Theorem 3.3.6 was proved using a combination of Theorem 3.3.5 and a so-called *expose-and-merge* technique that we will

introduce in Section 3.4 and apply to prove a new result in Chapter 4.

3.4 Expose-and-merge

The *expose-and-merge* method for studying random graphs was introduced by Matula in 1987 [191]. This paper improved Grimmett and McDiarmid's [134] result that $\chi(G(n,p)) \sim \frac{an}{\log_b n}$, for $a = 1$ to $a = \frac{2}{3}$. Bollobás then improved it to its tight value of $a = \frac{1}{2}$. In 1990, Matula and Kučera revisited the expose-and-merge method from [191], to derive an alternative proof of $a = \frac{1}{2}$.

The expose-and-merge method allows one to explore independently generated random subgraphs followed by a merger into the final random graph $G(n,p)$.

Let p be fixed and let $G \sim \mathcal{G}(n,p)$. Additionally, let $z = \frac{n}{\log_b^{10} n}$. As in the proof of Theorem 3.3.3, we look at one random subgraph induced by a vertex set of size z at the time. For a subgraph of this size, a "large enough" independent set is also a "large enough" independent set in G. As in the proof for Theorem 3.3.3, we repeatedly select sets S of size z from $V(G)$ and colour some of them. This time, however, we do it without the freedom of knowing that *all* such sets in G have sufficiently large independent subsets. We can do this by ensuring that the different sets S induce subgraphs that are independent and have a $\mathcal{G}(z,p)$ distribution. When proving Theorem 3.3.3, this was not true, because removing independent sets from G destroys its $G \sim \mathcal{G}(n,p)$ distribution. One circumvents this problem by generating the subgraphs on the S-sets independently. This is the so-called *expose* part. Because different S-sets overlap, some pairs of vertices can occur in more than one of them. To generate the S-sets independently of each other, they must be allowed to contradict: for example, if vertex i and j are in the S-sets S_q and S_t (we use these subscripts here for clarity, but they will normally be omitted for ease of reading), then it is possible that $(i,j) \in E(G_{S_q})$, but $(i,j) \notin E(G_{S_t})$. For the final graph G to be distributed as $\mathcal{G}(n,p)$, only the first time a pair of vertices i and j is exposed will determine whether or

not $(i, j) \in E(G)$.

We start by labelling all vertex pairs as "open". Let *assigning* edges to a set of vertex pairs denote picking a subset of the vertex pairs such that each pair belongs to the subset independently with probability p. *Closing* a set of vertex pairs means labelling them all as "closed" (some might already be closed) and adding only the previously *open* pairs to $E(G)$. Thus, assigning edges to a set of vertex pairs creates temporary edges and non-edges on them, and closing them makes some of them permanent but rejects others. Every vertex pair starts out as "open", and at some point in the process it is assigned an edge or non-edge for the first time. This will be the state of the vertex pair in the final graph. Every subsequent time it has an edge or non-edge assigned to it, this becomes its temporary state until it is closed again. When we refer to a *closed edge*, we mean a pair of vertices that was assigned an edge the first time. So, if (i, j) is a closed edge, it implies that $(i, j) \in E(G(n, p))$ of the final graph $G(n, p)$. A *closed non-edge* is defined similarly, and a *closed pair* is either a closed edge or a closed non-edge.

We present the version of the algorithm given in [192]. Our proof of Theorem 4.4.1 is based on this proof, but is very technical and may be complicated to read. The proof by Matula and Kučera will help readers get used to the method.

Theorem 3.4.1. *[192] For $0 \leq p \leq 1$ constant, we have that*

$$\chi(G(n,p)) \leq \frac{n}{2\log_b n}\left(1 + \frac{12\log_b \log_b n}{\log_b n}\right).$$

Proof. We use the following crucial lemma, which is proven in [192] by second-moment methods.

Lemma 3.4.2. *[192]For $0 \leq p \leq 1$, $\epsilon > 0$, and $\alpha > 2$ constant, the probability that $G(n, p)$ contains at least $\frac{n}{\log_b^\alpha n}$ disjoint independent sets of size*

$$s = \lfloor 2\log_2 n - 2\log_b \log_b n + 2\log_b 2 - 1 - \epsilon \rfloor$$

is $1 - O(\log_b^{-2\alpha+4} n)$.

Let

$$z = \left\lfloor \frac{n}{\log_b^{10} n} \right\rfloor,$$
$$\overline{z} = \left\lfloor \frac{n}{\log_b^{2} n} \right\rfloor,$$
$$u = \left\lfloor \frac{z}{\log_b^{3} z} \right\rfloor,$$
$$w = \left\lfloor 2\log_b z - 2\log_b \log_b z + 2\log_b e - 2\log_b 2 - 1 - \epsilon \right\rfloor.$$

We define and run the following algorithm, called Colour-MK, which takes a vertex set $V(G) = \{1, \ldots, n\}$ and empty edge set $E(G) = \varnothing$ as input, and creates and colours an instance of $G(n,p)$ on that vertex set by adding edges to $E(G)$ at random and colouring vertices as it goes along.

Lemma 3.4.3. *[192] The graph $G(V,E)$ generated by Algorithm 3.2 is distributed as $\mathcal{G}(n,p)$.*

Proof. Every edge is inserted independently into $E(G)$ with probability p exactly once and is then closed. This happens either in line 5 and line 15, or in line 21. After an edge is closed it is never inserted to or deleted from E. Additionally, it is never reopened. □

Lemma 3.4.4. *[192] At the end of Algorithm 3.2, no edge has endpoints of the same colour.*

Proof. A monochromatic edge could only appear as a result of the colouring step in line 12, where all vertices of ζ_i receive the same colour. In all other colouring steps, no two vertices receive the same colour. In line 12, ζ_i is an independent set with no closed edges. Therefore, it is a permanent independent set in G after its edges are closed in line 15. □

Algorithm 3.2 COLOUR-MK

mark all vertices of $V(G)$ as uncoloured;
mark all pairs of vertices of $V(G)$ as open;
while $|V(G) - \{v \in V(G) | v \text{ is coloured}\}| > \overline{z}$, **do**
    pick, uniformly at random, z vertices from $V - \{v | v \text{ is coloured}\}$ and call this set S;
    assign edges to S;
    if S has u independent sets of size w, **then**
        label these independent sets $\zeta_1, \ldots, \zeta_u$;
        for $i = 1$ to u **do**
            if ζ_i contains a closed pair of vertices **then**
                colour each of the vertices of ζ_i with a new colour;
            else
                colour the vertices of ζ_i, all with the same new colour;
            end if
        end for
        close the vertex pairs of S;
    else
        FAIL
    end if
end while
colour the uncoloured vertices with a new colour each;
assign edges to all open pairs and close them.

Let a *loop* be the sequence of statements in lines 3–19. Let a *succesful* iteration of the loop be one that does not result in FAIL in line 17.

Lemma 3.4.5. *[192] There is a constant δ such that the probability that in each iteration the number of closed pairs in S is less than $\frac{n^2\delta}{\log_b^8 n}$ tends to 1.*

We will not prove Lemma 3.4.5. It is straightforward to prove using second-moment methods, and we will state a very similar lemma (which we will prove), inspired by this one, in our proof of Theorem 4.4.1.

Lemma 3.4.6. *[192] With high probability, the number of sets ζ_i which are coloured in line 10 is $O(\frac{n}{\log_b^3 n})$.*

Again, we will present a similar lemma and proof in the proof of Theorem 4.4.1.

Lemma 3.4.3 asserts that Algorithm 3.2 outputs G with distribution $\mathcal{G}(n,p)$. Lemma 3.4.4 asserts that G is coloured properly. The number of colours used in line 12 is at most $\frac{n}{w}$, and the number of colours used in line 20 is at most $\overline{z}$. In view of Lemma 3.4.6, $wO(\frac{n}{\log_b^3 n}) = O(\frac{n}{\log_b^2 n})$ colours are used in line 10. If c is a constant, it follows that

$$w \geq 2\log_b z - 2\log_b\log_b z + c = 2\log_b n - 2(10+1)\log_b\log_b n + O(\log_b\log_b\log_b n).$$

Therefore, the number of colours used by Algorithm 3.2 to colour G is

$$\begin{aligned}\frac{n}{w} + O(\overline{z}) + O(\frac{n}{\log_b^2 n}) &= \frac{n}{2\log_b n}\left[1 + \frac{11\log_b\log_b n}{\log_b n} + O\left(\frac{\log_b\log_b\log_b n}{\log_b n}\right)\right] \\ &< \frac{n}{2\log_b n}\left(1 + \frac{12\log_b\log_b n}{\log_b n}\right).\end{aligned}$$

□

3.5 Colouring algorithms

In this section we will give a short overview of colouring algorithms. Sometimes algorithms are needed when the chromatic number has a practical applications [141, 278], and sometimes they are of theoretical value when they provide previously unknown bounds on the chromatic number, by constructing an explicit colouring. For example, in Sections 3.3 and 3.4 we saw algorithms that coloured $G(n, p)$. These algorithms (Algorithm 3.1 and 3.2) were not presented in a directly applicable way. Furthermore, they constructed $G(n, p)$ in the process, so they cannot be used on a given graph G. This section should give the reader a flavour of algorithmic approaches to the colouring problem, and we encourage them to compare this to an "unconstrained" version of colouring problem, *community detection*, in Chapter 5. For a more thorough overview of colouring algorithms, see, for example, [119, 165].

Karp proved in 1972 that deciding whether a given graph is k-colourable, for a given $k \geq 3$, is NP-complete [156].[2]. In 1976, Garey and Johnson proved that coming close to the real chromatic number is just as hard as finding the actual chromatic number [121] More precisely, if there exists a polynomial time algorithm A, that colours a graph G in $A(G)$ colours, such that $A(G) \leq r_1\chi_G + r_2$, for constants r_1 and r_2 are constants, then there exists a polynomial time algorithm A' such that $A'(G) = \chi_G$. Feige and Kilian proved that it is hard to approximate the chromatic number to within $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$ [105]. Even when it is known that G can be coloured with few colours, the problem remains hard. Colouring a graph G such that $\chi_G = 3$ with 4 colours is an NP-complete problem [138, 160]. For an overview of colouring algorithms, see references [119, 169, 202]. In this subsection, we only discuss a few algorithms in relation to random graphs and probabilistic methods.

[2]Deciding whether a graph is 2-colourable is easy and can be done in $O(m)$ time, because, after a first vertex of each component has been coloured, each vertex connected to an already coloured vertex only has one colour choice left. Therefore, deciding whether the graph is 2-colourable boils down to searching the graph once.

3.5.1 Greedy approach to k-colouring

The greedy algorithm [42, 134] is perhaps the simplest graph colouring algorithm imaginable. In Chapter 8 we will use this algorithm to colour brain networks (see page 157). Given a graph G and an (ordered) set of vertices $v_1, \ldots, v_n$, a greedy algorithm colours the vertices in their fixed order, and it assigns to each vertex the lowest (integer) colour that is not yet in use by one of its neighbours. This algorithm immediately gives an upper bound on the chromatic number in terms of the maximum degree. Namely, $\chi_G \leq \Delta + 1$ (because if a vertex has Δ neighbours, they cannot use all of the colours $1, \ldots, \Delta + 1$). The performance of this algorithm depends on the initial ordering of the vertices. There always exists an ordering such that the greedy algorithm uses exactly χ_G colours (if the vertices are already ordered by the colours of a proper χ_G-colouring). However, if the initial ordering is chosen badly, then the algorithm can be off by a factor of $O(n)$.

The greedy algorithm is a so-called *online* algorithm [161], because the colour choice for each vertex depends only on previously seen vertices and remains unchanged once it has been decided. In the case of $G \sim \mathcal{G}(n, p)$, this implies that one can use a vertex-exposure approach: at each step, one reveals one vertex and assigns its edges to all of the already-exposed vertices. Using that approach and considering $G(n, 1/2)$, it has been proven that almost every graph will be coloured by the greedy algorithm in no more than $n/\log_2 n - 3\log_2\log_2 n$ colours [134]. This is impressive for such a simple algorithm, as this is only twice the expected number. Subsequently, McDiarmid showed for almost every graph that, no matter what ordering is chosen, the greedy algorithm can colour the graph using at most $(1 + 5\log_2\log_2 n/\log_2 n)n/\log_2 n$ colours.

However, the greedy algorithm fails on sparse graphs. It was shown in [225] that the greedy algorithm uses $(1 + o(1))\log_2\log_2 n$ colours on $G(n, c/n)$ with high probability, even though almost all of these graphs have a constant χ_G.

Krivelevich and Vu used a slight alteration of the greedy algorithm to formulate a method that approximates the chromatic number of most graphs in polynomial expected

time.

Theorem 3.5.1. *[172] Let $G \sim \mathcal{G}(n,p)$, where $n^{-1/2+\epsilon} < p < .99$ and $\epsilon > 0$. Then there exists a deterministic colouring algorithm that approximates χ_G within a factor $O(\frac{\sqrt{np}}{\log n})$ and has polynomial expected running time.*

Let the *auxiliary matrix* $\mathbf{M}$ have components

$$\mathbf{M}_{ij} = \begin{cases} -\frac{1-p}{p}, & \text{if } (i,j) \in E(G), \\ 1, & \text{otherwise.} \end{cases} \tag{3.1}$$

Here we consider the case where $p = 1/2$, so $-\frac{1-p}{p} = -1$. We have that $\alpha(G) \leq \lambda_1(\mathbf{M})$ [172], where $\lambda_1(\mathbf{M})$ is the leading eigenvalue of the matrix $\mathbf{M}$. Let $GA(G)$ be the number of colours that the greedy algorithm colours G in. The algorithm is given as follows.

- **Step 1.** Run the greedy algorithm on G. If $GA(G) > 2n/\log_2 n$, go to Step 3.
- **Step 2.** If $\lambda_1(M) < 6\sqrt{n \log n}$, output $GA(G)$.
- **Step 3.** Find an optimal colouring by exhaustive search and output it.

Step 3 takes $O(n^n \text{poly}(n))$ time, but it only occurs with probability $O(n^{-n})$ [172]. Step 1 and 2 clearly take only polynomial time. For $a > 1/2$, no algorithms with good approximation ratios (the ratio between the number of colours used by the algorithm and the true chromatic number) and expected polynomial running times have been found yet for $G(n, n^{-a})$.

3.5.2 Deciding k-colourability of random graphs

The problem of deciding whether a random graph $G(n, .5)$ is k-colourable is NP-complete, but the answer is almost always "no".

Lemma 3.5.2. *[169] Fix a positive integer k. With probability* $1 - 2^{-\Theta(n^2)}$*, the graph* $G = G(n, 1/2)$ *is not k-colourable.*

Proof. For G to be k-colourable, it must contain an independent set of size $\geq n/k$. This occurs with probability

$$\mathbb{P}\left[\alpha(G(n,1/2)) \geq n/k\right] \leq \binom{n}{n/k}\frac{1}{2}^{\binom{n/k}{2}} < 2^n 2^{-\Theta(n^2)} = 2^{-\Theta(n^2)}.$$

□

Because the event that a graph is k-colourable is so rare, one would expect that it should at least be possible to decide k-colourability in polynomial expected time. In fact, this can be done in constant expected time, using a backtracking algorithm proposed by Wilf in 1984 [280]. However, in sparse graphs, this algorithm has an exponential expected running time of $e^{\Theta(1/p)}$ [23].

In 2002, Krivelevich found an algorithm for deciding k-colourability in expected polynomial time for $G(n,p)$, for $p(n) > c/n$ [173]. It uses the so-called *vector chromatic number* [155] (a graph parameter whose value, when large enough, can serve as a certificate for non-k-colourability) is both sufficiently large with high probability, and computable in polynomial time.

3.6 Equitable chromatic number

The chromatic number of a graph is the smallest number of colours needed for such any proper colouring. An *equitable colouring* of a graph is a proper vertex colouring such that the sizes of any two colour classes differ by at most one. The *equitable chromatic number* $\chi_=(G(n,p))$ of G is defined as the least positive integer k for which there exists an equitable colouring of a graph G with k colours. The property of being equitably k-colourable is not monotone: it is possible that a graph G is equitably k-colourable, but not equitably $(k+1)$-

colourable. The *equitable chromatic threshold* $\chi^*_=(G(n,p))$ of G is the least positive integer k such that for every $k_0 \geq k$ there exists an equitable colouring of G with k_0 colours.

The introduction of the *equitable chromatic number* has been attributed to Erdős in the literature. The first important result on equitable colourings is the following theorem by Hajnal and Szemerédi [140]. This theorem settled a conjecture by Erdős, although partial results had already been obtained [84, 135, 288]. The result was reproved more elegantly in [162], which also provides an algorithm that runs in $O(n^5)$ time.

Theorem 3.6.1. *[140] The equitable chromatic threshold* $\chi^*_=(G) \leq \Delta + 1$.

For other deterministic results, see references [5, 168, 180, 222].

Equitable colourings have applications in scheduling computer processes [29, 250] and other routing and load-sharing problems [269].

For equitable colourings of $G(n,p)$, Krivelevich and Patkós [170] proved the following Theorems 3.6.2 and 3.6.3.

Theorem 3.6.2. *[170] If* $n^{-1/5+\epsilon} < p < 0.99$ *for some* $\epsilon > 0$*, then almost surely*

$$\chi(G(n,p)) \leq \chi_=(G(n,p)) \leq (1+o(1))\chi(G(n,p)).$$

We omit the proof for Theorem 3.6.2, as it is rather complicated and involves mechanisms to create a colouring and then move vertices around between colour classes to create classes of the right size. We will do something similar in our proof in Chapter 4 and we will discuss the relevant techniques at that time.

Theorem 3.6.3. *[170] There exists a constant* C *such that if* $C/n < p < 1-\epsilon$*, for* $\epsilon > 0$ *then almost surely*

$$\chi(G(n,p)) \leq \chi_=(G(n,p)) \leq (2+o(1))\chi(G(n,p)).$$

Proof. The left inequality is trivially true. We outline the proof from [170] for the right inequality, but we omit the case in which $p = C/n$. It uses an adapted version of the

greedy colouring algorithm and edge exposure (see Section 3.5.1). Fix an integer k, which will be the future number of colour classes. Start with an empty graph G on a vertex set $V(G)$, and partition $V = V_1 \cup V_2 \cup \ldots \cup V_{\lceil \frac{n}{k} \rceil}$, such that $k = |V_1| = |V_2| = \ldots = |V_{\lfloor \frac{n}{k} \rfloor}|$, and $|V_{\lfloor \frac{n}{k} \rfloor}| = n - k\lfloor \frac{n}{k} \rfloor$. The greedy algorithm consists of $\lceil \frac{n}{k} \rceil$ rounds, and the vertices of one set V_i are coloured greedily in round i. Suppose that after round $i-1$ we have a proper colouring of $G(V_1 \cup V_2 \cup \ldots \cup V_{i-1})$, then we say that round i is *succesful* if we can extend the existing colouring to a colouring of $G(V_1 \cup V_2 \cup \ldots \cup V_i)$. The aim is to find a k-colouring such that we use exactly k colours inside all of the sets V_i (and $n - k\lfloor \frac{n}{k} \rfloor$ colours for $V_{\lceil \frac{n}{k} \rceil}$).

Define an auxiliary random bipartite graph B_i on $2k$ vertices. Let k vertices on one side of B_i represent the vertices of V_i and let the k vertices on the other side of B_i represent the k colours. There is an edge (v, c) in B_i if and only if it is possible to colour vertex v with colour c (*i.e.*, if there are no edges between vertex v and any vertices in $V_1 \cup V_2 \cup \ldots \cup V_{i-1}$ coloured previously with colour c). Thus, $B_i \sim \mathcal{B}(k, k, p_B)$ (see page 16) is a random bipartite graph with edge probability $p_B = (1-p)^{i-1}$.

Lemma 3.6.4. *[Remark 4.3 in [149] (page 84)] The probability that there is no perfect matching in a random bipartite graph with $k + k$ vertices and edge probability p_B is $O(ke^{-kp_B})$.*

Thus, by the Union Bound, Lemma 2.11.2, the probability that any one of the rounds fails is $O(ne^{-k(1-p)^{\frac{n}{k}}}) = O(e^{\log n - k(1-p)^{\frac{n}{k}}})$. Again using the Union Bound, the probability that the algorithm fails for any $k' \geq k$ is $O((n-k)ne^{-k(1-p)^{\frac{n}{k}}}) = O(e^{2\log n - k(1-p)^{\frac{n}{k}}})$. Therefore, if there exists a k such that $k(1-p)^{\frac{n}{k}} = \omega(\log n)$, then $\chi^*_=(G(n,p)) \leq k$ with high probability.

When $p = n^{-o(1)}$, this works for $k = \frac{n}{\log_b n - 2.1 \log_b \log_b d}$.

When $n^{-1=\epsilon} \leq p \leq \log^{-8} n$, with $\epsilon > 0$, this works for $k = \frac{d}{\log d - 2.1 \log\log d}$.

For $p = C/n$, we omit the proof for the same reasons as we did for Theorem 3.6.2. □

Krivelevich and Patkós introduced the following conjecture, which is the topic of Chapter 4.

Conjecture 3.6.5. *[170] There exists a constant C such that if $C/n < p < 1 - \epsilon$, for $\epsilon > 0$, then almost surely*

$$\chi(G(n,p)) \leq \chi_{=}(G(n,p)) \leq \chi^{*}_{=}(G(n,p)) = (1 + o(1))\chi(G(n,p))$$

holds (where we note that the first two inequalities are a trivial consequence).

We resolve Conjecture 3.6.5 in Chapter 4 for a wide range of p.

Chapter 4

Equitable Colourings of Random Graphs

This chapter represents work that I carried out with my supervisor, A. D. Scott. An early version of Section 4.3 appeared as part of a thesis, by me, for the degree of *MSc in Mathematics and Foundations of Computer Science*, which I obtained at the University of Oxford (under the supervision of A. D. Scott), in 2009. The proof in Section 4.4 is the more difficult part of the conjecture and was completed during the course of my DPhil.

4.1 Introduction

Chapter 3 gives the introductory material for this chapter, and contains several known results and proofs that provide the reader with some familiarity to the techniques used. This chapter is concerned with proving that $\chi^*_=(G(n,p)) \leq (1+o(1))\chi(G(n,p))$ for a wide range of p. The proof is split up into two theorems, one for the dense case in Section 4.3 and the sparse case in Section 4.4. We repeat the original conjecture by Krivelevich and Patkós.

Conjecture 3.6.5. [170] *There exists a constant C such that if $C/n < p < 0.99$, then almost*

surely

$$\chi(G(n,p)) \leq \chi_{=}(G(n,p)) \leq \chi_{=}^{*}(G(n,p)) = (1+o(1))\chi(G(n,p))$$

holds (the first two inequalities are true by definition).

We will prove this result for all but the sparsest case. Theorem 4.3.1 proves the conjecture for $n^{-\frac{2}{5}}\log^{\frac{6}{5}} n \ll p(n) \leq 1-\epsilon$, and Theorem 4.4.1 proves the conjecture for $p = n^{-(1-\eta)}$. The full Theorem is as follows.

Theorem 4.1.1. *If $n^{-1+\epsilon_1} < p < 1-\epsilon_2$, for $\epsilon_1, \epsilon_2 > 0$, then almost surely*

$$\chi(G(n,p)) \leq \chi_{=}(G(n,p)) \leq \chi_{=}^{*}(G(n,p)) = (1+o(1))\chi(G(n,p))$$

holds. (The first two inequalities are true by definition.)

We start by presenting some tools that we will use, in addition to the tools that we presented in Section 2.11.

In both the proof of Theorem 4.3.1 (dense case) and of Theorem 4.4.1 (sparse case), $V(G)$ is partitioned into four disjoint sets: $V(G) = A_L \cup A_R \cup B_L \cup B_R$. The sets A_L and A_R are big: they take up almost half of the vertices each, and B_L and B_R are small: they are of size $\Theta\left(\frac{n}{\log^{\alpha} n}\right)$ for some positive constant α. We describe algorithms that contain multiple different stages. During those stages, the algorithm is usually performed on one set of A_L, A_R and one set of B_L, B_R. To distinguish between the different combinations, we introduce the terms *parallel* and *diagonal* stages. Parallel stages are performed on both A_L, B_L or both A_R, B_R, and diagonal stages are performed on both A_L, B_R or both A_R, B_L. The process as a whole is symmetric, so the stages of the algorithm are always performed twice. Either, they are performed on A_L, B_L and A_R, B_R, respectively (parallel stages), or they are performed on A_L, B_R and A_R, B_L, respectively. Stage 1 and 3 are parallel stages, and Stage 2 is a diagonal stage.

4.2 Tools and definitions

We will also use the following theorems.

Theorem 4.2.1. *[171] Let $G \in \mathcal{G}(n,p)$. Let $p(n)$ satisfy $n^{-\frac{2}{5}}\log^{\frac{6}{5}} n \ll p(n) \leq 1-\epsilon$ for $0<\epsilon<1$, and let*

$$k_0 = \max\left\{k : \binom{n}{k}(1-p)^{\binom{k}{2}} \geq n^4\right\}.$$

Then

$$\mathbb{P}\left[\alpha(G) < k_0\right] = \exp\left[-\Omega\left(n^2/k_0^4 p\right)\right].$$

It is known that $2\log_{1/(1-p)} n \geq k_0 \geq 2\log_{1/(1-p)} n - C'\log_{1/(1-p)}\log_{1/(1-p)}(np)$, for some constant C'. This theorem ensures that $G(n,p)$ has at least one large independent set.

Theorem 4.2.2. *[[220], see also page 70 in [40]]* The min-cut max-flow theorem (vertex version). *We define a (directed) graph $G(V,E)$, with a source vertex, a, a sink vertex, b, a flow function $f : E \to R^+$, and a capacity function $c : V \to R^+$. The flow must satisfy*

$$\sum_{y|(y,a)\in E} f_{(y,a)} = 0,$$

$$\sum_{y|(b,y)\in E} f_{(b,y)} = 0,$$

$$\sum_{y|(x,y)\in E} f_{(x,y)} = \sum_{y|(y,x)\in E} f_{(y,x)} \leq c_x, \quad \forall x \in V \setminus \{a,b\}.$$

We define a cut $C \subseteq V \setminus \{a,b\}$ to be a set of vertices such that $G \setminus C$ does not contain a path from a to b. The capacity of a cut is defined as the sum of capacities of the vertices in the cut set. It then follows that the maximal flow value from a to b is equal to the minimum of the capacities of cuts separating a from b.

Let an (n,d,δ)-expander be an n-vertex d-regular graph G, such that for every set $S \subseteq V(G)$, $|S| \leq \frac{1}{2}|V(G)|$, there are at least $\delta d|S|$ edges connecting S and $V(G) \setminus S$.

Theorem 4.2.3. *[224] [6] There exists a fixed $\delta > 0$, such that for any $d \geq 3$ and a sufficiently large even integer n, there is an (n, d, δ)-expander.*

4.3 Dense Graphs

We let

$$\alpha = 50,$$

$$\beta = 1000.$$

Theorem 4.3.1. *Let $G \in \mathcal{G}(n,p)$ and let $p(n)$ satisfy $n^{-\frac{2}{5}} \log^{\frac{6}{5}} n \ll p(n) \leq 1 - \epsilon$, where $0 < \epsilon < 1$. Then, with high probability,*

$$\chi_{=}^{*}(G(n,p)) \leq \frac{n}{2\log_b n - \beta \log_b \log_b d}.$$

Proof. We first prove the statement for $\chi_{=}(G(n,p))$, and then we will generalise the result to $\chi_{=}^{*}(G(n,p))$ at the end of the proof. For $k > 2\chi(G(n,p))$, we can colour $G(n,p)$ by Theorem 3.6.3, so suppose that $\chi(G(n,p)) \leq k \leq 2\chi(G(n,p))$. We suppose that $s|n$. (If $k \nmid n$, then we have colour classes of two sizes that differ by one; these can be handled by a straightforward modification of the algorithm, deciding whether each colour class has size s or size $s+1$ in the course of the algorithm.)

Let

$$s = \lceil 2\log_b n - \beta \log_b \log_b d \rceil.$$

We begin by marking all vertex pairs as "open", and partitioning $V(G) = A_L \cup A_R \cup B_L \cup B_R$ such that:

- $|B_L| \sim |B_R| \sim \frac{n}{\log^{\alpha} n}$,
- $|A_L| \sim |A_R| \sim \frac{n}{2}$,
- $|A_L \cup B_L|$ and $|A_R \cup B_R|$ are both divisible by s.

Definition 4.3.2. (A, B, s)**-Gadget.** *An* (A, B, s)-gadget *of* gadget-size *t is a subgraph of a graph G defined as follows. An (A, B, s)-gadget is a set of t independent sets $\sigma_1, \sigma_2, \ldots, \sigma_t$ in $A \cup B$ such that:*

- *each independent set σ_i has exactly one (unique) vertex in B,*
- *for $i = 1, 2, \ldots, t-1$, we have $|\sigma_i \cap \sigma_{i+1} \cap A| = 1$,*
- *for i, j, such that $|i - j| > 1$, $|\sigma_i \cap \sigma_j| = 0$,*
- $|\sigma_i \setminus \left(\bigcup_{j \neq i} \sigma_j\right)| = s$.

See Figure 4.1 for an illustration. A and B refer to either A_L and B_L or A_R and B_R.

Definition 4.3.3. Joint. *We call the vertices that two sets σ_l and σ_{l+1} have in common the* joints *of the (A, B, s)-gadget.*

Definition 4.3.4. Rim. *We call the vertices of an (A, B, s)-gadget that lie in B the* rim.

Definition 4.3.5. Inner sets. *If we ignore the rim and the joints of an (A, B, s)-gadget, then the fully disjoint independent sets in A that we are left with are called the* inner sets *and are denoted $\zeta_1, \zeta_2, \ldots, \zeta_t$. The inner sets all have size $s - 1$.*

We need the following two lemmas.

Lemma 4.3.6. *Given an (A, B, s)-gadget, let v_i be an arbitrary vertex in the rim of the gadget. The (A, B, s)-gadget can then be transformed into t pairwise disjoint proper colour classes of size s, colouring all of the inner sets, the joints, and v_i (but not the rest of the rim).*

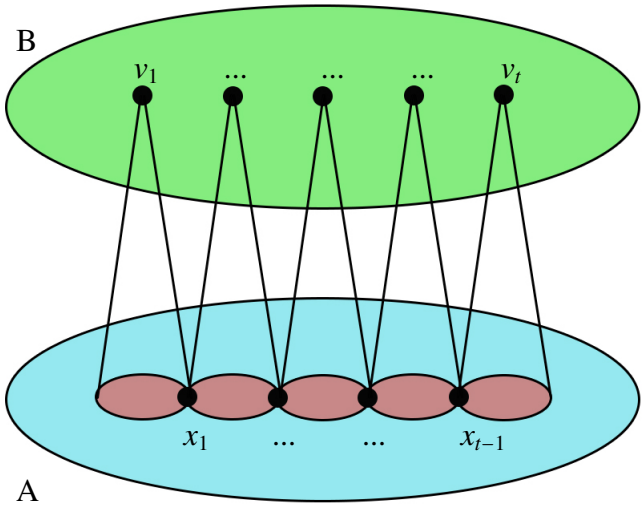


Figure 4.1: An (A, B, s)-*gadget* consists of a group of independent sets that have exactly one vertex in B each. Some independent sets are connected through one vertex such that, together, they form a chain.

Proof. Let $\zeta_1, \ldots, \zeta_t$ be the inner sets of a certain gadget. Colour each ζ_j with a new colour. Add v_i to the colour class of ζ_i. Now, add all joints x_j with $j < i$ to the colour classes of ζ_j and all joints x_j with $j \geq i$ to the colour classes of ζ_{j+1}. Every colour class contains all nodes in a set ζ_j, which has size $s - 1$, plus one extra vertex. See Figure 4.2. □

Lemma 4.3.7. *Let c_1, c_2 be positive integers, such that $c_1 > c_2$. Let $\epsilon > 0$, $p < 1 - \epsilon$, and $G \in \mathcal{G}(n, p)$. Let $D \subseteq V(G)$ be a set of size $\Omega(\frac{n}{\log_b^{c_1} n})$. Then, with high probability, for all sets $C = \{v_1, \ldots, v_c\} \subseteq V(G) \setminus D$ of size $c < c_2$, we have that*

$$|\overline{\Gamma}(C) \cap D| = \Omega(|D|).$$

Proof. Let X_i be the indicator of the event that vertex $i \in D$ is in $\overline{\Gamma}(C)$, and let $X = \sum_{i=1}^{|D|} X_i$. Clearly these events are independent, and $\mathbb{E}[X] = |D|q^c$, where $c \leq c_2$, and $q \geq \epsilon$. Let $\lambda = \sqrt{|D| \log_b^{c_1+1} n}$. By Lemma 2.11.5, it follows that

$$\mathbb{P}\left[|\overline{\Gamma}(C) \cap D| < |D|q^c - \lambda\right] \leq e^{-\frac{\lambda^2}{2\mu}} = O(e^{-\log_b^{1+c_1} n}).$$

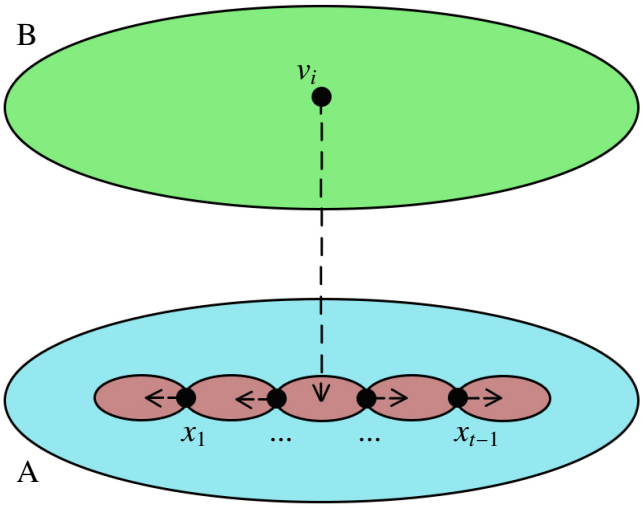


Figure 4.2: Creating colour classes from an (A, B, s)-gadget using all of the inner sets, joints, and exactly one (arbitrary) vertex from the rim.

The probability that this is true for all sets $C \subseteq V(G)$, $|C| \leq c_2$, is

$$\mathbb{P}\left[|\overline{\Gamma}(C) \cap D| < |D|q^c - \lambda : \forall C \subseteq V(G), |C| \leq c_2\right] = O(e^{-\log_b^{1+c_1} n}) \cdot \sum_{c=1}^{c_2} \binom{n}{c}$$
$$= O(n^{c_2} e^{-\log_b^{1+c_1} n}) = o\left(\frac{1}{n^{c_1-c_2}}\right).$$

□

We colour G in three stages.

Stage 1

Stage 1 is a parallel stage. We fix an $m_H = \min\left\{\frac{n}{\log_b^\alpha n}, \left\lfloor\frac{|B|}{2}\right\rfloor\right\}$. We now create a random hypergraph $H \in \mathcal{H}(|B|, m_H)$, independently of G. We pick H in the following way. Let $G_H \in \mathcal{G}(|B|, m_H, \rho_H)$ be a random bipartite graph in which one vertex set consists of the vertices of B (so either B_L or B_R) and the other vertex set is a set of m_H vertices representing the edges of H. The edges of G_H occur independently at random with probability $\rho_H = \frac{\log_b^{\alpha+2} n}{n}$. Let a vertex in H be part of an edge if and only if it is connected to a representing vertex

in G_H. This yields a random hypergraph H with m_H edges of edge-degree approximately $\log_b^2 n$. Note that multiple edges on the same vertex set are allowed.

Lemma 4.3.8. *With high probability, it is possible to create m_H (A, B, s)-gadgets such that the rim of each (A, B, s)-gadget forms exactly one edge in H and all edges of H are covered by an (A, B, s)-gadget in this way, and the gadgets are pairwise vertex-disjoint, apart from the rims.*

Proof. Observe first that, even though the number of (A_L, B_L, s)- and (A_R, B_R, s)-gadgets is, in both cases, m_H, their gadget sizes need not be equal. Using slightly sloppy notation, we let t be the number of vertices in a hyperedge in H, which is the gadget size.

By Lemma 2.11.5 and Lemma 2.11.2, the maximal size of a gadget, $t_{\max}$, is not too large, with high probability:

$$\mathbb{P}\left[t_{\max} > 2\mathbb{E}[t]\right] \leq m_H e^{-\Theta(\log_b^2 n)} = o\left(\frac{1}{n}\right). \tag{4.1}$$

Label all vertices of A as "unused". Each time we pick a vertex from A to be a joint or part of an inner set ζ_i, we mark the vertex as "used". For each hyper-edge $e = \{v_1, v_2, \ldots, v_t\}$ in H, we create an (A, B, s)-gadget as follows. For $l = 1, 2, \ldots, t-1$, pick an x_l from the common non-neighbourhood of v_l and v_{l+1}. These will be the joints of the gadget. Let $A^{un} = \{i \in A | i \text{ is unused}\}$. Now pick the independent inner set ζ_1 from the non-neighbourhood of v_1, x_1 in A^{un}, the inner set ζ_t from the non-neighbourhood of v_t, x_{t-1} in A^{un}, and the inner set ζ_i, for $i = 2, \ldots, t-1$ from the non-neighbourhood of v_i, x_{i-1}, x_i in A^{un}. The non-neighbourhoods in A^{un} are of size $\Omega(|A^{un}|)$, by Lemma 4.3.7, if a A^{un} remains sufficiently large (which we show below to be the case).

Since we need only m_H (A, B, s)-gadgets, we use at most $s \cdot m_H \cdot t_{\max}$ vertices from A to make our (A, B, s)-gadgets, which is $O\left(\frac{n}{\log_b^{\alpha-3} n}\right)$, with high probability. Therefore, $|A^{un}| = \Theta(|A|) = \Theta(n)$, with high probability. By Lemma 4.3.7 and Lemma 2.11.2, one of the non-neighburhoods of the needed 2 or 3 vertices in A^{un} fails to be sufficiently large with

probability

$$\begin{aligned}\mathbb{P}\big[|A^{un}\cap\overline{\Gamma}(v_i,x_{i-1},x_i)| = o(n), \text{ for any } v_i \text{ in any gadget} \mid |A^{un}| = \Theta(n)\big]\\ = m_H \cdot t_{\max} \cdot o\left(\frac{1}{n^{\alpha-3}}\right)\\ = o\left(\frac{1}{n}\right).\end{aligned} \tag{4.2}$$

One of the non-neighbourhoods of the needed 2 or 3 vertices in A^{un} contains no independent set of size s with probability

$$\begin{aligned}\mathbb{P}\big[\text{fail to find a } \zeta_i \mid |A^{un}| = \Theta(n)\big] = m_H \cdot t_{\max} \cdot \exp\left[-\Omega\left(\frac{n^2}{\log_b^4 p}\right)\right]\\ = \exp\left[-\Omega\left(\frac{n^2}{\log_b^4 p}\right)\right]\\ = o\left(\frac{1}{n}\right).\end{aligned} \tag{4.3}$$

by Theorem 4.2.1 and Lemma 2.11.2. □

Now colour each inner set in the (A, B, s)-gadget with a new colour. Because we do not colour the rim or the joints, every colour class has size $s - 1$ for now. We will add a vertex to them in Stage 2. The (A, B, s)-gadgets look like Figure 4.3.

Stage 2

Stage 2 is a diagonal stage, so that A and B, refer either to A_L and B_R or to A_R and B_L.

Lemma 4.3.9. *One can colour $A^{un} \cup B$ with u colours such that the colour classes each have size s, and all vertices are coloured except for m_H vertices in B.*

Proof. We can go on picking and colouring independent sets of size s until there are at most $\frac{n}{\log^{2\alpha} n}$ vertices left in A^{un}. We already checked the probability that all sufficiently large subsets of G contain a sufficiently large independent set in Equation (4.1). Call this set A'. For each vertex in A', look at its neighbourhood in B. This neighbourhood has a size of

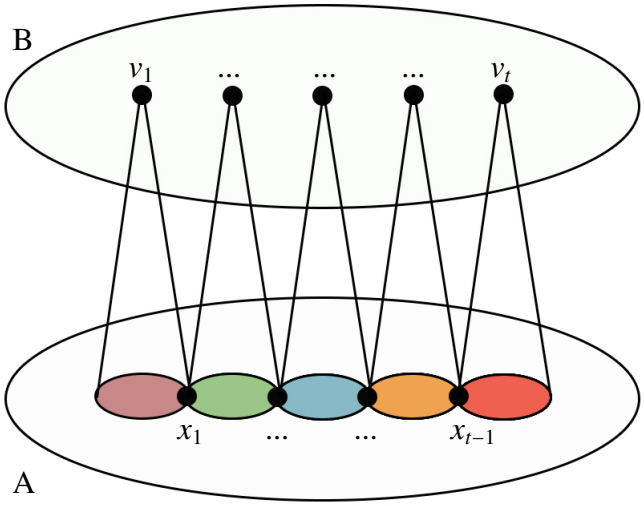


Figure 4.3: *All vertices in the* (A, B, s)*-gadget receive a colour except for the rim and joints. Every colour class has size* $s - 1$.

approximately $\frac{nq}{\log^{\alpha} n}$ by Lemma 4.3.7 and thus contains an independent set of size s. Then, we pick our next colour classes using one vertex from A' and $s-1$ vertices from B. We use at most $\frac{n}{\log^{2\alpha-1} n}$ vertices, so even for the last vertex of A' there are enough unused vertices in B for Lemma 4.3.7 to apply. We now pick independent sets of size s from B and colour them, until exactly m_H vertices are left. Recall that s divides $|A \cup B|$ and also recall that a gadget can be split into colour classes of size s using all of the inner sets and joints and exactly one vertex from the rim. It follows that we can keep picking and colouring independent sets of size s until the number of uncoloured vertices in B reaches exactly m_H. □

Stage 3

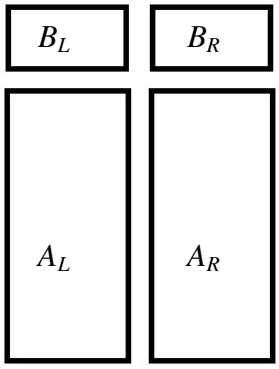
Stage 3 is a parallel stage, in which A, B refers to either A_L, B_L or A_R, B_R. We must pair each uncoloured vertex in B with exactly one (A, B, s)-gadget. Note that vertices in B can be part of multiple (A, B, s)-gadgets. We take the induced subgraph G'_H of G_H (as defined in Stage 1) by only considering the m_H uncoloured vertices from B. We have that $G'_H \in \mathcal{G}(m_H, m_H, \rho_H)$, because our choice of H is independent of which vertices ended up

uncoloured. By Lemma 3.6.4,

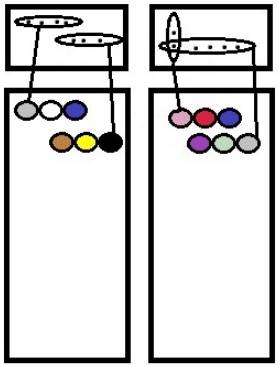
$$\begin{aligned}\mathbb{P}[G'_H \text{ has no perfect matching}] &= O(m_H e^{-m_H \rho_H}) \\ &= O\left(\frac{n}{\log_b^{\alpha} n} e^{-\log_b^2 n}\right) \\ &= o\left(\frac{1}{n}\right).\end{aligned} \tag{4.4}$$

Therefore, with high probability, G'_H has a perfect matching such that each uncoloured vertex in B is paired with an (A, B, s)-gadget, and we create the coloured independent sets as in Lemma 4.3.6.

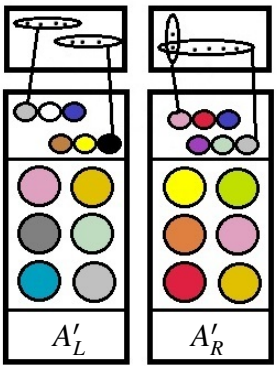
All that remains is to check that this process works for all $s' < s$. By Lemma 2.11.2, it is sufficient to confirm that the probabilities in Equations (4.1), (4.2), (4.3), and (4.4) are $o\left(\frac{1}{s}\right)$, and this is indeed the case. □



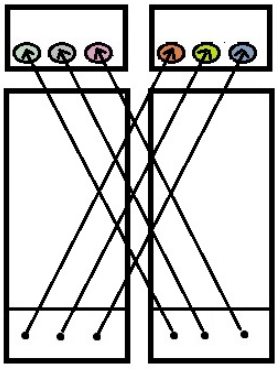
Split $V(G)$ into A_L, A_R, B_L, B_R.



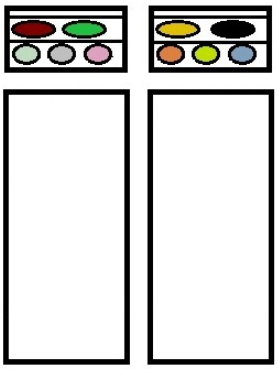
Create the (A, B, s)-gadgets. Stage 1.



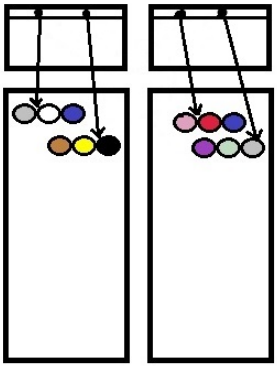
Colour almost all of the rest of A. Stage 2.



Form colour classes by finding independent sets of one vertex from A' and $s-1$ from B. Stage 2.



Colour almost all of the rest of B. Stage 2.



Match remaining uncoloured vertices of B up with an (A, B, s)-gadget each. Stage 3.

Figure 4.4: A cartoon of the colouring process for dense graphs.

4.4 Sparse Graphs

The proof of Theorem 4.4.1 has a lot of similarities to the proof of Theorem 4.3.1 (compare Figures 4.4 and 4.8, for example). We use the expose-and-merge technique discussed in Section 3.4 to find large enough independent sets, as we cannot guarantee that all sufficiently large subgraphs of G contain sufficiently large independent sets – *i.e.*, there is no sparse version of Lemma 4.2.1. This proof uses Theorem 3.3.5, which works on sparse graphs and finds many independent sets of sufficiently large size at once, as long as the edge distribution is independent, which is where the expose-and-merge technique comes in. The sparsity also allows us to merge independent sets into new independent sets by removing only a few 'bad' vertices, and letting sets of vertices 'flow' between colour classes.

The expose-and-merge technique that we use is similar to the one introduced in [191]. One difference is that in the original algorithm, any independent set with a closed vertex pair is discarded immediately, which we cannot afford. Instead, we check that there are not many closed edges in our independent set, and then we discard only those edges from it in order to turn it into an independent set in the final graph. We also need a more sophisticated version of the gadgets, as we will need a flow of multiple vertices running between sets, rather than single vertices. Instead of gadgets and inner sets, we will have special colour classes that have a graph imposed on them along the edges of which a limited flow of vertices is allowed. This will help us make up the sizes of the final colour classes in the very last step, just like the gadgets did in the proof of Theorem 4.3.1.

Theorem 4.4.1. *Let $G \in \mathcal{G}(n,p)$ and let $p = n^{-(1-\eta)}$ for $0 < \eta < 1$. Then, with high probability, $\chi^*_=(G) \sim \frac{d}{2\log d}$, where $d = np$.*

Proof. We let $s = \left\lfloor \frac{1}{p}(2\log d - 20\log\log d) \right\rfloor$ and suppose that $s|n$ and let $s = n/k$. (If $k \nmid n$, then we have colour classes of two sizes that differ by one; these can be handled by a straightforward modification of the algorithm, deciding whether each colour class has size s or size $s+1$ in the course of the algorithm.)

Figure 4.8 illustrates the entire colouring process with a cartoon, analogously to Figure 4.4 for the proof of Theorem 4.3.1. We use an expose-and-merge approach along the lines of [191], [192], and [182] (see Section 3.4).

We start out with a set of n vertices, and we assign edges as we go along such that we end up with a graph $G \in \mathcal{G}(n,p)$. We first prove that we can colour $G(n,p)$ with colour classes of size s. As in the proof of Theorem 4.3.1, for colour classes of size $s' < s/2$, we can colour $G(n,p)$ by Theorem 3.6.3, and at the end of the proof we will generalise our proof to colour classes of size s' such that $s/2 \leq s' \leq s$. We now define a set of values that we will use throughout the proof. Let

$$
\begin{aligned}
z &= \left\lfloor \frac{n}{\log^{17} n} \right\rfloor \\
\overline{z} &= \left\lfloor \frac{n}{\log^{13} n} \right\rfloor, \\
u &= \left\lfloor \frac{zp}{2\log^{6} zp} \right\rfloor, \\
w &= \left\lfloor \frac{1}{p}(2\log zp - 2\log\log zp) \right\rfloor, \\
t &= 2 \cdot \left\lfloor \frac{d\log^2 d}{2\log^5 n} \right\rfloor.
\end{aligned}
$$

Choose a 3-regular graph H with vertices $h_1, \ldots, h_t$ (we ensured that t is even), such that the property of Theorem 4.2.3 holds – *i.e.*, H is a $(t, 3, \delta)$-expander. Also, pick a constant $\delta' > 0$ such that $\frac{\delta'+1}{\delta'-1} - 1 < \delta$, which we will use later.

Just as in Section 3.4, let *assigning* edges to a set of vertex pairs denote picking a subset of the vertex pairs such that each pair belongs to the subset independently with probability p. *Closing* a set of vertex pairs means labelling them all as *closed* (some may already be *closed*) and adding only the previously *open* pairs to $E(G)$. Thus, *assigning* edges to a set of vertex pairs creates temporary edges and non-edges on them, and *closing* makes some

of them permanent but rejects others. Every vertex pair starts out as *open*; at some point in the process, it is assigned an edge or non-edge for the first time. This will be the state of the vertex pair in the final graph. Every subsequent time a vertex pair has an edge or non-edge assigned to it, that becomes its temporary state until it is closed again. A *closed edge* designates a pair of vertices that was assigned an edge the first time. So, if (i, j) is a closed edge, it means that $(i, j) \in E(G(n, p))$ of the final graph $G(n, p)$. A *closed non-edge* is defined similarly, and a *closed pair* is either a closed edge or a closed non-edge.

We begin by marking all vertex pairs as open and partitioning the vertices into four sets A_L, A_R, B_L, B_R (left and right), such that $|B_L| \sim |B_R| \sim n/\log^5 n$ and $|A_L| \sim |A_R| \sim n/2$. Additionally, we set $|A_L \cup B_L|$ and $|A_R \cup B_R|$ such that s divides them, respectively. We describe three stages which take place on a combination of an A and a B set. Stage 1 and 3 are parallel stages performed on A_L, B_L and A_R, B_R, and Stage 2 is a diagonal stage performed on A_L, B_R and A_R, B_L.

Stage 1

We assign edges to the vertex pairs of B and close them. Let G_B be the graph induced by G on the vertex set B. Because G_B has distribution $\mathcal{G}(n', p)$, with $n' = |B| \sim n/\log^5 n$, it follows from Theorems 3.3.1 and 3.6.3 that, with high probability, $\chi^*_=(G_B) \leq (2 + o(1))\chi(G_B) \leq 2n'p/(2\log n'p - 40\log\log n'p) < t$. Thus, we can divide G_B into t independent sets of roughly equal size (where the difference in size between two sets is at most 1). We label these sets $B_1, B_2, \ldots, B_t$.

Recall the 3-regular graph H that we created earlier. Let H^* be the graph on $V(H)$ and edges (i, j) if and only if i and j are distance at most 2 apart. Let $V(H) = \{h_1, h_2, \ldots, h_t\}$ be the vertex set of H, and let every vertex h_i of H correspond to B_i (one of the previously found independent sets of G_B). This is illustrated in Figure 4.5.

The graph H^* is a graph with $\Delta(H^*) \leq 9$, so $\chi(H^*) \leq 10$. Therefore, we can colour H^* equitably such that the size of each colour class is at most u (by colouring using 10 colours

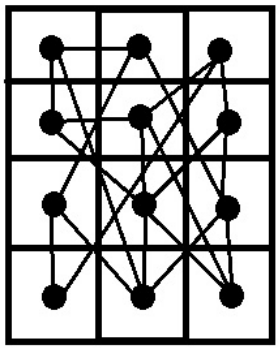


Figure 4.5: The set B is split up equitably into independent sets, and we construct a random graph H with vertices corresponding to each such set.

and then splitting classes up into classes of size $u-1$ and u). Let $H_1, H_2, \ldots, H_\kappa$ denote the sets of vertices of H in each colour class.

We assign edges to all vertex pairs with one vertex in A and one in B, and we continue with an altered version of the expose-and-merge technique described in [191], [192] and [182]. We perform the algorithms Expose and Expose-and-Merge (in that order) on A, B, H. This creates a set of colour classes from the vertices of A. Expose creates special colour classes $\sigma_1, \ldots, \sigma_t$. These special colour classes σ_i contain fewer than s vertices at first, but they will later absorb uncoloured vertices from B. This process is reminiscent of the gadgets in the proof of Theorem 4.3.1, where the inner sets absorbed one uncoloured vertex from B. The σ-sets will absorb larger sets of vertices without control over the sizes of those sets other than their maximum size. To overcome this problem we will then need to shuffle some vertices between the σ-sets until they all have exactly size s. This mechanism for shuffling is inserted during the Clean subroutine of Expose. The algorithm Expose-and-Merge colours almost all of the remaining uncoloured vertices in A with colour classes ζ_i of size s. Let

$$l = |B|\left(1 - \frac{2}{3\log^5 |B|}\right).$$

We now define two algorithms: Expose (and its subroutine Clean) and Expose-and-Merge.

Algorithm 4.1 EXPOSE

Input: $A, B_1, \ldots, B_t, H_1, \ldots, H_\kappa$

Output: $\{\sigma_1, \ldots, \sigma_t\}, \{\sigma_1^*, \ldots, \sigma_t^*\}$

partition A into κ sets of size z (and a remainder) and call them $S_1, \ldots, S_\kappa$;
for $j = 1$ to κ **do**
  assign edges to S_j;
  if S_j has u independent sets of size w **then**
    for every $i \in H_j$ **do**
      pick one of these (unlabelled) independent sets and label it σ_i;
    end for
  else
    FAIL
  end if
  Close the vertex pairs of S_j;
end for
for $i = 1$ to t **do**
  $\sigma_i, \sigma_i^*, \{\sigma_j | j > i, j \in \Gamma_H(i)\}$ = CLEAN$(\sigma_i, \{\sigma_j | j > i, j \in \Gamma_H(i)\})$
end for
for $i = 1$ to t **do**
  if $|\sigma_i| \geq s - l/t$ **then**
    colour $s - l/t$ vertices of σ_i (including all of σ_i^*) with a new colour and remove uncoloured vertices from σ_i;
  else
    FAIL
  end if
end for

Algorithm 4.2 CLEAN

Input: $\sigma_i, \{\sigma_j | j > i, j \in \Gamma_H(i)\}$

Output: $\sigma_i, \sigma_i^*, \{\sigma_j | j > i, j \in \Gamma_H(i)\}$

if $|\sigma_i - \Gamma(B_i) - \Gamma\left(\bigcup_{j|(i,j)\in E(H)} B_j\right) - \Gamma\left(\bigcup_{j|j<i,(i,j)\in E(H^*)} \sigma_j^*\right)| > k$ **then**
  remove all vertices from σ_i that lie in $\Gamma(B_i)$;
  pick $\lfloor \delta' l/t \rfloor$ vertices from $\sigma_i - \Gamma\left(\bigcup_{j|(i,j)\in E(H)} B_j\right) - \Gamma\left(\bigcup_{j|j<i,(i,j)\in E(H^*)} \sigma_j^*\right)$ and call this set σ_i^*;
  for all j s.t. $j > i$ and $(i, j) \in E(H)$, remove from σ_j all vertices that lie in $\Gamma(\sigma_i^*)$;
else
  FAIL
end if

Algorithm 4.3 EXPOSE-AND-MERGE

Input: A
Output: $\{\zeta_1, \ldots, \zeta_\tau\}$

1: $i = 1$;
2: **while** $|A - \{v \in A | v \text{ is coloured}\}| > \overline{z}$ **do**
3: pick, uniformly at random, z vertices from $A - \{v | v \text{ is coloured}\}$ and call this set S;
4: assign edges to S;
5: **if** S has u independent sets of size w **then**
6: label those independent sets that contain fewer than $1/p$ closed edges $\zeta_i, \ldots, \zeta_{u^*}$;
7: **for** $j = i$ to $i + u^*$ **do**
8: **while** $\exists$ closed edges in ζ_j **do**
9: pick a closed edge and remove one of its endpoints from ζ_j;
10: **end while**
11: colour s vertices of ζ_j with a new colour and remove uncoloured vertices from ζ_j;
12: **end for**
13: **else**
14: FAIL
15: **end if**
16: close the vertex pairs of S;
17: $i := i + u^*$
18: **end while**

Lemma 4.4.2. *With high probability,* EXPOSE *finishes and does not fail.*

Proof. EXPOSE only runs one loop, so it finishes successfully if and only if it does not fail in line 9 or 20 and CLEAN does not fail in line 6. We show first that, with high probability, S has u independent sets of size w every time, such that the algorithm EXPOSE does not fail in line 9 of EXPOSE. From Theorem 3.3.5 it follows that we fail at any loop with probability at most z^{-2}. By Lemma 2.11.2,

$$\mathbb{P}[S \text{ does not contain } u \text{ sets of size } w \text{ in some loop}] < z^{-2}\kappa = o\left(\frac{1}{n}\right). \tag{4.5}$$

With high probability, the algorithm does not fail in line 20 of EXPOSE. For a fixed i, before running CLEAN, let

$$Z = \left|\sigma_i \cap \Gamma\left(\bigcup_{j<i|(i,j)\in E(H^*)} \sigma_j^*\right)\right|.$$

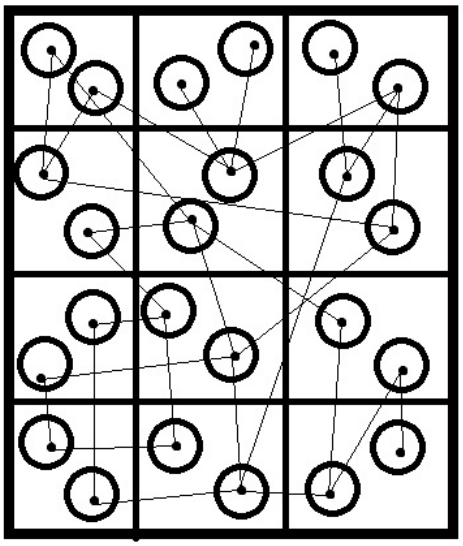


Figure 4.6: The set A is split up into sets $S_1, \ldots, S_\kappa$ of equal size, z. From these sets we pick the independent sets σ_i, in such a way that σ_i and σ_j are not picked from the same set if $(i, j) \in E(H^*)$.

We have that

$$\mathbb{E}[Z] \leq p|\sigma_i| \cdot 9 \cdot \left\lfloor \frac{\delta' l}{t} \right\rfloor \leq \frac{1}{p} = o(w)$$

and according to Lemma 2.11.5,

$$\mathbb{P}\left[Z > \frac{2}{p}\right] < O\left(e^{-\frac{1}{p}}\right) = o\left(\frac{1}{n}\right). \tag{4.6}$$

The probability that Z exceeds this bound for any i is $o\left(\frac{t}{n}\right) \to 0$.

With high probability, the algorithm does not fail in line 6 of Clean. For a fixed i, let

$$Y = \left|\sigma_i \cap \left(\Gamma\left(\bigcup_{j|(i,j)\in E(H)} B_j\right) \cup \Gamma\left(\bigcup_{j|j<i,(i,j)\in E(H^*)} \sigma_j^*\right)\right)\right|.$$

Observe that two sets σ_i and σ_j were not picked from the same S-set if $(i, j) \in E(H^*)$. Therefore, the edges between them occur independently with probability p. This is illustrated in Figure 4.6.

We have that

$$\mathbb{E}[Y] \leq p|\sigma_i| \cdot 9 \cdot \left(\frac{|B|}{t} + \frac{\delta' l}{t} \right) \leq \frac{2}{p} = o(w).$$

According to Lemma 2.11.5,

$$\mathbb{P}\left[Y > \frac{4}{p} \right] < O\left(e^{-\frac{2}{p}} \right) = o\left(\frac{1}{n} \right). \tag{4.7}$$

The probability that Y exceeds this bound for any i is $o\left(\frac{t}{n}\right) \to 0$.

□

Lemma 4.4.3. *With high probability,* Expose-and-Merge *finishes and does not fail.*

Proof. We first show that, with high probability, Expose-and-Merge finishes after at most $\frac{n}{2us} \leq \frac{1}{2}\log^{12} n \log^5 d$ loops without failing. At any iteration of the algorithm, there remain at least $\bar{z}$ uncoloured vertices. Let X be the number of edges created in line 4 of Expose-and-Merge. According to Lemma 2.11.5, $\mathbb{P}[X > pz^2] = O\left(e^{-\frac{pz^2}{2}} \right)$. Then,

$$\mathbb{P}[X > pz^2 \text{ at any step of the algorithm}] = O\left(e^{-\frac{pz^2}{2}} \frac{n}{2us} \right) = o\left(\frac{1}{n} \right). \tag{4.8}$$

Consider a set ζ_i formed at a given point in the process. There are at least $\bar{z}$ vertices left. If, during every previous loop of the algorithm, fewer than pz^2 closed edges were created, then

$$\mathbb{E}[\#\text{ closed edges in } \zeta_i] \leq pz^2 \frac{n}{2us} \frac{\binom{w}{2}}{\binom{\bar{z}}{2}} \leq \frac{1}{\log n} \frac{1}{p}.$$

Combining Lemma 2.11.1 and 2.11.5, we deduce that, with high probability, no more than a fraction $2\log^{-1} n$ of the sets contain more than $\frac{1}{p}$ closed edges.

With high probability, S has u independent sets of size w every time (such that the algorithm does not fail in line 14 of Expose-and-Merge). As we showed in Lemma 4.4.2, from Theorem 3.3.5 it follows that we fail at any loop with probability at most z^{-2}. So, by

Lemma 2.11.2,

$$\mathbb{P}[S \text{ does not contain } u \text{ sets of size } w \text{ in some loop}] < z^{-2}\frac{n}{2us} = o\left(\frac{1}{s}\right). \tag{4.9}$$

□

Stage 2

This is Stage 2, a diagonal stage. We consider the roughly $\bar{z}$ remaining uncoloured vertices in A, and name this set I. We assign edges to the vertex pairs of I and close them.

Lemma 4.4.4. *We can colour I with at most $2\bar{z}p\log d$ colours, and with each colour class having size at most $\frac{1}{p\log d}$.*

Proof. We colour the vertices of I greedily, where we label the new colours $1,\dots,r$ (we start labelling at 1 here for ease of notation: we are still using a new colour every time, rather than reusing colour 1 from the beginning of the colouring algorithm), in the following way. First, we order the vertices by repeatedly taking the vertex of minimum degree and placing it at the end of the list. We then colour the vertices in this order, assigning to each vertex $v_i^{(I)}$ the first colour that has not been used for vertices $\{v_j^{(I)} | (i,j) \in E(G), j < i\}$ and that is still "available". A colour class is "available" as long as it contains fewer than $\frac{1}{p\log d}$ vertices. We argue that this yields $r \le 2\bar{z}p\log d$.

Suppose that this is not the case. It would then follow that I spans a subgraph Λ with $\delta(\Lambda) \ge \bar{z}p\log d$ (there cannot be more than $\bar{z}p\log d$ "unavailable" colour classes since $|I| \le \bar{z}$). For a subgraph Λ of G,

$$\mathbb{E}[e(\Lambda)] < p|V(\Lambda)|^2.$$

If $\delta(\Lambda) \ge \bar{z}p\log d$, then $e(\Lambda) \ge \frac{1}{2}|V(\Lambda)|\bar{z}p\log d$. By Lemma 2.11.5, the probability that

this happens for a given random subgraph Λ with $\bar{z}p\log d < |V(\Lambda)| \leq \bar{z}$ is

$$\mathbb{P}[e(\Lambda) \geq \frac{1}{2}|V(\Lambda)|\bar{z}p\log d] = e^{-\Theta(|V(\Lambda)|\bar{z}p\log d)}.$$

So the probability of G containing a subgraph Λ with $\delta(\Lambda) \geq \bar{z}p\log d$ is bounded by

$$\sum_{i=\bar{z}p\log d}^{\bar{z}} \binom{n}{i} e^{-\Theta(i\bar{z}p\log d)} = e^{-\Omega((\bar{z}p\log d)^2)}.$$

Therefore, with high probability, no subgraphs Λ with at most $\bar{z}$ vertices exist such that $\delta(\Lambda) \geq \bar{z}p\log d$. □

Hence, we colour I with $r < 2\bar{z}p\log d$ colours and name these colour classes $I_1, \ldots, I_r$. We now assign edges to each of the vertex pairs for which one vertex is in I and one vertex is in the opposite B (so, B_L for $I \subseteq A_R$ and *vice versa*). Using Theorem 3.3.5, we find at least $\frac{|B|}{s\log^5|B|}$ independent sets of size $(1+o(1))\frac{1}{p}(2\log d - 12\log\log d)$ in B (where we ignore the partition into the sets B_j). We denote those sets by $J_1, J_2, \ldots$. For r of these independent sets, take $s - |I_k|$ vertices from $J_k - \Gamma(I_k)$ and add these to the colour class of I_k. This is permissible, because $\mathbb{E}[|J_k \cap \Gamma(I_k)|] < \frac{2}{p}\log d \cdot p \cdot (p\log d)^{-1} < \frac{2}{p}$ and, by Lemma 2.11.5,

$$\mathbb{P}\left[|J_k \cap \Gamma(I_k)| > \frac{4}{p} \text{ for any } k\right] = O\left(de^{-\frac{1}{2p}}\right) = o\left(\frac{1}{n}\right). \tag{4.10}$$

We have thereby obtained r more colour classes, each of which has size s. We continue to colour s vertices at a time with a new colour from unused sets J_k until there remain l uncoloured vertices in B. This is possible because s divides $|A \bigcup B|$ and we have coloured A and B with colour classes of exactly size s, except for the first t colour classes (which each have size $s - l/t$) in the Expose algorithm.

Stage 3

We now perform Stage 3, a parallel stage. The only vertices that remain uncoloured lie in the independent sets $B_1, \ldots, B_t$. We discard the coloured vertices from these sets. For $1 \leq i \leq t$, we add B_i to σ_i. Because of Clean, σ_i is still an independent set. Now, we will move around vertices between the colour classes σ_i, such that they all become size s. Before we added the B_i sets, the colour classes σ_i each had size $s - l/t$. The sets B_i have size l/t on average, but the size for each of them is between 0 and $|B|/t = (1+o(1))(p\log^2 d)^{-1}$. We are now going to use Theorem 4.2.2 and 4.2.3 to prove that we can redistribute some of the vertices between colour classes to ensure that all colour classes are of exactly size s. Consider the graph G_F in Figure 4.7. This graph consists of a source a_F and a sink b_F, and two vertex sets V_F and W_F. For each vertex set σ_i, there is a representing vertex $v_F^{(i)} \in V_F \subset V(G_F)$ and a representing vertex $w_F^{(i)} \in W_F \subset V(G_F)$. For $1 \leq i \leq t$, we have that $v_F^{(i)} w_F^{(i)} \in E_{G_F}$. For every (i, j) such that $(i, j) \in E_H$ we have that $v_F^{(i)} w_F^{(j)}, v_F^{(j)} w_F^{(i)} \in E_{G_F}$. The capacities of the vertices are as follows: $c_{a_F} = c_{b_F} = \delta' l$, $c_{v_F^{(i)}} = \delta' l/t + \varepsilon_i$ (which is always non-negative), and $c_{w_F^{(i)}} = \delta' l/t$. For all i, let $\varepsilon_i = |B_i| - l/t$. Now, a maximum flow of $\delta' l$ from a_F to b_F is achieved, if and only if the vertices can redistribute themselves between the sets σ_i, along the edges of H such that each of these colour classes σ_i obtains exactly size s.

Lemma 4.4.5. *A flow from a_F to b_F in G_F of size $\delta' l$ is possible.*

Proof. According to Theorem 4.2.2, if such a flow is not achievable, then there must be a cut $C_F \subset (V_F \cup W_F)$ that stops all flow from a_F to b_F, such that $\sum_{x \in C_F} c_x < l$. Because $\sum_{x \in V_F} c_x = \sum_{x \in W_F} c_x = l$, this cut cannot contain all of either V_F or W_F. By definition, there is already a matching between V_F and W_F, so $t \leq |C_F|$. Hence, $\max(|V_F \cap C_F|, |W_F \cap C_F|) \geq t/2$. Suppose that $|V_F \cap C_F| \geq t/2$. The set $W_F \cap C_F$ must cover all of the neighbourhood in W_F of $V_F \setminus C_F$.

It is possible that $|W_F \cap C_F| > |V_F \setminus C_F|$ (because the capacities can be unevenly dis-

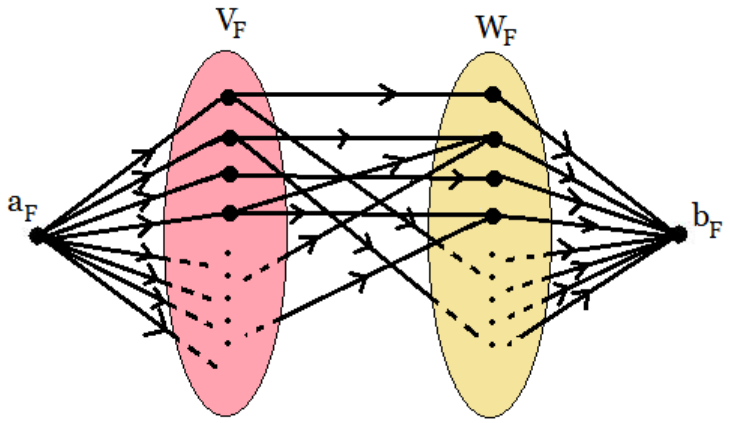


Figure 4.7: Any flow in this graph runs from the source, a_F, through a vertex $v_F^{(i)} \in V_F$, then a vertex $w_F^{(j)} \in W_F$ and then to the sink, b_F. Vertex a_F is fully connected to V_F and W_F is fully connected to b_F. There is an edge from vertex in $v_F^{(i)}$ to vertices in $w_F^{(j)}$ if $i = j$ or $(i, j) \in E_H$.

tributed). We have that

$$\sum_{v_i \in (V_F \cap C_F)} \varepsilon_i = - \sum_{v_i \in (V_F \setminus C_F)} \varepsilon_i$$

and that

$$(\delta' - 1)l/t \leq c_{v_F^{(i)}} \leq (\delta' + 1)l/t.$$

Therefore, $|W_F \cap C_F| \leq \frac{\delta'+1}{\delta'-1}|V_F \setminus C_F|$. We consider the neighbourhood of $V_F \setminus C_F$ in W_F. There is already a matching from each $v_F^{(i)}$ to a $w_F^{(i)}$. Consider the set of vertices h_i corresponding to the vertices $v_F^{(i)} \in V_F \setminus C_F$ and call this set S_H. We are interested $|\Gamma_H(S_H) \setminus S_H|$. Recall that we chose H such that

$$|\Gamma_H(S_H) \setminus S_H| \geq \delta |S_H| > (\frac{\delta' + 1}{\delta' - 1} - 1)|S_H|$$

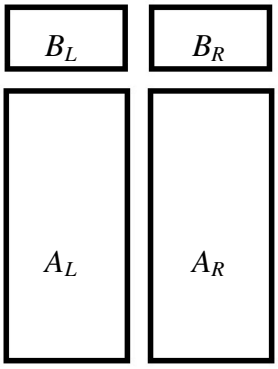
for every set $S_H \subseteq V(H)$ such that $|S_H| < |V(H)|/2$. This implies that $|\Gamma(V_F \setminus C_F) \cap W_F| > \frac{\delta'+1}{\delta'-1}|V_F \setminus C_F|$. Therefore, a cut of capacity less than $\delta' l$ does not exist.

If we suppose, instead, that $V_F \cap C_F < t/2$, then, a similar argument holds. Therefore,

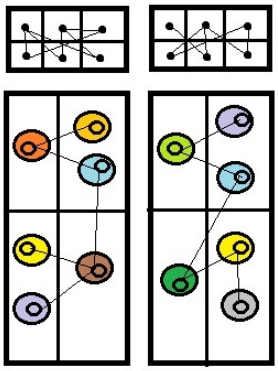
with high probability, no cut of capacity less then $\delta' l$ is possible.

□

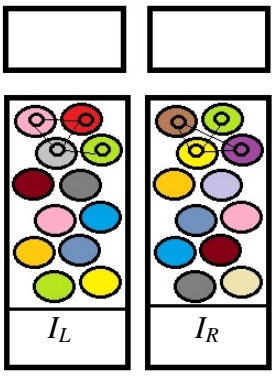
All that remains is to check that this process works for all $s/2 < s' < s$. By Lemma 2.11.2, it is sufficient to confirm that the probabilities in Equations (4.5)–(4.10) are $o\left(\frac{1}{s}\right)$, and this is indeed the case. This completes the proof of Theorem 4.4.1. □



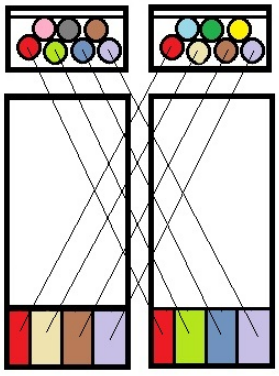
1. Split $V(G)$ into A_L, A_R, B_L, B_R.



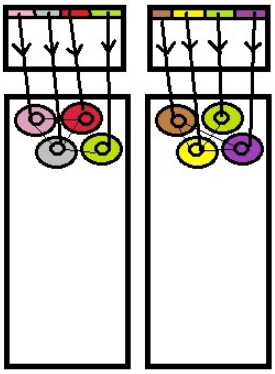
2. Create graph H, and corresponding sets in B and in σ_i sets in A.
Stage 1.



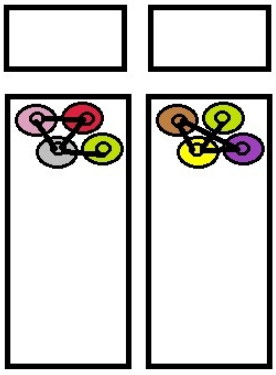
3. Colour almost all of the rest of A.
Stage 1.



4. Colour I greedily and complete its colour sets with independent sets from B.
Stage 2.



5. After having coloured almost all of B, move the remaining vertices into the corresponding σ_i sets.
Stage 3.



6. Shuffle vertices between the σ_i colour classes to correct their sizes.
Stage 3.

Figure 4.8: A cartoon of the colouring process for sparse graphs.

Chapter 5

Community Structures in Networks

This chapter is a literature review and contains no original research.

5.1 Introduction

Mesoscopic structures of networks are structures visible on a scale between the *microscopic* scale of individual nodes (such as node degree) and the *macroscopic* properties of an entire network such as mean degree, connectivity, etc. By the far the most well-studied mesoscopic property of networks is *community structure* [108, 229]. Intuitively, a community is a subset of the nodes in a network that are relatively densely connected to each other and relatively sparsely connected to the rest of the network. A community structure of a network is a partition of its nodes into such subsets.

This draws a parallel to the colouring problems described in Chapters 3 and 4. One seeks to partition a graph into subsets, based on rules about the number of edges inside and between the subsets. In the case of colouring, no edges are allowed inside subsets, and in the case of community structure, many edges are desired inside a community. Despite this similarity, these are not dual problems. In the case of proper colourings, there are no compromises. A colouring is either valid or it is not valid, and one proper k-colouring is not be better than another proper colouring. Community structure is a fuzzy problem. There may

be many solutions that give like a sensible community structure, and the preferred solution depends on the definition of the best community structure and the methods used to detect it. Even then, methods often have probabilistic elements and can give different solutions when presented with the same network multiple times. The problems are approached from completely different angles, but there certainly are links. There are fuzzy versions of colouring problems [130, 194, 205] that relax constraints or search for algorithms that accept near-optimal solutions. In Section 8.9 use a colouring approach with hard constraints to define communities in a human brain network.

One would expect many empirical networks to have some form of community structure [126], because grouping behaviour occurs in social networks (both human and animal) [116, 203, 287] and virtual social networks [266]. Additionally, biological networks such as protein-protein interaction networks [69] and brain networks [18] tend to be clustered into functional modules, and the internet includes sets of websites dealing with the same topic [107, 124]. The first analysis of community structure dates back as far as the 1920s, when Rice studied political blocs in networks created from roll call data (we study such data ourselves in Section 7.6.4) [234]. In 1950, Homans noticed that it can be useful to arrange social ties into an adjacency matrix and to infer information by rearranging the rows and columns by hand. Weiss and Jacobson conducted a sociological study of communities in organisational data in 1955 [276]. It was a paper by Girvan and Newman in 2002 that sparked the wealth of attention that this problem has received in the last decade [126, 213]. Girvan and Newman proposed a method of detecting community structure by finding edges that are most likely "between" communities and then deleting those edges to separate the communities from one another. We will discuss this method in Section 5.2.

The method by Girvan and Newman starts with a single community, which is then broken into pieces by cutting edges based on a chosen criterion. This is a *divisive algorithm*. Since then, other criteria for removing edges have been proposed to form variations of this top-down approach [110, 231]. Another option is to start with every vertex in its own

community and then merge the communities. This approach starts at a local scale and works its way up to a global scale. This is an *agglomerative algorithm* and gives another general approach. Other approaches use the eigenspectrum of the matrices related to the network, greedy heuristics, or probabilistic jumps between solutions. Many algorithms utilise combinations of these methods in order to exploit their respective advantages and circumvent disadvantages.

Ten years have passed and a myriad of journal articles have been written since Girvan and Newman proposed their method [108, 229], but there is still no consensus on the best definition of a community. A much-used quantifier for the quality of a community structure is the so-called *modularity function* Q [212, 213]. Nowadays, most algorithms are geared towards optimizing a modularity function, but a wealth of other definitions exist, and some of them have arguably stronger theoretical foundations.

In this chapter, we review some of the methods used to detect community structure in networks: both static networks and networks that change with time. The rest of this chapter is organised as follows: we first discuss four approaches that are not directly based on modularity optimization: the original method by Girvan and Newman (Section 5.2), spectral partitioning using the so-called *Laplacian matrix* (Section 5.3), methods based on random walkers (Section 5.4), and k-clique percolation (Section 5.5). In Section 5.6, we discuss the modularity function Q and a few of the many ways that have been proposed to optimise it. In Section 5.6.1 we discuss greedy heuristics, which form the basis of many fast and practical algorithms. In Section 5.6.2, we discuss simulated annealing; in Section 5.6.3, we discuss extremal optimization; in Section 5.6.4, we discuss a spectral approach to optimising modularity; and, finally, in Section 5.6.5, we discuss the Kernighan-Lin algorithm, which is often used as a refinement phase after using other algorithms. In Section 5.7, we present a generalisation of the modularity function to networks that consist of several layers (such as a network that changes over time). This generalized modularity can be optimised using the same methods that are already employed to optimise the usual

modularity function. We apply this notion to multilayer (time-dependent) functional brain networks in Chapter 8.

From now on, let $\mathbf{A}$ denote the adjacency matrix of a binary graph G, such that

$$\mathbf{A}(i,j) = \begin{cases} 1, & \text{if } ij \in E(G), \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

If G is a weighted network, then we use the matrix $\mathbf{W}$ instead. Its components are

$$\mathbf{W}(i,j) = \begin{cases} w_{ij}, & \text{if } ij \in E(G), \\ 0, & \text{otherwise,} \end{cases} \tag{5.2}$$

where w_{ij} is the weight of an edge between nodes i and j.

5.2 Removing edges

In 2002, Girvan and Newman proposed a method of detecting community structure by finding edges that are most likely to be "between" communities [126, 213]. One would expect edges that span two communities to be a bottleneck in a network, in the sense that many *shortest paths* between a vertex in one community and a vertex in a different community should run along them. A shortest path between x and y is a set of edges $(x, i_1), (i_1, i_2), (i_2, i_3), \ldots, (i_{k-1}, y)$ such that the length k of the path is minimized over all possible paths between x and y. Therefore, Ref. [126] defines the *edge betweenness* of an edge as the number of shortest paths in which that edge takes part between any pair of vertices to which the edge is not adjacent, divided by the total number of shortest paths between that pair of vertices. We will discuss the better-known notion of *node betweenness* in Section 6.3.2. This notion is defined in a very similar way but on nodes rather than edges.

The algorithm proposed in [126] calculates the betweenness of all edges and removes

the edge of highest betweenness. It then recalculates the betweenness of all edges (as the removal of one edge changes this value for others) and repeats. The communities are defined as the separate network components at any step of the algorithm.[1] Assuming that the graph is connected initially, this number runs from 1 to n (when all the edges have been removed). The choice of when to stop the algorithm and choose the current components as the communities can depend on a quality function (see Section 5.6) or an initially chosen number or size of communities. The entire process can be represented as a *dendrogram*: a tree diagram showing the hierarchy of clusterings. This algorithm runs in $O(m^2n)$ time, because calculating the betweenness for all edges takes $O(mn)$ time, and this needs to be repeated for every edge that is removed. We discuss the time for computing betweenness in Section 6.3.5.

In [213], Girvan and Newman discussed other variations of this method by choosing other notions of betweenness. Such notions include *random-walk betweenness*, which is given by a probability that a random walk between any two vertices passes through an edge, and the *current-flow betweenness*, which is given by the absolute value of the fraction of the total current along the edge summed over all pairs of vertices acting as a source or a sink, if a unit resistance is placed on every edge [213].

Similar methods, which detect communities by removing edges that are likely to span them, were proposed by Radicchi *et al* [231] and Fortunato *et al* [110].

The method by Radicchi *et al* chooses edges to remove based on the number of small circuits in which they take part (edges on the fewest small circuits are removed first). This method is fast because it takes into account only local properties of an edge. They defined an *edge clustering coefficient* as

$$c^{(3)}[(i,j)] = \frac{z^{(3)}[(i,j)]}{\min[(d(i)-1, d(j)-1]}.$$

[1] If a network is disconnected, most community detection methods are applied to each component separately.

This edge clustering coefficient is the ratio between the number of triangles in which an edge takes part, divided by the number of triangles in which it potentially take part given the degrees of the vertices to which it is adjacent. To avoid a clustering coefficient equal to 0, Radicchi *et al* used a modified clustering coefficient $\widehat{c}^{(3)}[(i,j)]$ by setting the numerator to $z^{(3)}[(i,j)] + 1$. This also ensures that nodes with degree 1 are not split into isolated communities. Defining an analogous quantity $\widehat{c}^{(g)}[(i,j)]$ for general cycles of length g, one can interpolate continuously between a local and global algorithm. For cycles of constant size, their algorithm runs in $O(m^2)$ time.

Fortunato *et al* proposed to use a so-called *information centrality* [179] of edges to decide the order in which to delete them. This notion takes the reciprocal of the distance between all nodes in a network into account and examines how this quantity changes with the removal of an edge or node. The concept is described in more detail in Section 6.3.4. This is a global quantity of the network that needs to be recomputed many times throughout the algorithm, so the running time is long: $O(m^3 n)$.

5.3 Spectral partitioning

Suppose that we want to find a partition V_1, V_2 with $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \varnothing$, which minimizes the $\text{cut}(V_1, V_2)$, where $\text{cut}(V_1, V_2) = \sum_{i \in V_1, j \in V_2} \mathbf{W}(i,j)$. Additionally, we would like to have some control over the size of V_1 and V_2. For example, we are not interested in the trivial solution where $V_1 = V$ and $V_2 = \varnothing$ (or *vice versa*).

A popular and well-studied method to solve this problem is *spectral bipartitioning*. A *Laplacian matrix* $\mathbf{L}$ that is sometimes called the admittance matrix or Kirchhoff matrix of a graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where the degree matrix $\mathbf{D}$

$$\mathbf{D}(i,j) = \begin{cases} d(i), & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases}$$

and $\mathbf{W}$ is the (weighted) adjacency matrix of the graph. In other words,

$$\mathbf{L}(i,j) = \begin{cases} \sum_{k \neq i} \mathbf{W}(i,k), & \text{if } i = j \\ -\mathbf{W}(i,j), & \text{otherwise.} \end{cases}$$

For many applications, the Laplacian matrix is a more fundamental matrix of the graph than the adjacency matrix because of the relation of its eigenvectors to numerous graph invariant–including connectivity, expanding properties, maximum and minimum cut, and independence number [198]. The sum of every row and column vector of the Laplacian matrix is 0, so the vector $\mathbf{v}_0 = (1, 1, 1, ...)$ is always an eigenvector of the Laplacian matrix with eigenvalue $\lambda_0 = 0$. Additionally, the Laplacian is positive semi-definite. Hence, λ_0 is in fact the smallest eigenvalue. The eigenvector associated with the second smallest eigenvalue λ_1 is called the *Fiedler eigenvector*. This eigenvector has positive and negative entries that partition a graph into components that are "weakly" linked to each other. There is no guarantee that this method yields an optimal cut, but it does well in some practical applications, especially when λ_1 is well-separated from the eigenvalues above it [136, 209, 258].

To divide a network into smaller components, one can apply this same method recursively on its partitions until the desired size criteria have been obtained. Such a method is an example of *recursive spectral bisection* [16]. Alternatively, one can use third-smallest, fourth-smallest, etc. eigenvalues (associated with the Laplacian) and their corresponding eigenvectors to partition a network more finely [235].

5.4 Random walkers

The intuitive idea behind using random walkers when detecting communities is that a random walker is likely to get "stuck" inside a community: once a random walker is in a community, it is likely to remain there in the next step, because a vertex in a community is

better connected to other vertices in that community than to vertices outside of that community. Therefore, different community-detection methods based on random walkers have been proposed. For example, Pons and Latapy proposed an agglomerative method based on the likelihood of a random walker moving between communities [226].

One can also use random walks to define a variation of a betweenness function on edges, as was done by Newman in 2005 [208]. This can then be combined with the top-down edge removal approach from Section 5.2.

Ref. [223] introduced the notions of an *α-community* and an *α-partition*. An α-community is a set of nodes such that if a random walker is currently on a node in the set, then it will remain there at the next step with a probability that is not less than some specified constant $\alpha \in (0, 1)$. For each potential community, one can also seek the maximum value of α to try to quantify the significance of that community. If a graph can be partitioned completely into disjoint α-communities, then this is called an "α-partition" of the graph. In Section 7.3.5, we discuss a very similar method used to detect network *cores* rather than communities.

5.5 *k*-Clique percolation

The *k-clique percolation method* is inspired by the idea that communities are likely to contain many cliques (*i.e.*, fully-connected subgraphs; see also Section 3.2) [218]. A *k-clique community* is a set of k-cliques in the network that form a *connected* set. Two k-cliques are connected if they share $k - 1$ of their vertices. As k is increased, the definition becomes more stringent. This definition allows partitions of a network into communities that do not cover all of the nodes, and it also allows for communities to overlap. We will not discuss overlapping communities in detail, although much attention has been given to them [2, 12, 83, 102]. We will revisit overlapping communities in Section 7.3.3 in the context of core-periphery structures in networks.

5.6 Modularity optimization

When looking for community structure in a network, the sizes of the communities are not in general known beforehand. Choosing arbitrary sizes will most likely prevent one from finding the best solutions. Therefore, we need a method that finds a "natural" division of a network into communities that does not use any prior knowledge other than the weighted adjacency matrix itself. The intuition of a community structure is that it is a division of the network into modules of nodes, such that the nodes are more densely connected within their own modules than to the rest of the network. Hence, instead of considering the absolute cut size, we need to consider the density of edges across cuts as compared to the aggregate edge density of a network. In other words, we are comparing the density of edges across cuts to the expected density of such edges if we were to pick our cuts at random. We then try to minimize this value. Alternatively, we can maximize the density of edges inside communities compared to the expected such density to obtain the same result. This notion leads to the definition of *modularity*, which is defined for a graph partition and is the total edge weight (and hence the number of edges for unweighted networks) within communities minus the expected total edge weight within communities. Formally,

$$Q = \frac{1}{2m} \sum_{ij} \left(\mathbf{W}(i,j) - \mathbf{P}(i,j) \right) \delta_{c(i),c(j)}, \tag{5.3}$$

where m is the total edge weight, $\mathbf{W}$ is the (weighted) adjacency matrix, $\mathbf{P}$ is a *null model matrix*, $c(i)$ is the community to which a node i is assigned, and $\delta_{i,j}$ is the Kronecker delta function:

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

The factor $\frac{1}{2m}$ does not play a role in the optimization; it is there to normalize modularity in a way that makes it more sensible to compare the optimal modularity of different networks

(with different edge densities) to one another. To simplify the expression, we call

$$\mathbf{B}(i,j) = \mathbf{A}(i,j) - \mathbf{P}(i,j) \tag{5.5}$$

the *modularity matrix*. The most popular choice of null model was proposed by Newman and Girvan [212, 213]. It is defined by

$$\mathbf{P}(i,j) = \frac{d(i)d(j)}{2m}.$$

This is closely related to the *configuration model* that was proposed by Bollobás [34] and was studied further by, for example, Molloy and Reed [200] and Chung and Lu [73].

Modularity is closely related to the Potts model [230, 285], which describes interacting particles that are placed on a network (usually a lattice). Each particle has a spin. In the simpler Ising model, the spins can only take one out of two values: up and down. In the Potts model, the spins can take $q \geq 2$ values. One can visualise this as each spin pointing in one of q equally-spaced directions that are specified by angles. Spins can interact with each other *ferromagnetically* (neighbouring spins aim to align with each other) or *antiferromagnetically* (neighbouring particles seek to have opposite spins). When neighbouring nodes interact ferromagnetically, non-neighbouring nodes interact antiferromagnetically, and the spins are seen as the community to which a node belongs, then minimizing the energy of the particle model becomes the same as maximizing the modularity as described in Equation (5.3) [108, 229, 232, 233]. The Potts Hamiltonian is the sum of the interaction energies:

$$H = -\sum_{ij} \mathbf{J}(i,j)\delta(\sigma(i),\sigma(j))$$

where $\sigma(i)$ denotes the spin of particle i. Thus, the energy of the interaction between two nodes is only considered for pairs of nodes that are in the same community. For nodes that share an edge with a higher weight than expected (given a null model), $\mathbf{J}(i,j) > 0$. For

nodes that share an edge with a lower weight than expected, $\mathbf{J}(i,j) < 0$. Thus, $\mathbf{J}(i,j) = \mathbf{B}(i,j)/(2m)$ gives $H = -Q$.

In terms of complexity, optimizing modularity is equivalent to solving the NP-hard MAX-CUT problem described in Section 5.3 [54], so there is almost certainly no polynomial-time algorithm that will find the optimal modularity for all networks. Furthermore, it has been shown that, in practice, the modularity function Q often has many nearly-optimal local maxima and does not have a clear global maximum [131, 177]. Another issue is the so-called resolution limit, which was first pointed out by Fortunato and Barthélemy [109]. Modularity optimisation misses communities that are smaller than a certain threshold with respect to the size of a network. This can be circumvented by incorporating a *resolution parameter* into the modularity function that allows one to zoom in and out and detect communities of different sizes. Such a feature is part of many of the methods that we will discuss in the next few sections.

5.6.1 Greedy algorithms

In [207], Newman presented a straightforward greedy algorithm aimed to find communities by optimising modularity. The algorithm starts by putting each vertex into its own community. At each step, it then joins two communities, based on creating the largest increase – or smallest decrease – in the modularity function Q. This continues until all vertices are part of the same community. One then chooses a nontrivial partition from one of the time steps, based on the maximal value for Q that was achieved during the process.

By keeping track of edges cleverly, [207] showed that this algorithm can run in $O((n+m)n)$ time. Unsurprisingly, it is also much better at finding a partition with optimal modularity than the methods described in Section 5.2. Later, in [76], the algorithm was sped up to run in $O(md\log n)$, where d is the depth of the dendrogram describing the process of merging communities. Thus, d is related to the number of time steps of the algorithm. The same paper also makes the case that many real networks possess an inherent hierarchical

structure, so that d is small and scales with $\log n$ (the smallest it can be).

Currently, perhaps the most popular method of modularity optimization is the Louvain method introduced by Blondel *et al* in 2008 [30]. It is similar to the method by Newman described above, but it only changes the community assignment of one vertex at a time. The algorithm starts by assigning each node to a separate community. At every time step, it chooses a node i and considers whether moving i to the community of any of its neighbours $j \in \Gamma(i)$ increases the modularity Q. If an increase is possible, i is reassigned to the community of the neighbour j that maximizes this increase. This process is applied repeatedly and sequentially for all nodes i until no further increase of the modularity is possible. The change in aggregate modularity when the community assignment of one node changes is a local quantity that only depends on the neighbours of i and their community assignments, and it is therefore cheap to compute (see also Section 5.6.3). Blondel *et al* provided empirical evidence that the outcome of this phase of the algorithm is reasonably robust to the order in which the nodes are considered, although the running time is affected by that. In the next phase of the algorithm, a new network is created by merging the communities that were formed in the first phase into single nodes. All of the weights of edges between two communities are added to form the weight of the edge between the two new nodes formed by the two communities. The weight of the edges within a community are added to form the weight of a self-loop on the new node created by that community. The greedy algorithm is then repeated to merge some communities even further until no further increase in Q can be achieved. This process is repeated until it stops leading to further changes. This method is widely used because of its perceived good speed and performance. However, such qualities have been backed up almost exclusively by empirical evidence [30, 196].

5.6.2 Simulated annealing

Simulated annealing is a probabilistic method for finding a global minimum of a cost function that can have many local minima [25]. We will use simulated annealing to help find

core-periphery structures in Chapter 7. It was introduced independently by Kirkpatrick, Gelatt, and Vecchi in 1983 [163] and Černỳ in 1985 [66]. The method is inspired by the way a metal sets into a crystalline structure when it is cooled down. The slower the cooling process is, the more homogeneous and lower-energy the final state will be (especially with ferrous metals). This so-called *annealing process* is widely use in metallurgy to refine the structure of metals and improve their ductility. When algorithmically simulated, the process has been extremely useful for solving a wide range of problems.

Simulated-annealing is a local search algorithm that moves from one candidate solution to a neighbouring candidate solution with some probability. This probability depends on both the current temperature and the difference between the two candidates in the function value of the function that is being optimized. Even if a neighbouring candidate has a worse function value, a simulated-annealing algorithm algorithm still moves there with some positive probability. This is similar to the so-called *Metropolis algorithm*, which does not depend on a temperature [10]. A basic simulated-annealing algorithm is defined as follows. Let $V(G) = \{x_1, \ldots, x_n\}$ be the set of all solutions, with G defining binary relationships between them. We present this algorithm as a simplified version of [270], which we use in Chapter 7. This algorithm aims to find an x_i that minimizes $f(x_i)$, and there is a given *cooling schedule* $T(t) \to 0$ at $t \to \infty$.

Algorithm 5.1 SIMULATED ANNEALING

1: $t := 0$
2: $T(0) := T_0$
3: choose $x_t \in V$ uniformly at random.
4: **while** $T(t) > T_{\min}$ && $t < t_{\max}$ **do**
5: choose $x_i \in \Gamma(x_t)$ uniformly at random;
6: set $x_{t+1} := x_i$ with probability $\min\{1, e^{-\frac{f(x_i)-f(x_t)}{T(t)}}\}$;
7: **else**, set $x_{t+1} = x_t$;
8: $t := t + 1$.
9: **end while**

One can also use other stopping criteria can be used, such a maximum number of tries or a maximum number of consecutive rejections within one temperature rather than on

some maximum number of tries [270].

In 1988, Sasaki and Hajek showed that, even though the problem of finding a maximal matching (a maximal set of mutually non-adjacent edges) in a graph is possible in polynomial time, simulated annealing fails to do so on average for all graphs [242].

Another example of a combinatorial optimization problem is graph bisection. A bisection is a partition of the vertex set $V(G)$ of a graph G into V_1 and V_2, such that $V(G) = V_1 \cup V_2$ and $|V_1| = |V_2| = \frac{|V|}{2}$. The bisection problem is about finding a cut such that the number of edges spanning the cut is minimized. Jerrum and Sorkin considered the problem of finding an optimal bisection in a random graph with distribution $\mathcal{G}(n, p, r)$, which is similar to $\mathcal{G}(n, p)$ but has a "planted" optimal bisection in it (see Section 2.5) [151]. In this model, $V(G) = V_1 \cup V_2$ is given, and edges appear with probability p within V_1 or V_2 and with probability $r < p$ across V_1 and V_2, such that the bisection V_1, V_2 is optimal with high probability. They showed that in this model, if $p - r = \Theta(\frac{1}{n^\alpha})$, with $0 \leq \alpha \leq \frac{1}{6}$, then simulated annealing finds the optimal bisection in time $O(n^2)$ with probability exponentially close to 1 [151]. We note for both the problem of finding a maximal matching and finding a minimal bisection that a cooling schedule is unnecessary. In other words, the Metropolis algorithm performs as well as simulated annealing, although it has been proven that this is not always the case [256, 274, 275].

We will use simulated annealing ourselves as part of our method for finding core-periphery structure (see Chapter 7). We give our motivation for using this method in Section 7.8.

5.6.3 Extremal optimization

An *extremal optimization* [13] approach to optimising modularity is described in [89]. Roughly, extremal optimization (EO) is a technique that searches for an optimum by identifying the worst performing components of a current solution and then swapping them. The approach described in [89] starts with an arbitrary bipartition of the vertices in a network,

and it defines a contribution to the aggregate modularity of a single node i as

$$q(i) = \frac{1}{d(i)} \left(\sum_{j \in \Gamma(i)} \delta_{c(i),c(j)} - d(i) a_{c(i)} \right), \tag{5.6}$$

where $a_{c(i)}$ is the fraction of edges that have at least one node in community $c(i)$. This is a way of writing the contribution of node i to the aggregate modularity in a way that is scaled by its degree and normalized between -1 and 1. One can run the algorithm as follows. The graph is split in half uniformly at random. The node with the lowest contribution to the modularity is swapped to the other side until this action no longer improves the modularity. This process is then repeated on both parts of the partition iteratively until the modularity can no longer be improved by splitting further. This approach depends heavily on the initial configuration, as it has no way of escaping a local maximum. To get around this drawback, [89] proposed to instead rank the nodes according to their individual contributions to the modularity and then to choose the node to swap with probability $\mathbb{P}(i) \propto r(i)^{\tau}$, where $r(i)$ is the rank of the node (in reverse order of the contributions $q(i)$). This is called τ-EO, and it was introduced in [32]. The total algorithm can be run in $O(n^2 \log n)$ time.

5.6.4 Newman's spectral method

This spectral method to optimize modularity and detect community structure in networks was developed by Newman in [209, 210]. It is similar to the method described in Section 5.3. One calculates the eigenvectors of the modularity matrix $\mathbf{B}$ (Equation 5.5) instead of the weighted adjacency matrix $\mathbf{W}$.

5.6.5 The Kernighan-Lin algorithm

Joint strategies are often employed for community-detection algorithms. Spectral approaches give a good idea of the community structure, and algorithms such as the Kernighan-Lin algorithm provide refinement [159]. The Kernighan-Lin algorithm starts with an initial

division of nodes into communities, and it improves the division by either swapping set numbers of nodes between communities (if the community sizes are set) or it moves individual nodes between communities one at a time.

To avoid getting stuck in local optima, moves that decrease Q by a small amount are also permitted. After a certain number of moves a decision is made on whether to accept the total change. The outcome depends highly on the initial partition, which is why the method is best used as a refinement stage after a good partition has been constructed using some other method.

5.7 Multilayer community structure

We describe a method proposed in [204] that was designed to study networks that consist of multiple layers. For example, these layers can describe a time-dependent network at different time steps or they can describe different networks created by different types of edges between a set of nodes. This method is a generalisation of optimizing modularity. Let N be a multilayer network. Each network layer N_s is represented by an adjacency matrix $\mathbf{W}_s$, where entries $\mathbf{W}_s(i,j)$ represent the edge between node i and j in layer s. Nodes are also connected to themselves between layers by interlayer couplings $t_{rs}(i)$ that represent the edge between node i in layer r and itself (node i) in layer s. The total strength of a node i within a layer s is $k_s(i) = \sum_j \mathbf{W}_s(i,j)$, and the total strength across layers is $g_s(i) = \sum_r t_{rs}(i)$. We then define the multilayer strength by $\kappa_s(i) = k_s(i) + g_s(i)$. We let $\mu_N = \frac{1}{2}\sum_{i,s} \kappa_s(i)$ be the total strength in the network and let $w_s = \sum_i k_s(i)$ be the total strength in layer s. This gives a multilayer generalisation of modularity:

$$Q = \frac{1}{2\mu_N} \sum_{ijrs} \left\{ \left(\mathbf{W}_s(i,j) - \gamma_s \frac{k_s(i)k_s(j)}{2w_s} \right) \delta_{r,s} + \delta_{i,j} t_{rs}(j) \right\} \delta(c_s(i), c_r(j)) \,, \tag{5.7}$$

where $c_s(i)$ is the community assignment of node i in layer s and γ_s is the *resolution parameter* of layer s. A resolution parameter allows one to tune the size scale of communities.

This method aims to find communities that maximize multilayer modularity and which can span different layers of a network. In Chapter 8, we apply this method to time-dependent fMRI networks of brains in humans who are learning a simple motor skill.

Chapter 6

Centrality Measures in Networks

This chapter consists of both a literature review and original work. The original work is in Section 6.6 and appears in a working paper by MPR and M. A. Porter [237].

6.1 Introduction

This chapter is a relatively short one. Its main purpose is to serve as an introduction to Chapter 7, where many of the concepts introduced and defined here will be used in the slightly different context of *core-periphery structure*.

In network theory, the term *centrality* in and of itself has little meaning. It is most often used as a synonym for *importance*, and it can be defined and computed in many ways, depending on the network structure and the context in which importance needs to be measured. Usually, centralities are defined as functions of the nodes of the network, but there are also different ways to define centralities on edges or other subgraphs. Arguably, measures of core-periphery structure are a centrality measure on sets of nodes rather than single nodes. Here we will discuss centrality measures based on degree, distances, shortest paths, eigenvectors, and core-periphery structure.

Although Cauchy [65] was the first to introduce eigenvectors in 1823, which are certainly a centrality measure (see Section 6.4), this was not initially done in the context of

graphs. The first mention of centrality in a graph can be found in a 1869 paper by Jordan [152], which presents two types of graph centres, called *"centre de premier/seconde espèce"* (centre of the first/second kind). These two measures will serve as a warm-up to the subject of centrality, both because they were chronologically first and because of their simple and intuitive definitions. They will not be used in the rest of the chapter, as they have the practical limitation of only being defined on trees[1].

(1) The first is a centre based on the size (by counting edges) of the graph segment that a vertex can "see" in each direction.

Lemma 6.1.1. *[152] Let T be a tree, x be a vertex in $V(T)$, and $\{y_1, y_2, \ldots\} = \Gamma(x)$ be the neighbours of x. Let $p_{x,i} = |\{z_1 z_2 \in E(T) \text{ such that } y_i \in SP(z_1, x)\}|$, for $1 \leq i \leq |\Gamma(x)|$ and let $q_x = \max_i p_{x,i}$. Then T contains either one vertex v, such that $q_v \leq \frac{m}{2}$, or two vertices v_1, v_2, such that $q_{z_1} = q_{z_2} = \frac{m+1}{2}$. For all other vertices x, the function $q_x > \frac{m+1}{2}$.*

Here, $SP(x, y)$ denotes the (set of) shortest path(s) from x to y, which we define now. A path in G between two vertices x and y is an ordered set of vertices $x = x_1, x_2, \ldots, x_l = y$, such that $x_i x_{i+1} \in E(G)$ for $1 \leq i \leq l - 1$, and a *shortest path* is one that minimizes l over all such sets. Note that on trees (but not in general; see Section 6.3.2), shortest paths are unique. In essence, Jordan noticed that for any tree, there will either be a vertex or an edge that is central in the tree, in the sense that it does not have "most of the tree" on one side of it.

(2) Let a *leaf* of a tree be a vertex i such that $d(i) = 1$. Note that every tree has at least two leaves. The second centre is obtained by pruning (removing) all of the leaves and their incident edges, from T simultaneously, and repeating this step until the remaining, pruned tree consists of just one edge or just one vertex. The centre is then defined as the one remaining edge, or the one remaining vertex.

Figure 6.1 contains a simple example to show that these are different measures. Al-

[1] Jordan did make an attempt in [152] to extend these measures to graphs of higher connectivity, but I do not think they are well-defined.

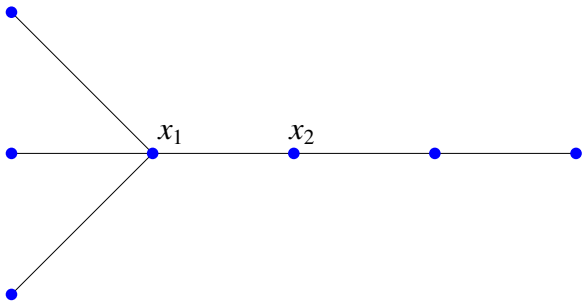


Figure 6.1: The vertex x_1 is a *centre de premier espèce* and x_2 is a *centre de seconde espèce*

though Jordan only proposed these methods as ways of finding a single centre, one can easily turn them into a continuous centrality measure. For example, one can define a centrality measure of the "premier espèce" as $c^{(1)}(x) = 1 - q_x/m$ and a centrality measure of the "seconde espèce" as $c^{(2)}(x) = s(x)/s(T)$, where $s(T)$ is 1 plus the total number of pruning steps until T is pruned, x is removed at the $s(x)$th step in the above pruning process, or $s(x) = s(T)$ if x is the central vertex of adjacent to the central edge.

The rest of this chapter is organised as follows. We introduce and define a set of existing centrality measures: degree centrality, closeness centrality, betweenness centrality, eigenvector centrality and the closely related PageRank centrality, and subgraph centrality. Subgraph centrality was introduced in 2005 by Estrada [100] –although it is a special case of *power centrality* [49] (see Section 6.5)– together with a conjecture about its ability to *distinguish* nodes from each other where other centrality measures fail to do so. In this case, we say that a measure can distinguish two vertices when it assigns different centrality scores to them. We partially solve Estrada's conjecture and discuss the larger problem of finding graphs in which certain centrality measures fail to distinguish any of their vertices.

6.2 Degree Centrality

Degree centrality is one of the simplest properties of individual vertices in a graph, so the notion itself is as old as graph theory. It is also the most basic type of centrality and is based only on the immediate neighbourhood of a node. It is defined as

$$d_i = \sum_{j=1}^{n} \mathbf{A}_{ij}.$$

Given that the degree is just the sum of one of the rows of the adjacency matrix, it can clearly be computed in $O(n)$ time. In binary graphs, this is interpreted as the number of neighbours of a node. In weighted graphs, sometimes the total weight of the edges incident to i rather than the number of edges coming out of node i is measured. This is called the *strength* of node i.

6.3 Centrality measures based on shortest paths

The concept of centrality intuitively relates to distances, so it makes sense to define centralities in terms of distances in a graph G (*e.g.* in terms of the lengths of paths). We present the two most popular centrality measures based on shortest paths: closeness centrality and betweenness centrality. Closeness centrality is defined using the length of shortest paths, and betweenness centrality is defined by counting shortest paths. In both cases, one needs to find the shortest paths between all pairs of vertices, but algorithms exist to do this in $O(nm)$ time (see Section 6.3.5).

Let $d(i, j)$ denote the (integer) distance between vertex i and j (the length of a shortest path between them), and let $d(G)$ denote the diameter of a graph G – *i.e.*, the greatest distance between a vertex pair of G taken over all pairs. We will use this notation throughout this chapter and these notions will come back in Section 6.6.

6.3.1 Closeness Centrality

The *closeness centrality* of a node was introduced by Sabidussi in 1966 [241] and is defined as the mean shortest path from a node to each of the other nodes in a network. This notion of centrality comes closest to the idea of a node taking a central position in space. It is given by

$$cc_j = \frac{1}{n} \sum_{i \in V \setminus j} d(i, j)\,,$$

where $d(i, j)$ is the length of a shortest path between i and j (or the sum of edge weights in a shortest path in the context of weighted networks). It should be noted that the sum of all distances from a node to other nodes in the network, $\sum_{i \in V \setminus j} d(i, j)$, had already been introduced as a measure by Harary in 1959 [142], where it was called the *distance sum* or *status of a node* j. Closeness centrality is sometimes called *median centrality* [57], where the vertex with minimal closeness centrality is called the *median* or the *distance center* of a graph G [139]. The vertex with the minimal closeness centrality over all vertices is also known in operations research as the "center of the graph", because in OR it is often important to locate the node that minimizes travel or response time to all other nodes [249]. Occasionally the term *closeness centrality* is used for other measures as well [211].

6.3.2 Betweenness Centrality

Betweenness centrality was introduced independently by Anthonisse in 1971 [8] and Freeman in 1977 [114], and it is intended to measure the extent to which a node controls information flowing through a network [211], by counting how many shortest paths (between pairs of other nodes in the network) on which the node lies. It is defined as

$$bc(i) = \sum_{j,k \in V \setminus i} \frac{P^*_{jk}(i)}{P^*_{jk}},$$

where P^*_{jk} is the number of different shortest paths running from node j to node k, and $P^*_{jk}(i)$ is the number of such paths that contain node i.

6.3.3 Stress centrality

Stress centrality was briefly introduced by Shimbel in 1953 [246] and is a precursor to betweenness centrality. It is defined as the total number of shortest paths on which a node lies, without normalizing it by the total number of shortest paths between the pairs of other nodes:

$$st(i) = \sum_{j,k \in V \setminus i} P^*_{jk}(i). \tag{6.1}$$

6.3.4 Delta centralities

Delta centralities form a family of centrality measures introduced by Latora and Marchiori in 2007 [179]. It measures how the "efficiency" of a network changes when a node i is removed. It is defined as

$$c^{\Delta}(i) = \frac{P[G] - P[G \setminus i]}{P[G]},$$

where $P[G]$ is some function that in some way quantifies the cohesiveness or efficiency of the graph. This is a very general measure. For example, if we let $P[G] = m$, then $c^{\Delta}(i)$ is simply the degree centrality of i (scaled by a constant). Latora and Marchiori proposed to use

$$P[G] = E[G] = \frac{1}{n(n-1)} \sum_{i \neq j \in V(G)} \frac{1}{d(i,j)},$$

where $d(i,j)$ is the distance between nodes i and j. This quantity is called the *efficiency* of a graph and it measures how well information can be exchanged between pairs of nodes [178]. This then defines the *information centrality* [179]:

$$ic(i) = \frac{E[G] - E[G \setminus i]}{E[G]}.$$

See also Section 5.2.

6.3.5 Algorithms

To compute the betweenness centrality of all vertices of a network, one needs to know the location of all shortest paths between all pairs of nodes. This is harder than simply knowing the length of a shortest path between all pairs of nodes, so betweenness centrality is more expensive to compute than closeness centrality.

Let $i, j, k \in V(G)$. We note that i lies on a shortest path between j and k in G if and only if $d(j,k) = d(i,j) + d(i,k)$. We also note that

$$P^*_{jk}(i) = \begin{cases} 0, & \text{if } d(j,k) < d(i,j) + d(i,k), \\ P^*_{ij} \times P^*_{ik}, & \text{otherwise.} \end{cases} \tag{6.2}$$

Most algorithms to compute betweenness centrality use the so-called *pair dependency*, defined as $\delta_{jk}(i) = P^*_{jk}(i)/P^*_{jk}$ [53]. The betweenness centrality of i is then the sum of pair dependencies:

$$bc(i) = \sum_{j \neq k \in V \setminus i} \delta_{jk}(i).$$

Determining betweenness centralities now requires two steps: (i) computing the length and the number of shortest paths between all pairs of vertices, and (ii) summing all pair dependencies. A fast algorithm by Brandes [53] finds all shortest paths between all pairs by using a breadth-first-search (for unweighted graphs) or Dijkstra's algorithm (for weighted graphs) approach, in $O(m)$ and $O(m+n\log n)$ running time, respectively. Summing over all pair-dependencies naively would take $O(n^3)$ time, but luckily this is not necessary because of their interdependence. Brandes' algorithm uses recursive relationships between them to find all the sums in $O(nm)$ time, so the entire process runs in $O(nm)$ time as well.

6.4 Eigenvector Centralities

6.4.1 Basic Eigenvector Centrality

The notion of eigenvectors has been around since Cauchy introduced them in 1823, to aid in solving sets of linear equations [65], but they also serve well as a centrality measure in networks. The standard *eigenvector centrality* [48, 75] of a node i is the value of the ith entry in the leading eigenvector of the adjacency matrix (the one corresponding to the largest eigenvalue). In other words, the eigenvector centrality of node i, denoted $ec(i)$, satisfies

$$ec(i) = \frac{1}{\lambda_1} \sum_{j \in \Gamma(i)} ec_j.$$

If we are thinking about centralities in terms of walks in the network (which we will do in Section 6.6), then one can think of the eigenvector centrality of node i as the sum over k of the number of distinct walks of length k starting at node i, divided by the total number of walks of length k in the network. Computing the complete set of eigenvectors and eigenvalues of a matrix within exponentially small error bounds takes $O(n^3)$ time [219], but, as we shall see in Section 6.4.2, fast algorithms exist with good approximation ratios.

6.4.2 PageRank

The PageRank algorithm was introduced by Page and Brin in 1999 and named after Page [217]. The PageRank centrality $pr(i)$ of node i is the ith entry of the leading eigenvector of a *modified adjacency matrix*. In this modified adjacency matrix, the entries are normalized such that $\sum_{j \neq i} M_{ij} = 1$ for all i, and each $M_{ij} = A_{ij}/d_i$ is the probability of a random walker at node i transitioning to node j in the next time step. The PageRank vector is the stationary probability distribution of a random walker on a network. PageRank and similar measures can be calculated quickly (in $O(n+m)$ running time) using an iterative approach. Because $pr(i) = \sum_j pr(j) M_{ij}$, one can repeatedly update all of the PageRank values until they sta-

bilize [26]. This is particularly useful when a good approximation to the vector is already known. This is the case with Google's PageRank algorithm[2], which updates regularly but does not need to compute the entire PageRank vector for the internet from scratch every day. One may wish to add damping factors and "teleportation" to better imitate a random walker that does not "get stuck" in certain parts of the graphs - for example, if the graph is disconnected or if a node has only incoming edges.

6.5 Katz centrality and communicability

Many problems in the study of networks focus on flows along the edges of a network [211]. Although flows such as a random walk are conservative, the flow of information tends to diminish as it moves through a network. The influence of one vertex on another decreases with increasing network distance between them [123].

Such considerations have sparked an interest in network *communicability*, in which a perturbation of one vertex is felt by the other vertices with differing intensities [98]. These intensities depend on all paths between a pair of vertices, though longer paths have smaller influence. This idea is expressed in a general communicability function

$$cm(i) = \sum_{k=0}^{\infty} \sum_{j=1}^{n} r(k)(\mathbf{A}^k)_{ij}. \tag{6.3}$$

A special version of communicability is the Katz centrality [157], where $r(k) = r_1^k$, where $0 < r_1 < 1$ is some constant, (also see Bonacich power centrality [49]). See [99] for an extensive review of network communicability.

Note that one can compute the *self-communicability* (*i.e.*, *karma centrality*) of vertices:

$$kc(i) = \sum_{k=0}^{\infty} r(k)(\mathbf{A}^k)_{ii}\,. \tag{6.4}$$

[2]Of course, Google's actual algorithm (which is not in the public domain) is far more complicated than what is presented here.

From a sociological perspective, one can construe $kc(i)$ as the "karma" of vertex i, as it measures its ability to receive its own influence via the graph.

6.5.1 Subgraph centrality

Subgraph centrality was introduced in [100] and it characterizes the participation of each node in all subgraphs in a network. The number of closed walks of length k that start and end on vertex i in a network is given by $(\mathbf{A}^k)_{ii}$, which is the ith diagonal entry of the kth power of the adjacency matrix $\mathbf{A}$. The subgraph centrality of the vertex i is defined as the "sum" of closed walks of different lengths (trivial as well as nontrivial closed walks) in a network that start and end at vertex i. Shorter walks are given more weight than longer walks. The subgraph centrality is

$$sc(i) = \sum_{k=0}^{\infty} \frac{(\mathbf{A}^k)_{ii}}{k!}.$$

Subgraph centrality can be obtained from the spectra of the adjacency matrix [100], giving

$$sc(i) = \sum_{j=1}^{n} (v_j^i)^2 e^{\lambda_j},$$

where $\lambda_1 \geq \lambda_2 \geq \ldots$ are the eigenvectors of the adjacency matrix $\mathbf{A}$, and $v_1, v_2, \ldots$ are the corresponding eigenvectors.

6.6 Discriminant power of centrality measures

The work in this section appears in a working paper by MPR and M. A. Porter [237]. It was noted in Ref. [97] that centrality measures such as degree, closeness, betweenness, and eigenvector centralities are unable to distinguish between any of the vertices on certain graphs, even for graphs that are not *vertex-transitive*. By definition, a graph is "vertex-transitive" if its vertices are indistinguishable: in other words, for all pairs of vertices i and

j, there exists an automorphism $f : V(G) \to V(G)$ such that $f(i) = j$. We now demonstrate this using a few examples.

In Fig. 6.2, we show the Frucht graph, which is a cubic (*i.e.*, 3-regular) graph on 12 vertices that has no nontrivial symmetries (*i.e.*, no automorphisms other than the identity). This is the smallest regular graph with this property [17]. Because degree centrality and eigenvector centrality can be described purely in terms of the degrees of vertices and their neighbours, these two measures fail to distinguish between the vertices of the Frucht graph.

In Fig. 6.3 shows a graph on 6 vertices that is clearly not vertex-transitive (as there exist vertices of degrees 3 and 4) for which betweenness centrality is unable distinguish between the vertices [97]. Such graphs are called *betweenness-self-centric*, and Ref. [118] discusses more examples of such graphs.

In Fig. 6.4, we show a graph on 8 vertices that is not vertex-transitive (three vertices form part of a triangle, but the other five do not) for which closeness centrality is unable to distinguish the vertices from each another [97]. In this example, degree centrality and eigenvector centrality are also unable to distinguish vertices from each other.

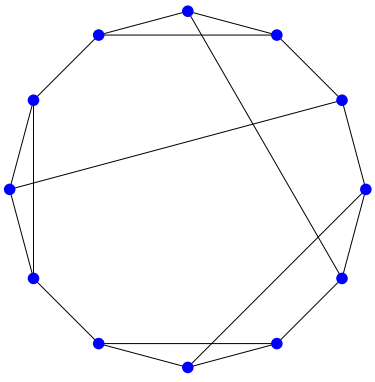


Figure 6.2: The Frucht graph, which is an example of a asymmetric graph on which degree and eigenvector centrality measures are unable to distinguish the vertices.

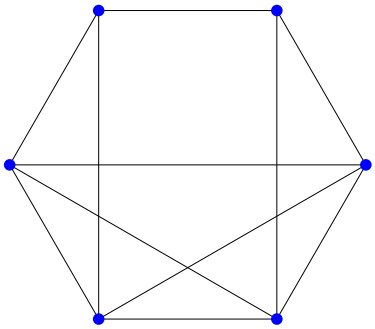


Figure 6.3: A graph on 6 vertices that is not vertex-transitive and on which betweenness centrality is unable to distinguish the vertices.

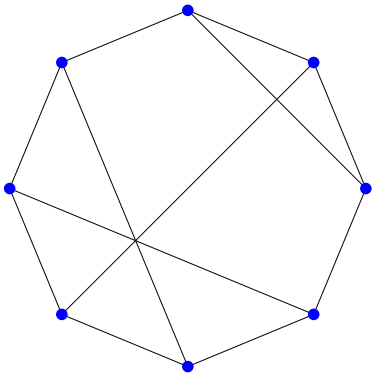


Figure 6.4: A graph on 8 vertices that is not vertex-transitive on which closeness, degree, and eigenvector centralities are unable to distinguish the vertices.

6.7 Estrada's Conjectures

E. Estrada posed the following two conjectures concerning the discriminating power of subgraph centrality.

Conjecture 6.7.1. *[100]* $\text{var}(\mathbf{sc}_G) = 0 \Rightarrow \text{var}(\mathbf{d}_G) = \text{var}(\mathbf{ec}_G) = \text{var}(\mathbf{cc}_G) = \text{var}(\mathbf{bc}_G) = 0.$

Conjecture 6.7.2. *[97]* $\text{var}(\mathbf{sc}_G) = 0 \Leftrightarrow G$ *is walk-regular.*

Here $\mathbf{sc}_G$ denotes the vector of subgraph centralities $sc(i)$ for the nodes of graph G (and similarly for $\mathbf{d}_G$, **ec**, **cc**, and **bc**), and $\text{var}(sc_G)$ denotes the variance of $\mathbf{sc}_G$:

$$\text{var}(\mathbf{v}) = \frac{1}{n}\sum_i v_i^2 - \left(\frac{1}{n}\sum_i v_i\right)^2. \tag{6.5}$$

When $\text{var}(\mathbf{sc}_G) = 0$, all entries are the same.

In this section, we show that Conjecture 6.7.1, which was originally posed in [100], is at least partially false by giving a counterexample for betweenness centrality and closeness centrality.

A graph is called *walk-regular* if, for every $k \in \mathbb{N}$, the number of closed walks in G of length k that start at vertex i is the same for all $i \in \{1, \ldots, n\}$. Because $\mathbf{sc}_i$ depends only on the number of *closed walks* (walks that start and end at the same vertex) of length k that start at i, it follows that

Corollary 6.7.3. *[100]* G *is walk-regular* $\Rightarrow \text{var}(\mathbf{sc}_G) = 0.$

A graph is walk-regular if and only if $\mathbb{P}_i(Z(k) = i) = \mathbb{P}_j(Z(k) = j)$ for all k and for all vertices $i, j \in V(G)$ [122], where $\mathbb{P}_i(Z(k) = i)$ denotes the probability that a random walk that starts at i will be back at its starting point after k steps. All vertex-transitive graphs are walk-regular, but the converse is not true [128].

A *distance-regular* graph is a k-regular graph with diameter $d(G)$ and a set

$$\{b_0, b_1, \ldots, b_{d(G)-1}, c_1, c_2, \ldots, c_{d(G)}\},$$

with $b_0 = k$ and $c_1 = 1$, such that for all vertices $i, j \in V(G)$ with distance r (where the distance is the length of a shortest path between them), the number of vertices in $G_{r-1}(i)$ that are adjacent to j is c_r and the number of vertices in $G_{r+1}(i)$ that are adjacent to j is b_r [56,216]. All distance-regular graphs are walk-regular, but a distance-regular graph need not be vertex-transitive [64].

There are walk-regular graphs that are neither vertex-transitive nor distance-regular [128].[3] We show an example in Fig. 6.5. This graph on 12 vertices, which we call the *Godsil-McKay graph*, gives a counterexample to Conjecture 6.7.1. Its centrality vectors are

$$\begin{aligned}
\mathbf{sc}_{GM} &\approx 6.7035 \times [1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1], \\
\mathbf{bc}_{GM} &= [7,\ 8,\ 7,\ 8,\ 7,\ 7,\ 7,\ 7,\ 8,\ 7,\ 8,\ 7], \\
\mathbf{cc}_{GM} &= \left[\frac{1}{18},\ \frac{1}{19},\ \frac{1}{18},\ \frac{1}{19},\ \frac{1}{18},\ \frac{1}{18},\ \frac{1}{18},\ \frac{1}{18},\ \frac{1}{19},\ \frac{1}{18},\ \frac{1}{19},\ \frac{1}{18}\right].
\end{aligned}$$

In Fig. 6.5, we use gray diamonds to designate the vertices with a betweenness centrality of 8 and a closeness centrality of $\frac{1}{19}$. We use blue disks to designate vertices with a betweenness centrality of 7 and a closeness centrality of $\frac{1}{18}$.

In Fig. 6.6, we give two examples of graphs that are distance-regular (and walk-regular) but not vertex-transitive. The examples are the Tutte 12-Cage [24, 221, 263] and one of the Chang graphs [67, 277]. The vertices of these graphs cannot be distinguished by degree, eigenvector, closeness, betweenness, or subgraph centralities. The Tutte-12-Cage and the Chang graphs are the only known distance-regular graphs that are not vertex-transitive[4]. Additionally, it has been proven that betweenness centrality cannot distinguish distance-regular graphs.

Theorem 6.7.4. *[118] G is distance-regular* $\Rightarrow \mathrm{var}(\mathbf{bc}_G) = 0$.

[3] Note that the paper [128] has a mistake in the plot of this graph. We plot the correct graph in Fig. 6.5, and one can also find the correct graph in the erratum appended to the paper at `http://cs.anu.edu.au/people/bdm/papers/WalkRegular.pdf`.

[4] See also the discussion at `http://mathoverflow.net/questions/106589/is-every-distance-regular-graph-vertex-transitive`

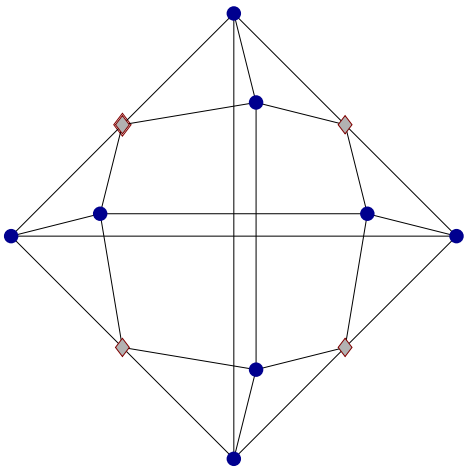
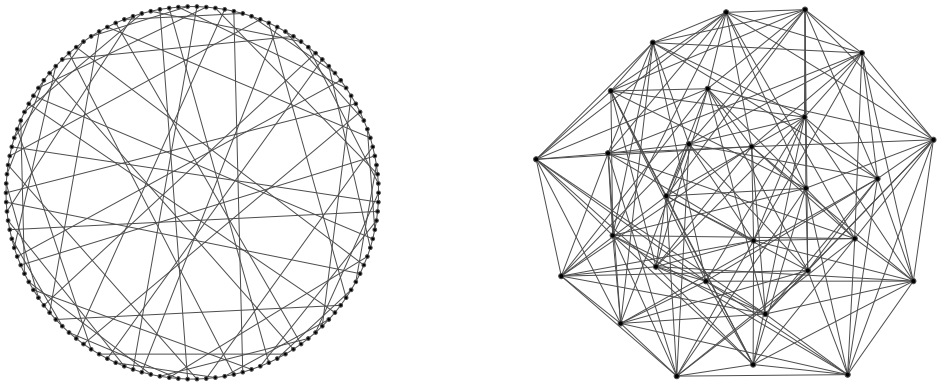


Figure 6.5: The Godsil-McKay graph, which is walk-regular but neither vertex-transitive nor distance-regular [128].



(a) (b)

Figure 6.6: Two examples of graphs that are distance-regular (and hence also walk-regular) but not vertex-transitive: (a) the Tutte 12-Cage [221] and (b) the first Chang graph [277].

In fact, this is also true for closeness centrality:

Theorem 6.7.5. *G is distance-regular* $\Rightarrow \mathrm{var}(\mathbf{cc}_G) = 0$.

Proof. Let $\Gamma^r(i) = \{j | d(i,j) = r\}$ and $1 \leq r \leq d(G)$. This is a generalization of the neighbourhood $\Gamma(i) = \{j | ij \in E(G)\}$. Because we are only considering connected graphs, every $j \in V(G) \setminus i$ has a distinct finite integer $d(i,j)$. Then, for any i, the sets $\Gamma^r(i)$ are pairwise disjoint and $i \cup \Gamma^1(i) \cup \ldots \cup \Gamma^{d(G)}(i) = V(G)$. We consider the subgraphs on two

consecutive sets $\Gamma^r(i)$ and $\Gamma^{r+1}(i)$ that include the edges running between the two sets, and we denote these bipartite graphs by B_r. A *biregular graph* is a bipartite graph in which two vertices on the same side of the given partition have equal degrees. Let j be an arbitrary vertex in $\Gamma^r(i)$ and k an arbitrary vertex in $\Gamma^{r+1}(i)$. From the definition of a distance-regular graph, we know that

$$|\Gamma(j) \cap \Gamma^{r+1}(i)| = b_r,$$
$$|\Gamma(k) \cap \Gamma^{r}(i)| = c_{r+1}.$$

Therefore, every B_r is a biregular graph. It follows that $|E(B_r)| = b_r|\Gamma^r(i)| = c_{r+1}|\Gamma^{r+1}(i)|$, for all bipartite graphs B_r. It follows that

$$|\Gamma^r(i)| = \frac{b_{r-1}b_{r-2}\dots b_0}{c_r c_{r-1}\dots c_1}, \tag{6.6}$$

independently of our choice of i. Because

$$\mathbf{cc}_i = \frac{1}{n}\sum_{j\in V\setminus i} d(i,j) = \frac{1}{n}\sum_{r=1}^{d} r|\Gamma^r(i)|, \tag{6.7}$$

it follows that $\mathrm{var}(\mathbf{cc}_G) = 0$ for distance-regular graphs. □

The converse of Thm. 6.7.5 is not true (see Fig. 6.4 for a counterexample).

Based on the results in Section 6.7, we propose the following conjecture, which we call the "Modified Estrada Conjecture".

Conjecture 6.7.6. $\mathrm{var}(\mathbf{sc}_G) = 0 \Leftrightarrow$ *G is walk-regular, and*
$\mathrm{var}(\mathbf{sc}_G) = \mathrm{var}(\mathbf{bc}_G) = \mathrm{var}(\mathbf{cc}_G) = 0 \Leftrightarrow$ *G is distance-regular.*

Note that the $\Leftarrow$ part of both statements in Conjecture 6.7.6 are known to be true because of Corollary 6.7.3 and Theorems 6.7.4 and 6.7.5 (and the fact that all distance-regular graphs are walk-regular).

6.8 Centrality and core-periphery structure

The notion of centrality in a network is closely related to the notion of a *core-periphery structure*. Both aim to rank nodes by their taking a central position within a network (however such a position may be defined). In a core-periphery structure, this central position usually includes a significant proportion of the nodes that form a central structure together, both by being well connected to each other and by being well-connected to the rest of the network. As we will see in Chapter 7, methods have been proposed to define core-periphery structure in terms of existing centrality measures, such as the ones we described in this chapter. Instead of defining a binary core-periphery organisation (where nodes are either in a unique core or a unique periphery, without any further quantification of their role), one can also define a continuous core-periphery structure, where nodes are scored according to how strongly they belong to a core. The latter is also a centrality measure (even the former is, albeit a crude one that can only take one of two values).

Chapter 7

Core-Periphery Measures in Networks

Much of the material presented in this chapter, in particular Section 7.4 and beyond, is written up in the article in [238], entitled "Core-Periphery Structure in Networks", by M.P. Rombach, M.A. Porter, J.H. Fowler and P.M. Mucha. MPR is the main author and main contributor to this paper, but gratefully acknowledges the guidance, helpful discussions and feedback, as well as iterations of the manuscript in the writing up stage of the article from the other three authors.

7.1 Introduction

As was the case with community structure in Chapter 5, core-periphery structure is an intuitive concept, but it is hard to pin down mathematically. Again, different definitions and detection methods have been proposed. A core-periphery structure of a network is a division of the nodes into a densely connected core and a sparsely connected periphery. The nodes in the core should also be reasonably well-connected to the nodes in the periphery, but nodes in the periphery should not be well-connected to each other, or to the core. Hence, a node belongs to the core if and only if it is well-connected both to other nodes in the core and to nodes assigned to the periphery. A core structure in a network is thus not merely densely connected but also tends to be *central* to the network in some other way.

There are different ways in which node *centrality* can be defined, and they may give rise to analogous notions of core-periphery structure. See Chapter 6 for an extensive discussion of *centrality* measures. Indeed, for centrality measures such as degree [72], betweenness [215], closeness [215], and eigenvector centrality [50, 82] analogous core-periphery definitions have been proposed.

Several results underscore the importance of considering core-periphery structure in addition to community structure. For example, Fan Chung and Linyuan Lu [72] showed that power-law random graphs, in which the number of nodes of degree k is proportional to $k^{-\beta}$, almost surely contain a dense subgraph that has short distance to almost all other nodes in the graph when the exponent $\beta \in (2, 3)$. They show that such graphs contain a structure that can be described as an "octopus": a combination of a dense core with small *diameter* (maximum distance between any pair of vertices in a graph) and tentacles such that the remaining vertices have bounded distance from the core. More precisely, they showed that, in power-law random graphs as described above, the diameter of the core is almost surely $O(\log \log n)$, where the core is the set of vertices with degree at least $n^{1/\log \log n}$. Also, almost all vertices with degree at least $\log n$ are almost surely within distance $O(\log \log n)$ from the core.

This suggests that it is sensible for networks with heavy-tailed degree distributions to contain some sort of cohesive core, and there is strong evidence that this is indeed the case in many real-world networks (such as many social networks and the World Wide Web) [31, 211, 271]. Social scientists have given attention to core-periphery structures in networks before network scientists started to formalise the notion. Snyder and Kick presented networks of international relations, based on *e.g.* trade flow and diplomatic relations, and provide evidence for a block-model core-periphery structure in this data, as was expected theoretically [253]. Nemeth and Smith, and later Smith and White noticed core-periphery structure in international trade [206, 251]. Core-periphery structures were also found in networks of national trade [240, 282], academic journals [85, 283], human

social networks [133,279], social networks in monkeys [79,247], and interactions between scientists [55,115].

The rest of this chapter is organised as follows. To tie this chapter back to Chapter 5, we briefly discuss the simultaneous existence of core-periphery structure and community structure in Section 7.2, and we will discuss such a coexistence in Sections 7.6.1 though 7.6.4. In the rest of the chapter we will examine the ways in which core-periphery structures have been defined thus far, such as the block model approach by Borgatti and Everett (Section 7.3.1), the minimum residual method (Section 7.3.2), the method of k-cores (Section 7.3.4), and core-periphery structure in terms of network capacity (Section 7.3.6). We propose a new method of detecting such structures in Section 7.4. In Section 7.5, we compare several different methods when applied to a random network with an artificial core-periphery structure. Finally, in Section 7.6, we examine several real-world networks in terms of their core-periphery organisation – the infamous Zacharay Karate Club network (Section 7.6.1), the network of the London Underground (Section 7.6.2), a co-authorship network of network scientists (Section 7.6.3), and a voting similarity network of US Senators (Section 7.6.4).

7.2 Simultaneous core-periphery and community structure

Networks can have a core-periphery structure, community structure, or both a core-periphery structure and a community structure. In Figure 7.1, we show images of the adjacency matrices of idealized block models that illustrate (a) community structure, (b) core-periphery structure, (c) a global core-periphery structure with a local community structure, and (d) a global community structure with a local core-periphery structure. By permuting rows and columns of the adjacency matrix, one can see that (c) and (d) are equivalent. Of course, core-periphery and community structure can interact in other ways as well. For exam-

ple, one could think of a network where one of the communities is relatively dense and well-connected and thus also forms the core of the network. Or, multiple communities can overlap and thereby create a core-periphery structure, as was proposed by Leskovec [286] (see Section 7.3.3). In most cases, core-periphery and community detection methods provide different sets of information about a network, as we shall see in the empirical examples in Sections 7.6.1 through 7.6.4.

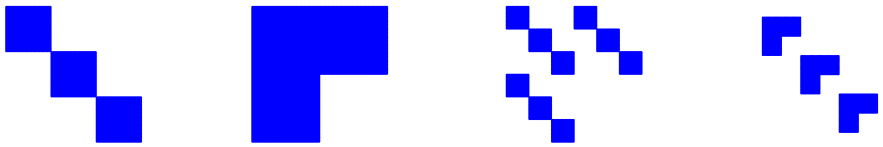


Figure 7.1: Examples of network block models. (a) Community structure, (b) core-periphery structure, (c) global core-periphery structure with local community structure, and (d) global community structure with local core-periphery structure. Note that (c) and (d) are equivalent.

7.3 Existing methods

In this section, we present a brief overview of techniques to detect core-periphery structure algorithmically. Arguably the notion started to be formalised in the setting of network science by Borgatti and Everett in 1999 [50] (see Section 7.3.1). The methods described form the basis of the algorithms implemented in the popular social science analysis software UCINet, where the default method uses a form of the MINRES algorithm described in Section 7.3.2. Other methods utilise notions from graph theory - such as k-cores, random walks, network capacity (see Sections 7.3.4 through 7.3.6).

7.3.1 Borgatti & Everett

The notion of a core-periphery structure was formalized in social networks by Borgatti and Everett in 1999 [50], who proposed algorithms for detecting both discrete and continuous

versions of core-periphery structure in weighted, undirected graphs. Their discrete notion of core-periphery structure is based on comparing a network to an ideal block model that consists of a fully-connected core and a periphery that has no internal edges but is fully connected to the core. Their algorithm aims to find a vector C of length n whose entries can either be 1 or 0. The ith entry C_i equals 1 if the corresponding node is assigned to the core, and it is 0 if the corresponding node is assigned to the periphery. Let $C_{ij} = 1$ if $C_i = 1$ or $C_j = 1$ and $C_{ij} = 0$ otherwise, and define

$$\rho_C = \sum_{i,j} A_{ij} C_{ij} \,, \tag{7.1}$$

where A_{ij} are the elements of the graph's adjacency matrix. The matrix element A_{ij} represents the strength (*i.e.*, weight) of the tie between nodes i and j, and it equals 0 if nodes i and j are not connected. The Borgatti-Everett method seeks to maximize ρ_C for every given number of nodes in the core. It then compares this maximal value to the expected value of ρ if C is shuffled such that the number of 1 and 0 entries are preserved but their order is randomized. The output is the vector C that gives the highest z-score (number of standard deviations away from the expected value) for ρ_C.

As a variation of the above discrete notion of core-periphery structure, Borgatti and Everett defined

$$C_{ij} = \begin{cases} 1 \,, & \text{if } C_i \text{ and } C_j = 1 \,, \\ a \in [0,1] \,, & \text{if } C_i = 1 \text{ or } C_j = 1 \,, \text{ and not both} \\ 0 \,, & \text{otherwise.} \end{cases} \tag{7.2}$$

They also defined a continuous notion of core-periphery structure in which a node is assigned a 'coreness' value of C_i and $C_{ij} = C_i \times C_j = a$. When using this model of core-periphery structure, one is essentially approximating the adjacency matrix A_{ij} by a *structure matrix* CC^T.

7.3.2 The Minimum Residual Method

The *minimum residual method* (MINRES) [77], is a method of *factor analysis*. Factor analysis is used to describe observed correlations between variables in terms of a lower number of unobserved variables, called the factors [81]. The MINRES method aims to find a vector C, such that the following sum is minimized:

$$F(A,C) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} \left(A_{ij} - C_i C_j\right)^2,$$

where $C_i \geq 0$ for all i. Note that the diagonal elements of the adjacency matrix are ignored, and that, because CC^T is symmetric this method works best for undirected network adjacency matrices A (for directed networks, the method can be complemented with a singular value decomposition method to model the asymmetry [51]). In practice, UCINet reports $C/\sqrt{\sum_i C_i^2}$. Differentiating gives

$$C_i = \frac{\sum_{\substack{j=1 \\ j\neq i}}^{n} A_{ij} C_j}{\sum_{\substack{j=1 \\ j\neq i}}^{n} C_j^2}.$$

For our computations, in order to have more control over the script, we use a MATLAB script that uses the above equations in a simple iterative process (following the outline in [77]), starting from the vector of degrees. Note that this is essentially the same as starting from a vector $\underline{1} = (1,\ldots,1)^T$.

7.3.3 Overlapping communities

In Section 7.2 we discussed the possible interaction between community structure and core-periphery structure. In [286] Yang and Leskovec argue that in some cases, core-periphery structure is an effect of community structure containing overlapping communities. They studied large social, collaboration, and information networks, where the ground truth, *i.e.*

the community memberships as declared by the nodes themselves, is known. They found that the overlaps between communities contained more high-degree nodes and were therefore denser than expected (by existing models of community structure). They present a model, called the *community-affiliation graph model*, to capture dense overlaps between communities. In this model, the likeliness of two nodes connecting to each other is proportional to the number of communities they have shared membership to.

7.3.4 Core-periphery in terms of k-Cores

A k-core of a graph is a maximal connected subgraph such that every vertex has degree at least k within the subgraph. In other words, it is a graph obtained by repeatedly deleting vertices of degree less than k until that is no longer possible (this process may lead to multiple, disconnected k-cores). Both k-cores and the related notion of graph *degeneracy* (the largest value of k such that a graph has a k-core) are well-studied in the field of graph theory and they form a generalization of connected components (1-cores) [40, 183]. In particular, in random graph theory and statistical mechanics attention has been given to their sudden emergence in random graph processes, analogous to the emergence of the giant component (see Section 2.3). In 2005, Holme defined a core-periphery coefficient [146]

$$c_{cp}(G) = \frac{C_C\left(V_{\text{core}}(G)\right)}{C_C\left(V(G)\right)} - \left\langle \frac{C_C\left(V_{\text{core}}(G')\right)}{C_C\left(V(G')\right)} \right\rangle_{G' \in \mathcal{G}(G)}, \tag{7.3}$$

where the angled brackets indicate the average and $\mathcal{G}(G)$ is an ensemble of graphs with the same degree sequence as the unweighted, undirected graph G,

$$C_C(U) = \left(\left\langle \langle d(i,j) \rangle_{j \in V \setminus \{i\}} \right\rangle_{i \in U} \right)^{-1}, \tag{7.4}$$

and $d(i,j)$ is the distance (i.e., number of edges in the shortest path) between nodes i and j. A k-core of the graph G is a maximal connected subgraph in which all nodes have degree at least k, and V_{core} is the k-core with maximal $C_C(U)$. Using k-cores to examine core-

periphery structure is computationally fast (and we note that one can generalize Holme's method for weighted graphs [120]), but it entails very strong restrictions on the notion of a network core.

7.3.5 Random walkers

We have already seen community detection methods using random walkers (Section 5.4). Della Rossa *et al.* recently proposed a method of detecting a continuous core-periphery profile of a (weighted) network by studying the behaviour of a random walker on the network [82]. Let w_{ij} be the weight of the edge ij. A random walker currently at node i jumps to node j with probability $p_{ij} = w_{ij} / \sum_h w_{ih}$. Let π_i be the asymptotic probability of visiting node i. When a network is undirected, π_i has the closed form solution $d_i / \sum_{j \in V(G)} d_j$, where d_i is the degree of node i, or, in the weighted case, the strength. Let α_S be the *persistence probability*, or the probability that a random walker currently on a node in a set of nodes S remains in the set S at the next time step:

$$\alpha_S = \frac{\sum_{i,j \in S} \pi_i p_{ij}}{\sum_{i \in S} \pi_i}. \tag{7.5}$$

In an ideal core-periphery structure, one could find the periphery by finding the largest subset S of nodes such that $\alpha_S = 0$, but in empirical networks we are unlikely to find such an ideal structure. Therefore, Della Rossa *et al.* proposed the notion of an α-periphery: the largest subnetwork S with $\alpha_S \leq \alpha$. They provided a heuristic for finding such a periphery by building a periphery node by node while keeping it as weakly connected as possible.

7.3.6 Network capacity

One expects a core of a network to have high connectivity to other parts of the network, so Da Silva *et al*. introduced a measure of connectivity known as network *capacity* [215]:

$$K = \sum_{l=1}^{M} P_l^{-1}\,, \tag{7.6}$$

where M is the total number of connected pairs of nodes and P_l is the length of the shortest path between the lth pair of nodes. Da Silva *et al*. then defined a core coefficient as $cc = n'/n$, where n is the total number of nodes in the network, n' satisfies $\sum_{m=0}^{n'} K_m = 0.9 \sum_{u=0}^{n} K_u$, and K_m is the capacity of the network after the removal of m nodes[1]. The nodes are removed in reverse order of closeness centrality (so the nodes with the lowest centrality are removed first), which is defined as the mean shortest path from a node to each of the other nodes in a network [215]. Note that in the remainder of this chapter, we will use the following definition for closeness centrality of a node j (there are several different ones available in the literature):

$$CC_j = \frac{1}{n} \sum_{i \in V} P(i, j)\,,$$

where $P(i, j)$ is the sum of edge weights in a shortest path in the context of weighted networks. Da Silva *et al*. considered only binary networks, but their method can be generalized straightforwardly to weighted networks.

7.3.7 Modularity

In the same paper [215] that we just discussed in Section 7.3.6, Da Silva *et al*. considered using modularity to detect core-periphery structure by treating the core as a dense, well-connected community (see also Section 7.2). On their example networks (metabolic networks), they found that the algorithm tends to split the core in many modules, because the

[1] The constant was chosen to be .9 based on results on empirical metabolic networks, so other choices may be more appropriate in different settings.

core is too well-connected elsewhere in the network. To avoid this problem, they proposed a method that first extracts the core nodes according to a centrality measure (closeness, betweenness, or degree) and then divides the rest of the network into communities optimising modularity. This is not a method that uses modularity alone to find a core-periphery structure. The idea of a method that employs a form of modularity maximization adapted specifically to core-periphery detection, rather than straightforwardly using the notion that was designed for community detection, has not received enough attention. See also Section 9.4.

7.4 A new method: core score

The following work is part of a collaboration with M. A. Porter, J. H. Fowler, and P. J. Mucha and appears in [238]. Our new method to study core-periphery structure in weighted, undirected networks is motivated by the continuous formulation of Borgatti and Everett [50] that we described in Section 7.3.1. However, our method takes cores of different size and shapes into account. The idea for this method was partly motivated by the coexistence of community structure and core-periphery structure (see Section 7.2). We will support this when we study the examples in Sections 7.6.1 through 7.6.4. By taking cores of different size and shapes into account, the method gives credit to all nodes that take part in a core, and it weights this credit by the quality of the associated core. As we discuss below, we employ a *transition function* to interpolate between core and periphery nodes. Additionally, we construct elements C_{ij} of a *core matrix* to compute the quality of a core. We will present several viable choices for both the transition function and the core matrix.

Define the *core quality*

$$R_\gamma = \sum_{i,j} A_{ij} C_{ij} \,, \tag{7.7}$$

where γ is a vector that parametrizes the core quality (see the discussion below), the elements C_{ij} of the core matrix are given by $C_{ij} = f(C_i, C_j)$, and $C_i \geq 0$ is the *local core*

value of the *i*th node. The local core values are elements of a *core vector* C. Our example calculations in this chapter usually use a product form

$$C_{ij} = C_i C_j \,, \tag{7.8}$$

but we discuss other viable choices in Section 7.4.1.

We seek a core vector C that maximizes R_γ and is a normalized (so that its entries sum to 1) shuffle of the vector C^* whose components $C_i^* = g(i)$ are determined using a *transition function* g. The number of components of the vector C^* is equal to the number of nodes in the network, and C_i^* gives the local core value of the ith node. Our example calculations in this chapter usually use the transition function given by the sharp (because it has a discontinuous derivative with respect to i – thinking of i as a continuous variable –, and is discontinuous itself in the limit $\alpha = 1$) function

$$C_i^*(\alpha,\beta) = g_{\alpha,\beta}(i) = \begin{cases} \frac{i(1-\alpha)}{2\beta n} \,, & i \in \{1,\ldots,\beta n\} \,, \\ \frac{(i-\beta n)(1-\alpha)}{2(n-\beta n)} + \frac{1+\alpha}{2} \,, & i \in \{\beta n+1,\ldots,n\} \,. \end{cases} \tag{7.9}$$

The parameter β sets the size of the core: as β varies from 0 to 1, the number of nodes included in the core varies from n to 0. We think of the first βn nodes as the nodes in the periphery. The parameter α sets the size of the score jump between the highest scoring periphery node and the lowest scoring core node. In the limit in which $\alpha = 1$, this yields a discrete classification into a unique core and unique periphery that assigns each node to either the core or the periphery.

With the transition function (7.9) and the product form (7.8) for the core-matrix elements, the core quality is given by

$$R_\gamma = R_{\alpha,\beta} = \sum_{i,j} A_{ij} C_{ij} = \sum_{i,j} A_{ij} C_i C_j \,. \tag{7.10}$$

For a given value of $\gamma = (\alpha, \beta)$, we seek a shuffle C of C^* such that R_γ is maximized.

For any choice of core matrix and transition function, we define the aggregate *core score* of each node i as

$$CS(i) = Z \sum_{\gamma} C_i(\gamma) \times R_\gamma , \tag{7.11}$$

where the normalization factor Z is chosen so that $\max_k[CS(k)] = 1$, where $k \in \{1, \ldots, n\}$ indexes the nodes. A core score gives a notion of network centrality [211, 271]. As discussed above, our usual choice in this chapter is to maximize the core quality (7.10) that uses the product form (7.8) for the core matrix and the sharp transition function (7.9) to interpolate between core and periphery nodes. See Section 7.4.1 for a discussion of other choices for constructing the core matrix and Section 7.4.2 for other choices of transition function.

In the results that we present here, we assign the values of $C_i^*(\alpha, \beta)$ to the nodes to obtain a $C_i(\alpha, \beta)$ that maximizes $R_{\alpha,\beta}$ using a simulated-annealing algorithm [163][2]. Other computational heuristics can, of course, be faster. In all of our examples, we sample α and β uniformly over a discretization of the square $[0, 1] \times [0, 1]$. In particular, we always use $\alpha = \beta = [0.01 : 0.01 : 1]$ (in MATLAB notation).

It is also interesting to consider the core quality of specific values of α and β, and one could in principle improve the speed of our general approach by developing procedures for choosing α and β selectively in a manner that takes advantage of the structure of particular networks or families of networks. We consider an example of this in Chapter 8, where we apply our method to human brain networks obtained from fMRI data.

[2]The MATLAB code that we used for simulated annealing was written by Joachim Vandekerckhove [270]. It uses the following parameters: an initial temperature of 1; a final temperature of 10^{-8}; a cooling schedule of $.8 \times T$ (where T represents the temperature); a maximum number of consecutive rejections of 1000; a maximum of 300 tries at one given temperature; and a maximum of 20 successes at one given temperature. See also Section 5.6.2

7.4.1 Ideal cores

In most of the calculations in this chapter, we construct the core-matrix elements C_{ij} using a product form $C_{ij} = C_i C_j$. However, other choices are also viable. An idealized core-periphery structure entails that core nodes are well-connected to other core nodes as well as to periphery nodes and that periphery nodes are not well-connected to each other. Let v_1 and v_2 be core nodes and let w_1 and w_2 be peripheral nodes. We then want $C_{w_1 w_2}$ to be small and $C_{v_1 v_2}$ and $C_{v_i w_j}$ to be large. For example, the block structure in panel (b) of Figure 7.1 satisfies these conditions.

As one can see from Figure 7.2, one can try to approximate such an idealized block structure using various ways of constructing C_{ij}. For example, in addition to the product form (7.8), one can instead use a p-norm and write

$$C_{ij} = \|(C_i, C_j)\|_p = \sqrt[p]{C_i^p + C_j^p}\,. \tag{7.12}$$

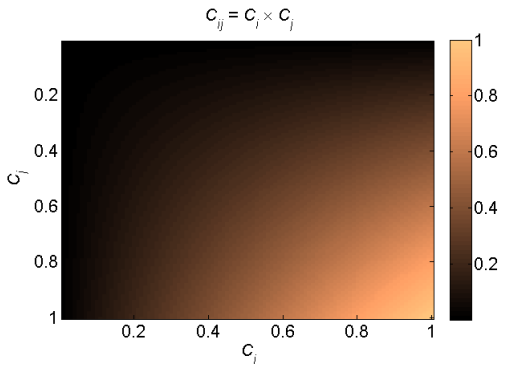
As one considers progressively larger p, this will look more and more like an ideal core-periphery block model (in which core-core edges and core-periphery edges produce a value of 1 in a network adjacency matrix, but periphery-periphery edges produce a value of 0).

7.4.2 Transition functions

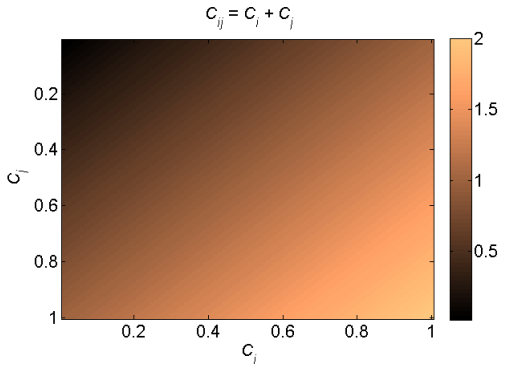
Our methodology to compute core-periphery structure entails choosing a transition function to interpolate between core and periphery nodes. In most of the calculations in this chapter, we use the sharp two-parameter function (7.9) to illustrate our approach. However, there are many other viable choices for the transition function.

One variant is to construct the vector C^* using a smooth transition function $g(i)$. For example, one possibility is

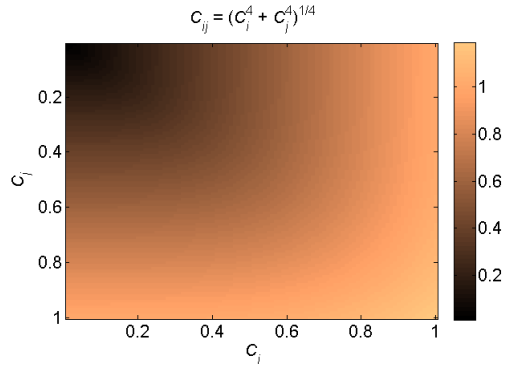
$$C_i^*(\alpha,\beta) = g_{\alpha,\beta}(i) = \frac{1}{1 + \exp\{-(i - n\beta) \times \tan(\pi\alpha/2)\}}\,, \tag{7.13}$$



(a)



(b)



(c)

Figure 7.2: Several options for the core-matrix element C_{ij} include (a) the product form $C_{ij} = C_i \times C_j$, (b) the 1-norm $C_{ij} = \|(C_i, C_j)\|_1 = C_i + C_j$, and (c) the 4-norm $C_{ij} = \|(C_i, C_j)\|_4 = \sqrt[4]{C_i^4 + C_j^4}$.

which has parameters $\alpha \in [0,1]$ and $\beta \in [0,1]$. The parameter α sets the sharpness of the boundary between the core and the periphery. The tan function is there so the sharpness can be set between 0 and 1, rather than 0 and ∞. The value $\alpha = 0$ yields the fuzziest boundary and $\alpha = 1$ gives the sharpest transition: as α varies from 0 to 1, the maximum slope of C^* varies from 0 to ∞. The parameter β sets the size of the core: as β varies from 0 to 1, the number of nodes included in the core varies from n to 0.

Another option, which allows our method to be significantly faster, is a transition function that has only one parameter. One can choose such a parameter to control the size of the core, the sharpness of the boundary, or some combination of the two. For example, one possibility is

$$C_i^*(\alpha) = g_\alpha(i) = \frac{1}{2}\tanh\left(8\exp\left\{\frac{-10 * (\alpha - n/2)^2}{2}\right\}(i-\alpha) + 1\right). \tag{7.14}$$

We plot (7.14) for various values of α in Figure 7.3. One can then average over values of α to produce aggregate core scores.

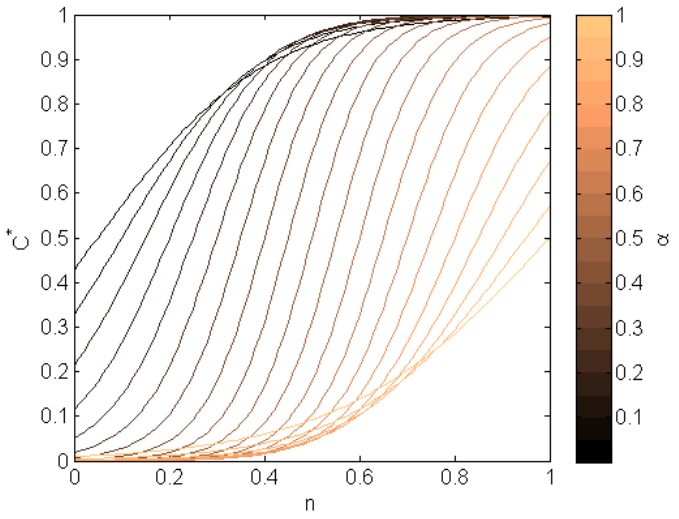


Figure 7.3: An example of a one-parameter transition function in which the parameter α controls both the size of a network core and the sharpness of the boundary between core and periphery nodes. This is function (7.14).

In this chapter, we calculate aggregate core scores using formulations with both two-parameter and one-parameter transition functions. In the former case, we always average over 10000 values of (α,β) that are sampled uniformly from $[0,1] \times [0,1]$. (In particular, we use $\alpha = \beta = [0.01 : 0.01 : 1]$.) In the latter case, we always average over 10000 values of α that are uniformly sampled from $[0,1]$. (In particular, we use $\alpha = [0.0001 : 0.0001, 1]$.)

7.4.3 Interpreting core scores

There are several ways to use and interpret the results of our approach for studying core-periphery structure. One can average over a set of parameter values—e.g., in the (α,β) parameter plane if one uses a two-parameter transition function—and obtain a set of aggregate core scores that yield a continuous centrality measure for the nodes in a network. Alternatively, one can compute a core-periphery structure at a single set of parameter values, such as the one that produces the largest value of the core quality R (7.7). (See the discussion of the Zachary Karate Club network in Section 7.6.1.) Sometimes, as with the London Underground network in Section 7.6.2, one can observe a clear dichotomy between core and periphery nodes after calculating continuous core scores. Finally, it can be useful to impose a specific core size in advance (and thereby dichotomize core and periphery nodes), as we do with the synthetic benchmark networks in Section 7.5.

The flexibility described in the above paragraph is a beneficial feature of our method, which can be used either to produce a continuum of core scores or a discrete classification of core versus periphery. The utility for both of these perspectives, and hence the desirability for the development of methods to study core-periphery structure that have such flexibility, was recognized more than two decades ago [50, 68, 251]. For example, studies of international relations include vehement arguments as to whether countries should be classified discretely (e.g., into core, semiperipheral, and peripheral countries) or along a continuum [251], and methods that can produce both discrete and continuous perspectives on core-periphery structure ought to be helpful.

7.5 Synthetic benchmark networks

In this section, we examine our method using an ensemble of random networks with an imposed core-periphery structure to demonstrate that it performs well at detecting the kind of core-periphery structure envisioned by Borgatti and Everett [50].

We define a family of synthetic networks that only have a core-periphery structure [see Figure 7.1(b)], and we use $CP(n, d, p, k)$ to denote this ensemble of networks. (We will consider networks with both core-periphery structure and community structure when we examine real networks. For example, see the London Underground network in Section 7.6.2 and the network of network scientists in Section 7.10.) Each network in the ensemble $CP(n, d, p, k)$ has n nodes, where dn of the nodes are core nodes, $(1 - d)n$ of the nodes are peripheral nodes, and $d \in [0, 1]$. The edges are assigned independently at random. The edge probabilities for periphery-periphery, core-periphery, and core-core pairs are p, kp, and kp, respectively. Note that $p \in [0, 1]$ and $k \in [1, (1/p)^{1/2}]$. We fix $n = 100$, $d = 1/2$, and $p = 1/4$ and compute the core-periphery structure averaged over 100 different instances of $CP(n, d, p, k)$ for each of the parameter values $k = 1, 1.1, 1.2, \ldots, 2$. In Figure 7.4, we show our results of determining core nodes by computing the aggregate core score (7.11) with core quality (7.10) and transition function (7.9). The synthetic networks in $CP(n, d, p, k)$ possess a discrete core-periphery structure, whereas our method produces a continuous ranking, which we recall makes the aggregate core score a notion of centrality.

We also examine the results of attempting to determine the core nodes using various types of centrality: closeness (see Section 6.3.1 and 7.3.6), degree (see Section 6.2), PageRank[3] (see Section 6.4.2), geodesic node betweenness (see Section 6.3.2), and MINRES (see Section 7.3.2), which are designed to measure notions of node importance. We only test continuous node-ranking notions, which we evaluate by counting how many of the 50 core nodes—recall that the networks have $nd = 50$ core nodes by construction—are placed in the top 50 according to each method. In Figure 7.4, we show the fraction of nodes that are

[3]The MATLAB code that we used for finding PageRank centrality was written by David Gleich [127].

correctly identified as one of the top 50 core nodes. When testing the methods, we used a random permutation of the labels of the nodes to try to prevent any bias. In this case, none of the tested methods should suffer from such a bias. (Note that our method starts the optimization with a random permutation of the vector C^*.)

As we have indicated, our method examines core-periphery structure as a type of centrality. Nodes are more likely to be part of a network's core if they have high strength (i.e., weighted degree) *and* if they are connected to other core nodes. Neither notion of importance is sufficient on its own. Nodes with high degree are construed as important in many situations, and the latter idea is reminiscent of quantities like eigenvector centrality and PageRank centrality [217], which recursively define nodes as important based on having connections to other nodes that are important [211]. We will also compare core scores with notions of centrality when we discuss political voting-similarity networks in Section 7.6.4.

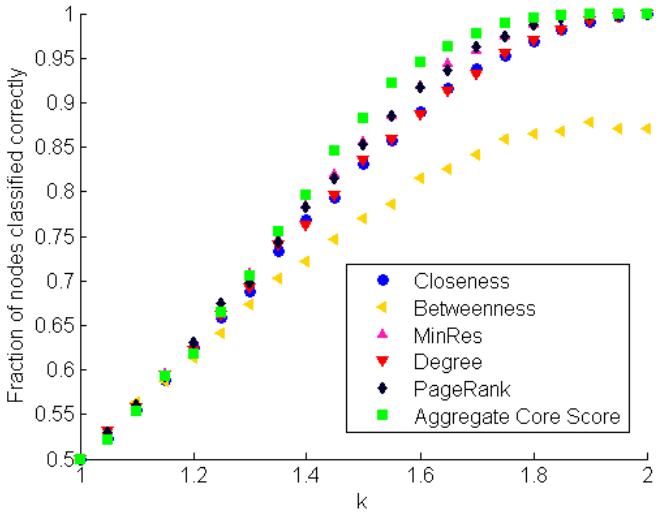


Figure 7.4: Fraction of core nodes correctly identified by computing aggregate core score averaged over 100 realizations of networks in the ensemble $CP(100, .5, .25, k)$. We compute the aggregate core score (7.11) using the core quality (7.10) and the transition function (7.9).

7.6 Empirical networks

We examine several real-world networks in terms of their core-periphery organisation. These examples are case studies that are intended to provide insights into how the method works and what structures it uncovers. We start with the famous Zachary Karate Club network of social ties in a university sports club, which is a small and (to most network scientists) familiar example. The rest of the examples cover a wide range of types of networks. First we discuss the London underground public transport network, then a network of co-authorships between scientists working in the field of network science, and, finally, a so-called *similarity* network of senators. In this network, the edges between two senators are weighted by the fraction of roll call votes that they agreed on. In Chapter 8, we will investigate an empirical network in much more detail using a range of methods, including this one. There we study similarity networks of regions in the brain, based on their activity levels over time.

7.6.1 The Zachary Karate Club

We first consider the infamous Zachary Karate Club network [287], which consists of friendship ties between 34 members of a university karate club in the United States in the 1970s. (We use the unweighted version of this network.) A conflict led the club to split into two new clubs, and the (unweighted) Zachary Karate Club network has become one of the standard benchmark examples for investigations of community structure [108, 229].

We visualize the network in Figure 7.5, where we have identified the nodes according to the split that occurred as a result of a longstanding disagreement between the instructor (Mr. Hi) and the club president (John A.)[4]. These two primary actors are represented, respectively, by nodes 1 and 34.

In Table 7.1, we the show the nodes along with their aggregate core scores (7.11) com-

[4]These names are pseudonyms introduced in Ref. [287]

puted using the core quality (7.10) and the transition function (7.9). We also show the node degrees, which have a high positive correlation with the aggregate core scores. Unsurprisingly, the two leaders (Mr. Hi and John A.) (nodes 1 and 34) have the highest aggregate core scores. One can see additional structure by considering all values of the parameters α and β rather than averaging over them. (Recall that we consider $\alpha = \beta = [0.01 : 0.01 : 1]$.) In particular, the fact that node 1 has the highest aggregate core score does not imply that it has the highest value of $C_1^*(\alpha,\beta)$ for all α and β. In Figure 7.6, we show how the top node varies as a function of α and β. Node 1 has the highest core value only about 20% of the time, whereas node 34 is the top node about 74% of the time. However, the values for α and β for which node 34 is the top node have lower core qualities R (7.10) on average than those for which node 1 is at the top. Such nuances are invisible if one attempts to examine coreness using only the notion of degree. Figure 7.6 also shows us that we find different cores for different values of α and β.

Some of the nodes (e.g., 15, 16, 19, 21, and 23) in the Zachary Karate Club network are automorphs of each other (such nodes are *role equivalent* [86, 103]), as one can swap their labels without changing the network structure. In the limit as the number of runs in computing core-periphery structure becomes infinite, such nodes will be assigned the same aggregate core score.

We illustrate the Karate Club example by plotting the core quality R (7.10) as a function of α and β (see Figure 7.7). The landscape of top core nodes can be complicated, especially as one considers larger networks, but examining it in a small network like the Zachary Karate Club is convenient for illustrating both how our method works and how it exposes multiple possible core-periphery structures in one network.

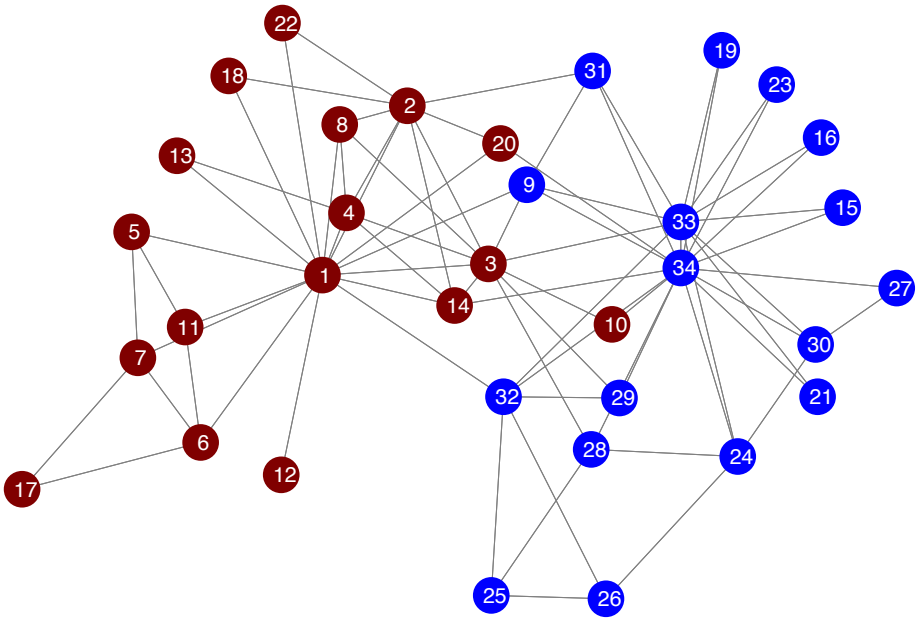


Figure 7.5: The Zachary Karate Club network [287], which we visualize using the implementation of the Kamada-Kawai algorithm [154] in Ref. [265]. The colors represent the two groups into which the club split while it was under study.

Table 7.1: Nodes in the Zachary Karate Club network nodes along with their aggregate core scores (7.11) computed using the core quality (7.10) and the transition function (7.9). We also give the node degrees

Node	Core Score	Degree	Node	Core Score	Degree
1	1.0000	16	19	.2255	2
34	.9951	17	15	.2254	2
3	.9702	10	21	.2254	2
33	.8719	12	23	.2244	2
2	.8577	9	16	.2244	2
9	.7755	5	26	.2196	3
14	.7546	5	25	.2038	3
4	.7537	6	7	.1840	4
8	.6441	4	6	.1840	4
31	.5849	4	18	.1787	2
32	.5377	6	22	.1785	2
24	.4661	5	11	.1580	3
20	.4499	3	5	.1579	3
30	.4152	4	13	.1425	2
28	.3957	4	27	.1050	2
29	.3784	3	12	.0477	1
10	.2506	2	17	.0343	2

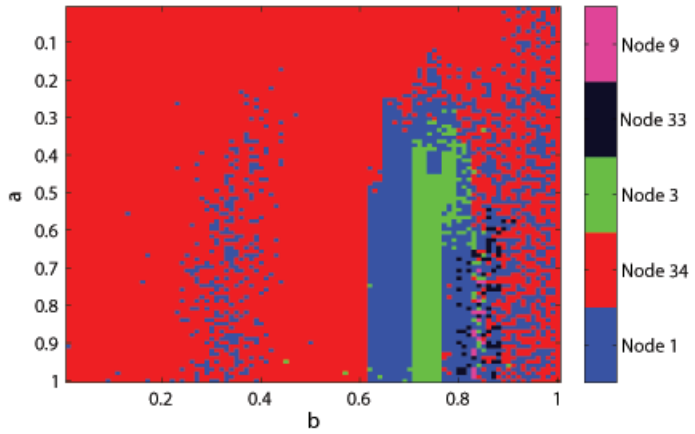


Figure 7.6: The node of the Zachary Karate Club that has the top core score (i.e., $\arg\{\max_k(C_k)\}$, where $k \in \{1,\ldots,34\}$ indexes the nodes) as a function of α and β. We computed core scores using the core quality (7.10) and the transition function (7.9).

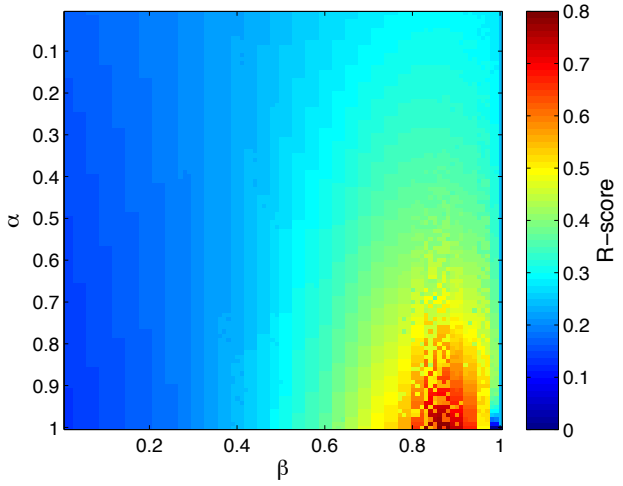


Figure 7.7: Core quality R (7.10) of nodes in the Zachary Karate Club as a function of the parameters α and β. We used the transition function (7.9).

7.6.2 The London Underground

One expects many metropolitan (metro) and subway transportation networks to exhibit a core-periphery structure [239]. To illustrate this, we compute core scores for the London Underground ('Tube') transportation network, which exhibits a strong core-periphery structure and a weak community structure. We collected the data for this example using the website for the London Underground (http://www.tfl.gov.uk). The Tube network that

we assembled has 317 nodes (one for each station) and weighted edges that represent the number of direct, contiguous connections between two stations. For example, Baker Street and Edgware Road share an edge of weight 2, as they are adjacent stations on both the Circle Line and the Hammersmith & City Line. They are also connected by the Bakerloo Line; however, they are not adjacent stations on that line, so it does not affect the weight of the edge between them.

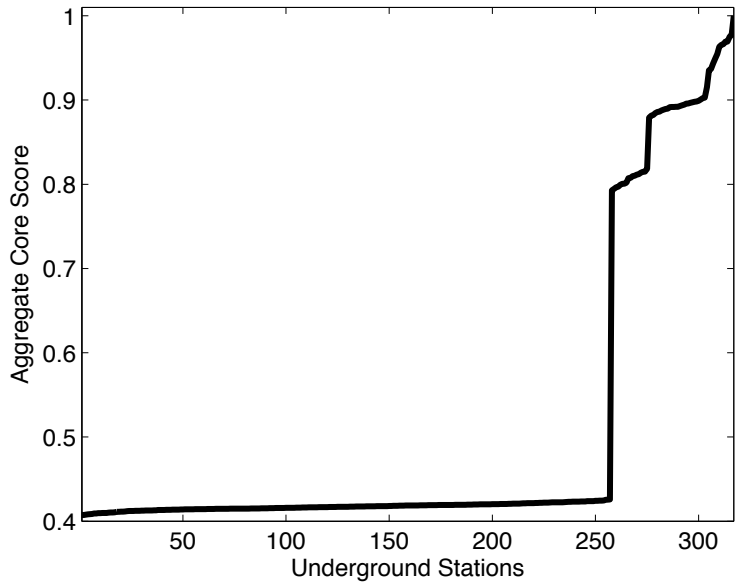
We partitioned the network into communities algorithmically by optimizing the modularity quality function [108, 211, 229] using the Louvain [30] computational heuristic (see Section 5.6.1). This splits the network into 21 communities, and the largest community that we obtained contains 19 nodes[5]. Most of these communities consist of groups of stations on a single line.

In Table 7.2, we show the results that we obtained for the London Tube network by computing aggregate core scores (7.11) using the core quality (7.10) and the transition function (7.9). We list the top ten stations and their corresponding aggregate core scores.

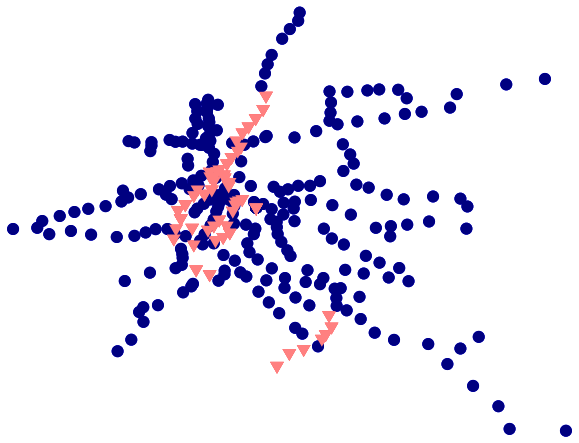
Station	**Core Score**
King's Cross St. Pancras	1.0000
Farringdon	0.9773
Barbican	0.9751
Paddington	0.9693
Great Portland Street	0.9692
Moorgate	0.9663
Embankment	0.9653
Euston Square	0.9632
Edgware Road	0.9546
Baker Street	0.9490

Table 7.2: The ten most core-like nodes in the London Underground network along with their aggregate core scores (7.11), which we obtained using the core quality (7.10) and the transition function (7.9).

[5]The Louvain method is stochastic, so one can get slightly different network partitions in different runs of the algorithm. We simply wanted a reasonable community structure as a means of comparison, so we used a single run of the algorithm in each situation for which we compute community structure. See also Section 5.6.

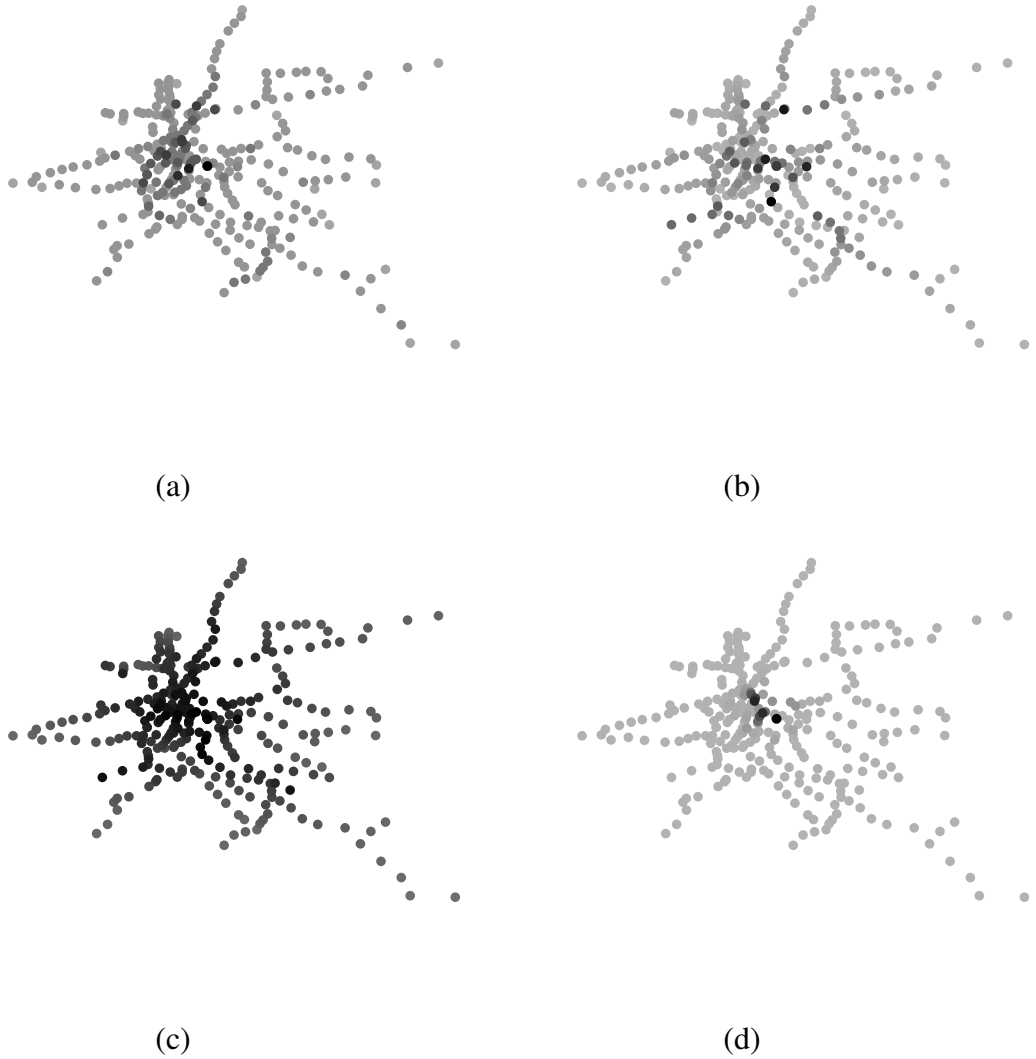


(a)



(b)

Figure 7.8: (a) The ordered list of aggregate core scores (7.11) for the London Underground stations suggests that there are 60 important stations. [We use the core quality (7.10) and the transition function (7.9).] (b) We plot the stations using their geographical locations. The ▼ symbol designates the 60 most important stations, and the • symbol designates the 257 other stations.



(a) (b)

(c) (d)

Figure 7.9: We plot the stations using their geographical locations, and shade them according to different centrality measures: (a) degree, (b) betweenness, (c) closeness, and (d) PageRank. See also Section 1 in the Appendix.

In Figure 7.8(a), we plot the aggregate core scores for the stations in order of ascending values. This reveals a sharp jump in aggregate core score and thereby suggests that the London Tube has a core group of about 60 stations, and periphery of 257 stations. (When ordered by degree, stations 240 through 287 have equal degree.) In Figure 7.8(b), we plot the stations using their geographical locations. The ▼ symbol designates the 60 most important stations, and the ● symbol designates the 257 other stations. In this example, we see that it is reasonable to construe the network as dichotomized into (about) 60 core nodes and (about) 257 peripheral nodes. The large set of ▼ nodes in the middle constitute the stations in Central London (e.g., King's Cross/St. Pancras and Paddington, which are both associated with major train stations). The ▼ nodes that are farther towards the bottom right constitute the stations near Waterloo, which is another major train station in London. A possible explanation for the split core is that the two clusters of core stations are separated geographically by the river Thames, which runs through central London. Most of the historical landmarks are found north of the Thames, such as Buckingham Palace, Trafalgar Square, and the Tower of London. The so-called South Bank (centred around Waterloo) is a 60s arts hub containing the Royal Festival Hall, the National Theatre, and the London Eye. In Figure 7.9 we have plotted the same image for other (continuous) centrality measures for comparison. We note that they do not show the same split.

7.6.3 Networks of network scientists

We now consider networks of co-authorships between scholars who study network science. We study two such networks—one from 2006 [209] and another from 2010 [90]. These networks (which both concentrate on papers written by physicists) have 379 and 552 nodes, respectively, in their largest connected components. The nodes correspond to scholars working in the field of network science, and an edge between two of them has a weight based on the number of papers that they have co-authored. (Note that the 2006 network is not a subset of the 2010 network.)

In Table 5 of the Appendix, we show the names of the scholars from both 2006 and 2010 with the top thirty aggregate core scores (7.11) using the core quality (7.10) and the transition function (7.9). In Table 6 in the Appendix, we give the top 30 aggregate core scores for the 2010 network using three variant computations: (left) using the one-parameter transition function (7.14) with the product form (7.8) for the core-matrix elements, (middle) using the smooth transition function (7.13) with the product form (7.8) and (right) using the usual transition function (7.9) with the p-norm (7.12) with $p = 2$ for the core-matrix elements. The ordering of the top 30 scholars is similar across different variations of the methodology, although there are some differences.

The networks of network scientists have both a sensible community structure and a sensible core-periphery structure [recall the block model in Figure 7.1(c) and (d)]. We illustrate this point in our visualization of the network in Figure 7.10. Each pie chart represents a community, which we computed by optimizing modularity using the Louvain algorithm [30]. Each pie is composed of the nodes in a single community, and each node is represented by a segment colored according to its aggregate core score (7.11) computed using the core quality (7.10) and the transition function (7.9). One can plainly see that the network's core nodes are distributed throughout the various communities and that many communities have both core and peripheral nodes.

We calculated community structures in which the 2006 network is split into 19 communities and the 2010 network is split into 25 communities, although different community-detection methods yield somewhat different partitions of the networks [132]. For example, one previous examination [236] of community structure in the 2006 network of network scientists using a spectral tripartioning method identified three large groups: one in which A.-L. Barabási is the key node (in the sense of having the largest 'community centrality' [209] in the group), one in which M. E. J. Newman is the key node, and one in which A. Vespignani and R. Pastor-Satorras are the two key nodes. As shown in Table 5 in the Appendix, all four of these nodes have very high aggregate core scores.

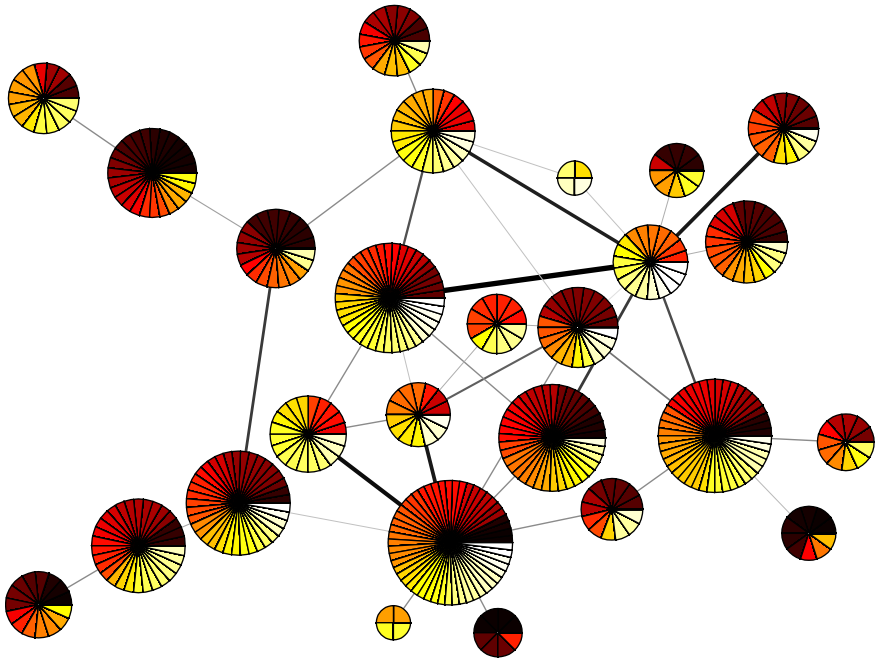


Figure 7.10: Visualization of the 2010 network of network scientists. Each pie represents a community, and the colors represent the rank order of the nodes' aggregate core scores (7.11), which we computed using the core quality (7.10) and the transition function (7.9). Darker colors indicate higher rankings; the colors are spaced evenly over all (aggregate) core scores and contain no information about the score distribution. Each wedge represents a single node, and larger pies contain more nodes. The darkness of the edges represents the total strength of connections between communities. We produced this visualization using code described in Ref. [265] that uses the Kamada-Kawaii algorithm [154] to locate the centers of the pies. We then tweaked the center locations by hand.

Individual communities in both the 2006 and 2010 networks exhibit a core-periphery structure. As indicated above, the core nodes are distributed throughout the communities. In the 2006 network, 12 of the 19 communities contain at least one node among those with the top 30 aggregate core scores in Table 5. In the 2010 network, 9 of the 25 communities contain at least one node in the top 30 from Table 5. Additionally, each of the communities in the two networks includes one or two highly connected (i.e., high-strength) nodes and several other nodes with low strengths. In the 2006 network, the mean strength is 4.8, and 17 of the 19 communities contain a node with a strength of at least 9. (There are 43 such nodes in the entire network.) In the 2010 network, the mean strength is 4.7, and 20 of the 25 communities contain a node with a strength of at least 10. (There are 50 such nodes in the entire network.) This network is an example that contains both an identifiable community structure and an identifiable core-periphery structure. However, methods to detect core-periphery structure need not indicate anything about community structure and vice-versa. As we discussed previously, community structure and core-periphery structure provide different lenses with which to view a network [286]. There can be examples in which a core and a periphery are describable as separate communities, but community structure and core-periphery structure are different concepts.

In Figure 7.11, we zoom in on the largest community (53 nodes) in the 2010 network of network scientists. This community includes the node (A.-L. Barabási) with the highest aggregate core score. This figure illustrates that nodes with high scores indeed occupy a well-connected position inside their community as well as in the entire network.

7.6.4 Voting-similarity network of the United States Senate

Finally, we consider similarity networks constructed using roll-call votes from the United States Congress. One can build such a network from a single 2-year Congress of either the Senate or the House of Representatives [227, 228, 273]. For each House and Senate, one constructs a complete (or almost complete) weighted network in which each node

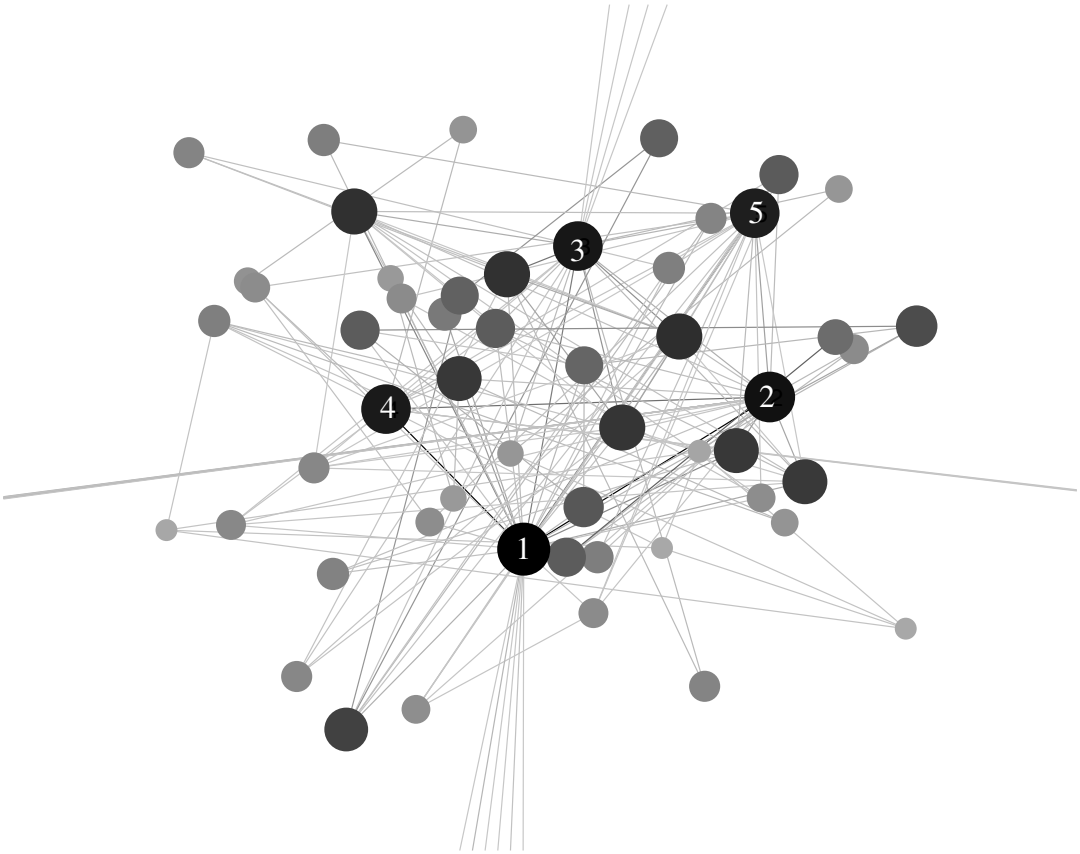


Figure 7.11: Magnification of the largest community in the 2010 network of network scientists. The darkness of the edges corresponds to the strength of the edges, and the size and darkness of the nodes represent the aggregate core score. (Edges that leave the picture are connected to nodes in other communities.) The five labeled nodes, and their corresponding core scores, are A.-L. Barabási (1), H. Jeong (.9181), T. Vicsek (.8856), R. Albert (.8737), and Z. N. Oltvai (.8550).

represents a legislator and a weighted edge between two legislators indicates the similarity of their voting patterns. In our calculation, each adjacency-matrix element A_{ij} is equal to the number of times that legislator i and j voted in the same way divided by the total number of bills on which both i and j cast a vote. This type of network is called a 'similarity network', because the weights of the edges give a measure of similarity between the nodes to which they are incident. (As was recently discussed in the context of resolutions in the United Nations General Assembly [184], one can also construct networks from voting data in several other ways.)

As an example, we consider the similarity network for the 108$^{\text{th}}$ Senate, which occurred during the third and fourth year of George W. Bush's presidency (2003–2005). Methods of community detection do well at separating the two main parties – Democrats and Republicans – for modern congresses (after 1979), indicating that voting coalitions align with party divisions [273]. In Table 7 of the Appendix, we give for each Senator the aggregate core score (7.11) computed using the core quality (7.10) and the transition function (7.9). In Figure 7.12, we show scatter plots between the strength centrality and various other centrality measures for the 108$^{\text{th}}$ Senate network. We color Republicans in red and Democrats in blue. The strong similarity between the MINRES and the PageRank computation is due to the type of network and the scores being relatively close together (see the definition os MINRES in 7.3.2). They need not be as similar in other networks.

Some of the centrality measures in Figure 7.12 have been used previously to study Senators and Representatives in legislation cosponsorship networks [112, 113], which have in turn been compared to modularity-based measures of political partisanship studied using roll-call voting networks [289]. As one can see from Figure 7.12, the different centrality measures do indeed measure different things. Observe in particular that none of the centrality measures by themselves separate the communities very well, whereas a combination of two of them can sometimes distinguish a community of (mostly) Republicans and a community of (mostly) Democrats. Investigation of core-periphery structure using aggre-

gate core scores thus complements examination of community structure by allowing one to examine a different type of meso-scale structure. As panel (c) illustrates, it also nicely complements existing centrality measures.

7.7 Summary

Quantitative computations of core-periphery structure have only started to receive attention from network scientists relatively recently (compared to community structure). Our method contributes to this ongoing exploration with new family of methods to investigate core-periphery structure in networks. We generalized ideas from Borgatti and Everett [50] and designed an approach that assigns values (*i.e.*, core scores) to nodes along a continuous spectrum between nodes that lie most deeply in a network core or at the far reaches of a network periphery. Our approach can be used with a wide variety of different functions to transition between core and peripheral nodes, and it also allows one to use different ways to measure core quality. The importance of such flexibility, combined with the ability to use our method either to produce a centrality measure for coreness or discrete divisions of core and periphery nodes, has long been recognized by sociologists as an important aspect of core-periphery structure [251].

As we have illustrated, networks can contain community structure, core-periphery structure, both, or neither. For example, the 2006 and 2010 networks of network scientists exhibit both types of meso-scale structures in a meaningful way. In these networks, investigating core-periphery structure reveals a global "infrastructure" that remains invisible if one searches only for community structure.

7.8 Choice of simulated annealing

The decision to use simulated annealing was motivated by its simplicity and robustness (when given enough time). The robustness was further improved by the fact that we ran the

algorithm 10,000 times in most cases, and studied the weighted average of those outcomes. Our method provides new intuitions rather than a fast practical method, and we only tested it on relatively small networks. Therefore, running times were not an issue. If this method were to be developed further, the algorithm needs major improvement in terms of efficiency (see also Section 9.4). In Section 8.9, we exploit the probabilistic nature of the simulated annealing algorithm by considering multiple of its possible solutions to gain insight into the structure of brain networks.

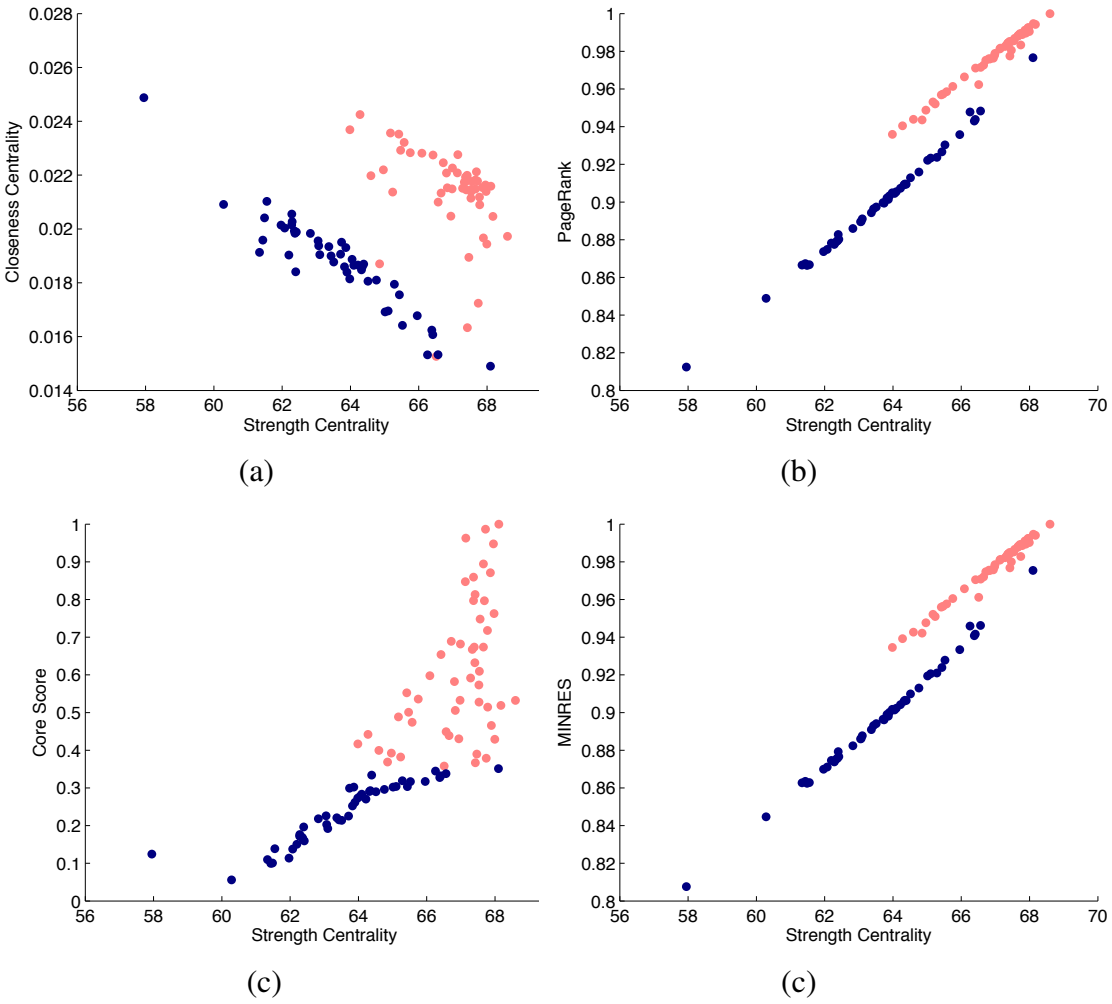


Figure 7.12: Scatter plots between strength and various other centrality measures for the 108th Senate voting-similarity network. We show Republicans in red and Democrats in blue. In panel (c), we computed aggregate core scores (7.11) using the core quality (7.10) and the transition function (7.9).

Chapter 8

Brain Networks

This chapter represents work that was carried out in a collaboration between D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, and S. T. Grafton, and me. Sections 8.2 and 8.3 provide some background to the methods used in brain mapping. Section 8.4 describes the set-up of the experiment carried out by N. F. Wymbs and S. T. Grafton at the Department of Psychological and Brain Sciences and UCSB Brain Imaging Center at University of California, Santa Barbara. They, together with D. S. Bassett, were also the ones who carried out the wavelet analysis of the fMRI data from which the functional brain networks were constructed. Section 8.5 describes the computation of dynamic community structure, which was carried out by D. S. Bassett, M. A. Porter, and P. J. Mucha. Section 8.6 describes the static core-periphery analysis, which was carried out by me. Section 8.7 describes results that were obtained by comparing the outcomes of these two methods to each other and to the accompanying information on skill learning for each participant. The results from this project have been written up in a journal article which will appear in *PLoS Computational Biology* [20]. Section 8.9 is further work that I did independently using the data from the aforementioned imaging experiment. Figures that were *not* created by me are Figures 8.1, 8.2, 8.3, 8.4, and 8.9; I created all other figures.

8.1 Introduction

Modelling the human brain as a network is becoming an increasingly popular and useful way of studying it [62]. There are many ways of representing the brain and its functionality as a set of nodes and edges. One can construct *structural brain networks* based on the anatomical links between brain regions or synapses. One can also construct *functional brain networks* based on the similarity in activity levels between different parts of the brain, either during rest or during a mental exercise. These functional brain networks are what we will be discussing in this chapter. To construct them, one can use data from magnetic resonance imaging (MRI) [1], a method that uses nuclear magnetic resonance to visualize nuclei of atoms inside the brain, functional magnetic resonance imaging (fMRI) [20, 144, 186], a type of MRI that measures activity by detecting changes in blood flow to the different regions of the brain, or electroencephalography (EEG) [259], a method that measures electrical activity along the scalp. Different types of brain networks, both at a global and at a cellular scale, have shown network characteristics such as degree heterogeneity and community structure [59, 62].

It was shown in [186] that temporal changes in the modular structure of brain dynamics play an important role in learning new skills. Recently, the use of the discrete wavelet transform has led to many new insights into simultaneous characterisation of dynamic interactions across the whole human brain, that have been used to, for example, establish links between functional brain network configuration and intelligence [61]. In the study that we will present in Sections 8.5, 8.6, and 8.7 we focused on skill learning over long time scales (minutes to hours), and we found that learning success is correlated with temporal changes in the community structure of the brain network, which in turn can be described by the core-periphery organisation of the static (aggregated) network.

The rest of this chapter is organised as follows. In Section 8.2 we introduce the method of constructing functional brain networks from fMRI data. The fMRI data that we analysed was obtained from participants as they went through a process of learning a simple motor

skill. In Section 8.3 we describe a way of quantifying how quickly a person learns such a skill. In Section 8.4 we describe the experimental set-up in more detail. In Section 8.5 we present the results obtained from dynamic community structure analysis on the brain networks as they change with time. In Section 8.6 we describe the core-periphery structure of the static networks obtained by averaging over the time-varying networks. In Section 8.7 we compare the outcomes of the measures in Sections 8.5 and 8.6 to the *learning parameter* that we introduced in Section 8.3 and measured for each participant in our experiment. In Section 8.9 we describe a new way of finding communities in functional brain networks based on the ways that brain regions participate in the different cores from Section 8.6 together.

8.2 Constructing brain networks

In order to construct similarity networks that have brain regions as their nodes, neuroscientists parcellate the human brain into standardised regions. One such standardised parcellation is the Harvard-Oxford atlas [252, 284], which subdivides the brain into 112 cortical and sub-cortical regions. Section 4 in the Appendix contains a list of the brain regions identified by this atlas. The mean level of activity will then create a time series associated with each of these brain regions.

8.2.1 Stationary wavelet transform for constructing human brain networks

The research presented in this chapter is based on the analysis of data obtained through fMRI brain scans. Oxygen-rich blood and oxygen-poor blood have different magnetic resonances, which fMRI uses to measure blood flow towards different areas of the brain, indicating their activity levels over time. Different functions in the brain are associated with different frequencies of activity levels, so intuitively it makes sense to analyse the fMRI

time series data by looking at the frequency spectrum. In the brain, one would also expect oscillations caused by a part of the brain performing a task to occur briefly and then decay again, rather than extending infinitely with a certain frequency and phase (such as the sine and cosine waves used in Fourier analysis) [60]. Other intuitive advantages of wavelet transforms for an fMRI time series analysis include its whitening or decorrelating properties, which is convenient statistically, its denoising and compressing properties (the signal to noise ratio is usually about 1–2 in fMRI data), and the fact that it is fast to compute ($O(n)$ [185], compared to $O(n \log n)$ for the fast Fourier transform).

As with Fourier series, a time series function can be decomposed into an orthonormal basis of wavelets. The wavelets are characterized by their scale and their location (in time), whereas the waves in an ordinary Fourier Transform are only characterized by their scale.

A more general transform than the orthonormal one is the so-called *redundant wavelet transform*, or *stationary wavelet transform*, or *maximum-overlap wavelet transform*. Adding redundancy has been shown to provide a greater robustness to noise [111]. The latter may seem counter-intuitive to the non-stationary behaviour of the brain mentioned previously, but in the study in this section we are interested in connectivity and similarities between the behaviour of different parts of the brain, for which a stationary method has been proven useful and used extensively in the past [52, 60, 61].

8.2.2 Comparing pairs of time series

For the purpose of these experiments, we are interested in high-frequency signals which are thought to be related to cooperative temporal dynamics in the brain during a task. The wavelet frequency band used is one the scale of 0.06–0.125 Hz. We extracted the wavelet coefficients for several blocks of the EXT, MOD, and MIN sequences for each of the 20 participants and for each of the 4 scanning sessions. For each of these block data sets, we constructed an 112×112 (where 112 is the number of separate brain regions representing the nodes of the brain network) adjacency matrix W, where the entries are the pairwise

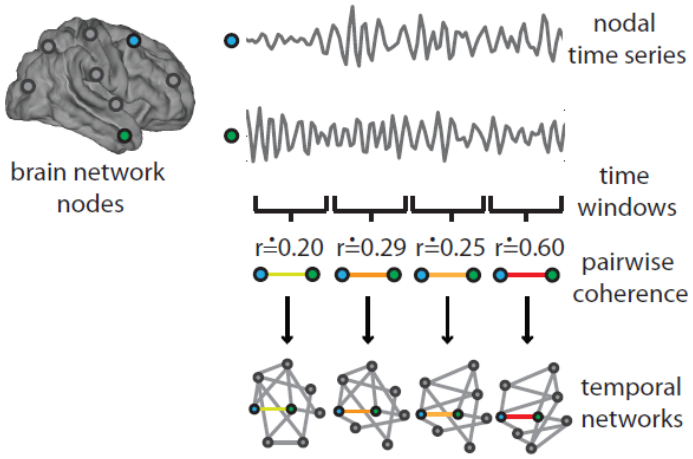


Figure 8.1: A schematic figure for obtaining a time-dependent brain network from fMRI time series data. The brain is parcellated into 112 regions, each of which exhibits a time-dependent level of activity. For any pair of regions and block of time (the time it takes to perform one trial), the coherence is measured as a similarity in wavelet coefficients of the two time series. From there, we obtain a network that changes over time. This figure was created by D. S. Bassett and appears in [20].

functional connections present in the brain during that window (the block of time it took to complete one particular trial) in a given participant and for a given scan. The weight of the edges between regions i and j is the magnitude squared spectral coherence as a measure of nonlinear functional association between any two wavelet coefficient time series. The wavelet analysis was performed in the same way as was done in [18]. In order to create a time-dependent (multilayer) network, 6–10 adjacency matrices for each sequence type (EXT, MOD, or MIN), participant and scanning session were pasted together in chronological order with edges added between them (to obtain a multilayer network, as described in Section 5.7). This will be described in more detail in Section 8.5. The method of obtaining multilayer networks from the fMRI time series data is illustrated in Figure 8.1.

8.3 Measuring learning

The movement time (MT) is defined as the difference in time between the first button press and the last button press for any one sequence. The learning rate of a person is approximated by fitting an exponential function, plus a constant to the data consisting of the times it took to finish each sequence as a function of how often the sequence has been practised. The MT is approximated by

$$MT(t) = D_1 e^{t/\kappa} + D_2,$$

where D_2 is the estimated fastest time attainable and $D_1 + D_2$ is the estimated starting time for a participant and sequence. The data can be fitted without prior knowledge of D_1, D_2, k and is a standard procedure in motor control and learning studies [243]. This data was fitted in MATLAB using a robust outlier correction in order to obtain, for each participant and each sequence, the exponential drop-off parameter κ, which we call the "learning parameter".

8.4 Experimental set-up

The group of volunteers consisted of 22 right-handed participants, of which 9 were female and 13 were male, and the mean age was about 24. None of the participants regularly practised any musical instruments. Two participants were later excluded due to incomplete data. All of the participants completed 30 training sessions, 3 fMRI scans during training and one pre-training fMRI scan. A scanning session was conducted after 10 training sessions on 10 days over a 2-week period. The training was completed on a personal laptop. The participants were presented with a sequence of stimuli, consisting of a row of squares which were highlighted in turn. They then had to respond by pressing a corresponding key under one of the five fingers of their right hand. The squares were aligned from left to right, with the left-most corresponding to the thumb and the right-most corresponding to the little

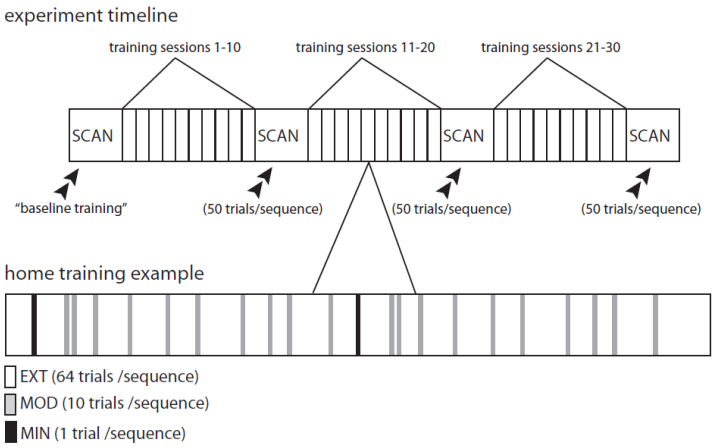


Figure 8.2: This figure illustrates the sequence practice timeline. Each participant completed a 50 trial sequence during every scanning session, and 75 trial sequences during each home practice session. This figure was created by D. S. Bassett appears in [20].

finger. There were 6 sequences in total, each containing 10 elements. Before the start of each sequence, the participant was presented with an identity cue. Each session contained two extensively trained sequences (EXT), two moderately trained sequences (MOD) and two minimally trained sequences (MIN), that were each repeated for 64, 10, and 1 trials, respectively. Hence, each practice session contained 75 trials in total, in a uniformly randomised order. The score was reported as the number of error-free sequences after each block of 10 trials and the mean time to complete an error-free sequence. After completing 10 training sessions the participants were tested in the lab on all three types of sequences (EXT, MOD and MIN) in an fMRI scanner. Figure 8.2 illustrates the timeline of the experiment. Much more detail on the experimental set-up and procedure can be found in the supplementary material of [20].

8.5 Dynamic communities

We use the multilayer modularity proposed in [204] to examine the dynamic behaviour of the brain networks created in the previous section. We describe those using the generalized

modularity function

$$Q = \frac{1}{2\mu_N} \sum_{i,j,l,r} \left\{ \left(\mathbf{W}_l(i,j) - \gamma_l \frac{k_l(i)k_l(j)}{2m_l} \right) \delta_{l,r} + \delta_{i,j} t_{l,r}(j) \right\} \delta_{g_l(i),g_r(j)}, \tag{8.1}$$

where $\mathbf{W}_l(i,j)$ is a weighted edge between nodes i and j in layer l, the parameter γ_l is the structural resolution parameter of layer l, $c_l(i)$ is the community assignment of node i in layer l, $t_{l,r}(j)$ is the coupling strength, *i.e.* the strength of he connection between node j in layer l and node j in layer r. We also have

$$k_{i,l} = \sum_j \mathbf{W}_l(i,j), \qquad g_l(j) = \sum_r t_{l,r}(j),$$
$$\kappa_l(j) = k_l(j) + g_l(j), \qquad \mu_N = \frac{1}{2} \sum_{j,r} \kappa_r(j),$$
$$m_l = \frac{1}{2} \sum_{i,j} \mathbf{W}_l(i,j).$$

See also Section 5.7 for more details on modularity in multilayer networks. In this case we let $t_{l,r}(j) = t$ for adjacent slices (*i.e.* $|l - r| = 1$) and 0 otherwise. We also let $\gamma_l = \gamma$ for all layers. Let $\omega = \gamma = 1$ [20]. Optimizing the modularity using the networks from Section 8.2 and the parameters chosen above, yields a partition of the brain network into communities for each layer.[1] We are interested in the way in which individual nodes move between communities (if they move at all). Define the *flexibility* $f(i)$ of node i as the number of times that a node changes community assignment in a given multilayer network [19, 204], divided by the number of possible changes, i.e. one less than the number of slices in that network (remember that the number of slices in the networks differ). The flexibility of an entire multislice network is defined as the mean flexibility of the nodes: $F_N = \frac{1}{n} \sum_{i=1}^{n} f_i$. We found a large range of flexibilities for different brain regions.

To decide whether a person's brain was more or less flexible than expected, we compared them to a null model. The null model was constructed by rewiring the inter-layer

[1] We performed the dynamic community detection procedure using freely available MATLAB code [153] that optimises multilayer modularity using a Louvain locally greedy algorithm [30].

edges uniformly at random (in a way that keeps a perfect matching between two consecutive layers; see also Section 2.7) and examining the expected strength distribution. This is similar to a configuration model on each of the set of edges between two consecutive layers, except that it preserves the edge property of spanning two layers. In the null model, the number of community swaps is measured by examining the number of community changes as one moves along the nodes on a path created by the inter-layer edges, starting at the first layer and ending at the last chronological layer in the network. (There are 112 such paths as the set of inter-layer edges between layer l and $l + 1$ is a matching and remains a matching after rewiring.)

We found the distribution of expected mean nodal flexibility values by averaging over 100 instances of a randomly rewired network and all of the 20 participants. We used a Louvain-like method [30, 153] to optimise the modularity for the original networks. This method is not deterministic, so we ran the optimizations 100 times for each multislice network and averaged those values.

An earlier study on this type of brain flexibility and learning performed by the same group [18] demonstrated that the amount of flexibility in the brain of an individual during a training session could be used to predict the speed of learning in the following training session. The experiments reported in [18] used a similar set of skills that the participants had to learn, but the training sessions took place within a shorter time scale (3 sessions in 5 days) and all training sessions took place inside an MRI scanner. This study [20], as well as the previous one [18], showed that community structure changes with the number of trials performed for all participants and scanning sessions. In Figure 8.3, we show the modularity function Q, the number of communities, and mean flexibility as a function of the number of trials practised. The modularity increases at first (up to 200 trials practised), and then starts decreasing with the number of trials practised, suggesting that community structure becomes less pronounced with learning. Both the number of communities and the flexibility increase with the number of trials practised. The mean flexibility of the brain

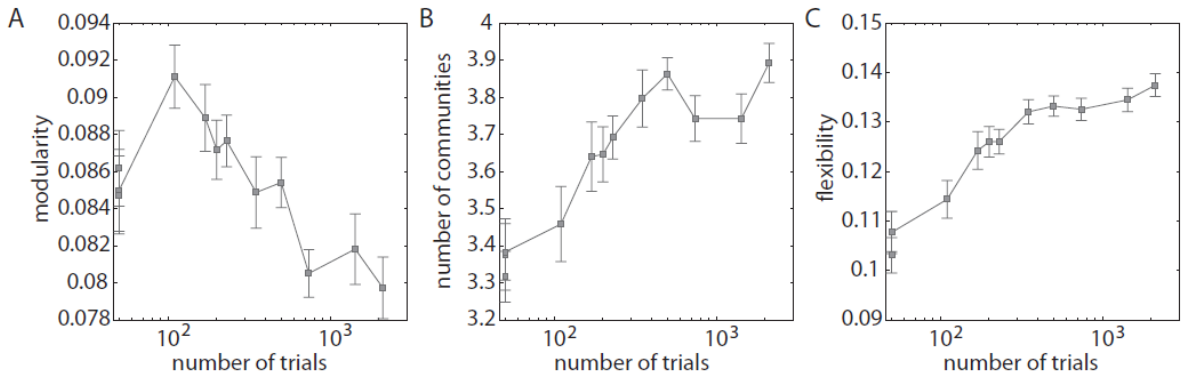


Figure 8.3: (A) Modularity, (B) number of communities, (C) mean flexibility calculated as a function of the number of trials. The flexibility is averaged over all 112 brain regions and the error bars indicate the standard error of the mean over the 20 participants. This figure was created by D. S. Bassett and appears in [20].

regions varied between approximately .04 and .14, and is clearly non-Gaussian (see Figure 8.4, left panel). There is a heavy tail of regions with low flexibility.

From the results of the flexibility computations we defined a so-called *temporal core, periphery* and *bulk*. The core consists of the most stiff regions (i.e. the ones with that showed the least community swaps), the periphery consists of the most flexible regions, and the bulk consisting of the regions with a close to average (expected) flexibility. We defined the temporal core, periphery and bulk as follows. A region falls in the temporal core if its flexibility is below the 2.5% confidence bound of the null model distribution, in the temporal periphery if its flexibility is above the 97.5% confidence bound, and in the temporal bulk if its flexibility is between those bounds. The right panel in Figure 8.4 shows the anatomical locations inside the brain of the regions that we classified as core, bulk and periphery. The core contains primary sensorimotor and visual processing areas, the periphery contains multimodal association areas, and the bulk contains the remainder of the brain (and is therefore composed predominantly of frontal and temporal cortex) [20]. The sensorimotor area of the brain is responsible for combining information from visual and tactile senses and is needed for, *e.g.*, hand-eye coordination. The multimodal association areas of the brain are responsible for processing sensori stimuli. Rapidly executing

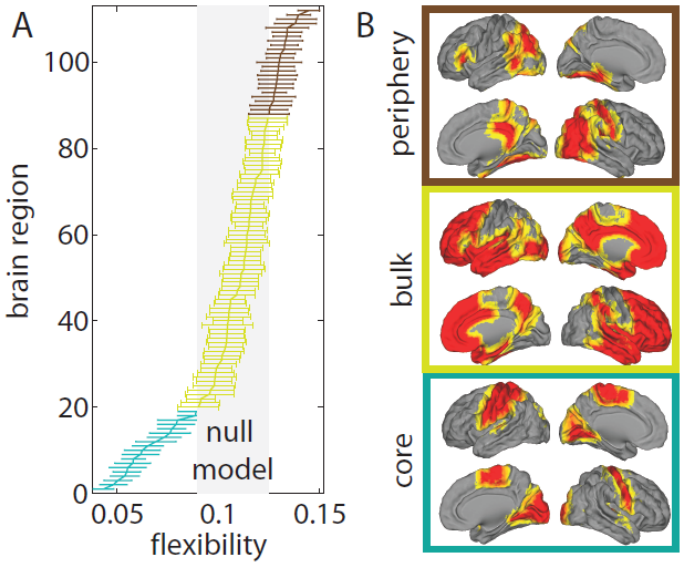


Figure 8.4: (A) mean flexibility for different brain regions. The error bars indicate the standard error of the mean over the 20 participants. The colors indicate classification of the regions into the temporal core (blue), bulk (yellow), and periphery (red). (B) anatomical distribution inside the brain of regions in the core (red), bulk (yellow) and periphery (grey). This figure was created by D. S. Bassett and appears in [20].

complex motor sequences requires extensive practice. These are known to be generated by the sensorimotor and visual processing areas (which we classified as part of the temporal core) [187].

When first learning a skill, the brain can choose between a variety of cognitive strategies involving the sensorimotor and visual processing areas as well as other areas of the brain. Therefore, we hypothesized that participants with a clearly distinguishable temporal core-periphery organisation would learn better than those without one.

The hypothesis was tested by calculating the Spearman rank correlation coefficient (which assesses how well the relationship between two variables can be described using a monotonic function) between the skewness and the kurtosis of the flexibility distribution of the first scanning session (before the participants had any practice) and the learning parameter κ (see Section 8.3) observed subsequently. The skewness, denoted α_3, is the third

central moment of a distribution, defined as

$$\alpha_3 = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right], \tag{8.2}$$

where $\mu = \mathbb{E}(X)$ and $\sigma = \sqrt{(}\mathbb{E}((X-\mu)^2))$, are the mean and standard deviations of X. Arguably, in this setting, because the skewness measures how much a distribution 'leans' to one side of the mean, the skewness can be used to measure the presence of a temporal core and periphery. The kurtosis, denoted α_4, is the fourth central moment of a distribution, defined as

$$\alpha_4 = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right]. \tag{8.3}$$

In this setting, because the kurtosis in a sense measures how prone a distribution is to outliers, one can argue that the kurtosis can be used to measure the separation between the temporal core and the temporal periphery. The results showed both the skewness and the kurtosis are negatively correlated with the learning parameter. For the skewness, the Spearman rank correlation is $\rho \approx -.48$, and the p-value is $p \approx 0.033$. For the kurtosis, the Spearman rank correlation is $\rho \approx -.49$, and the p-value is $p \approx 0.027$. Hence, participants with a smaller separation between the temporal core and periphery learn better than participants with a greater separation between the temporal core and periphery. Additionally, those with more skew of the flexibility over the brain regions learned better than those with less skew of the flexibility. The data suggests that, for fast learning, it helps to have a strong temporal core and periphery, but a smooth transition between the two.

8.6 Core-periphery structure of the static brain networks

In this section we consider the slices of the network as separate, static networks, and study their *geometric* core-periphery structure, i.e. the core-periphery structure as discussed in Section 7.4 that is defined on networks without time-dependency. We used the functions

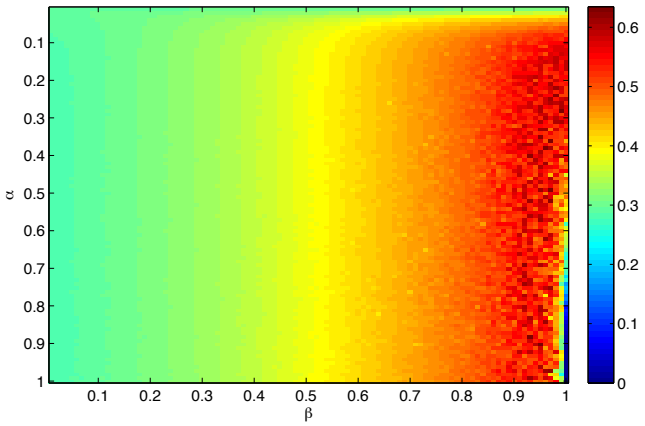
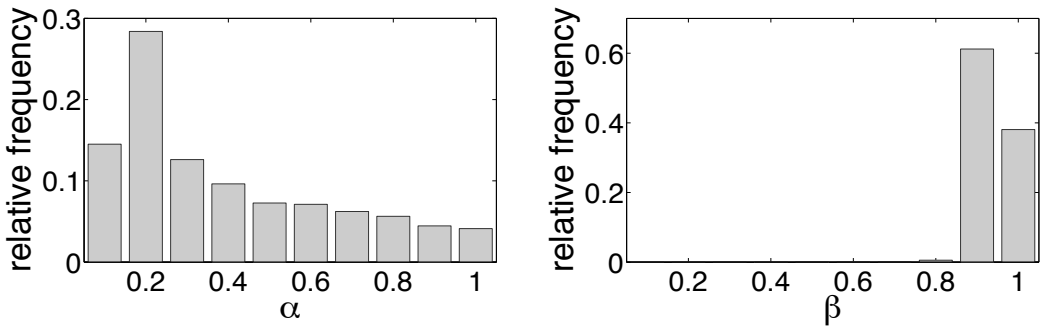


Figure 8.5: Core quality R (7.7) in the (α,β) parameter plane for a typical participant

(7.13) and (7.8) for initial calculations. The landscape of the R-scores (7.7) as a function of α and β turned out to look nearly identical for different individuals, giving the impression of a 'typical' core shape for these types of brain network. In Figure 8.5 we show the landscape of R-scores for a typical participant. This shows that the α parameter is far less important than the β parameter when it comes to the R-score. This was the case with other instances of real-world networks (see Sections 7.6.2, 7.6.3, and 7.6.4), but this example is extreme in that respect. In Figure 8.6, we show the distribution of the α and β parameters over all network layers, participants, scanning sessions, and sequence types. The distribution over α is far less concentrated than over β (standard deviations are .26 and .05, respectively), but that is not surprising given the typical landscape in Figure 8.5.

Figure 8.7 shows the ordered C vector from (7.13) with the values of α and β set to the mean values of those that maximize the R-score for all network layers, participants, scanning sessions, and sequence types. The mean values are $\alpha \approx .40$ and $\beta \approx .94$. This figure demonstrates that the typical (geometric) core-periphery organization in the networks under study is a mixture between a discrete core-periphery organization, in which every node is either in the core or in the periphery, and a continuous core-periphery organization, in which there is a continuous spectrum to describe how strongly nodes belong to a core.



(a) (b)

Figure 8.6: The distribution of the α and β values that maximize the core quality R. We compute this distribution over all network layers, participants, scanning sessions, and sequence types.

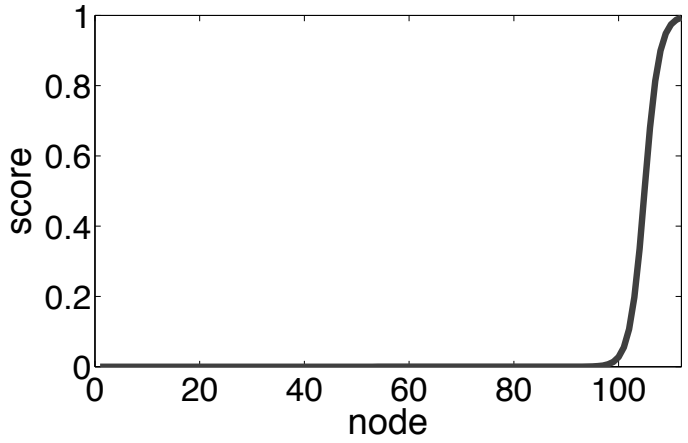


Figure 8.7: Mean core shape. Plot of the ordered C vector with the values of α and β set to the mean values of those that maximize the R-score for all network layers, participants, scanning sessions, and sequence types.

In these networks, the majority of nodes do not belong to the core, but those nodes that do (roughly 10% of the nodes) have a continuum level of association strengths with the core.

8.7 Core score and learning

We calculated the geometric core score in the previous sections on the static network 'layers'. We can, however, still consider its change over time (with learning). It turns out that regions that have a high geometric core score in the first scanning session and in blocks of the extensively practised sequences (EXT) were more likely to have a high geometric core score in later scanning sessions and in moderately and minimally practised blocks (MOD and MIN). Figure (8.8) shows the mean variance of the distribution of the geometric core scores over brain regions as a function of the number of trials completed after a scanning session. Error bars indicate the standard error of the mean over participants (where the data point from each participant is the mean geometric core score over brain regions, scanning sessions, sequence types, and network layers). The substantial decrease in variance suggests that, on average, the separation between the core and the periphery becomes less pronounced with learning: that the transition becomes less pronounced. This observation is consistent with the earlier observation in Section 8.5 that participants who are fast learners start out with a smoother transition between the dynamic core and periphery in the initial scanning session (before learning has begun) than those participants who turn out to be slow learners. We will discuss the correlation between the temporal core and the geometric core in Section 8.8.

8.8 Comparing flexibility and core-periphery structure

At the end of Section 8.7, we briefly discussed a result that hints at a correlation between the temporal core described in Section 8.5 and the geometric core described in Section 8.6. This does indeed turn out to be the case. The temporal core-periphery structure (which is a

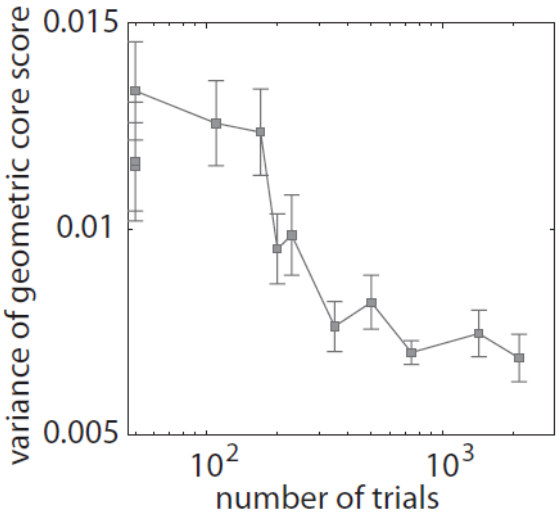


Figure 8.8: The mean variance of the distribution of the geometric core scores over brain regions as a function of the number of trials completed after a scanning session. Error bars indicate the standard error of the mean over participants (where the data point from each participant is the mean geometric core score over brain regions, scanning sessions, sequence types, and network layers). This figure was created by D. S. Bassett and appears in [20].

dynamic measurement) is strongly correlated with the geometric core-periphery structure (which is a measure of network structure). This indicates that regions with low flexibility tend to be strongly-connected core nodes in (static) network layers. In Figure 8.9, we show the relationship between temporal and geometric core-periphery structures for networks constructed from blocks of extensively, moderately, and minimally trained sequences on scanning session 1, 2, 3, and 4. The Pearson correlation between the temporal and the geometric core-periphery organization is very significant for the EXT ($r \approx -0.91$, $p \approx 3.3 \times 10^{-45}$), MOD ($r \approx -0.92$, $p \approx 2.2 \times 10^{-49}$), and MIN ($r \approx -0.93$, $p \approx 4.7 \times 10^{-50}$) data.

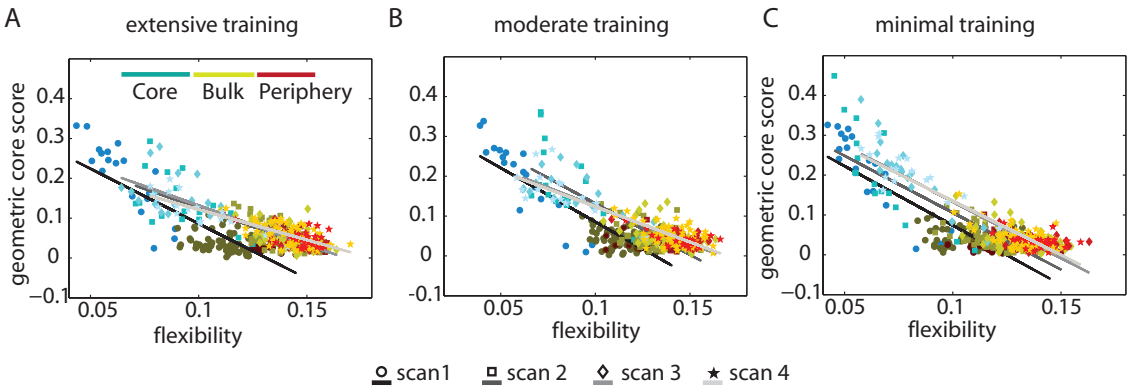


Figure 8.9: Relationship between temporal and geometric core-periphery structures for networks constructed from blocks of (A) extensively, (B) moderately, and (C) minimally trained sequences on scanning session 1 (circles), session 2 (squares), session 3 (diamonds), and session 4 (stars). We show temporal core nodes in cyan, temporal bulk nodes in gold, and temporal periphery nodes in maroon. Colour darkness of data points indicates scanning session with darker colors indicating scan 1 and lighter colors indicating scan 4. Lines with colors ranging from black to light gray indicate the best linear fits in grayscale for session 1 (black) through session 4 (light gray). This figure was created by D. S. Bassett and appears in [20].

8.9 Community structure from core-periphery structure (multiple cores)

The simulated-annealing approach used to compute the core-periphery scores in Section 8.6 is not deterministic. Multiple runs with the same parameters can yield different outcomes because the algorithm can get stuck in a local optimum. This stochasticity can provide additional information about the structure of brain networks during a task. We ran the simulated-annealing algorithm 100 times for each network of a participant, scanning session, and network layer using the 'ideal' core shape presented in Section 8.6 (there are 36–40 such network layers per participant). In this core, there are about 10 nodes in the core (see Figure 8.7). For this section, we only use the sequence type EXT (*i.e.*, the most frequently trained sequences).

For each set of those 100 runs, we create a graph where nodes represent runs and (unweighted) edges indicating that there is no overlap between the two cores (top 10 regions)

Table 8.1: Relative frequency of different numbers of possible cores. We average results over a total of 792 networks over all participants, scanning sessions and network layers.

# Cores	0	1	2	3	4
Relative frequency	.01	.57	.34	.06	.01

of a pair of runs. We then seek a clique of maximal size in these graphs, which indicates maximal sets of non-overlapping cores. We designate the cores from that clique the *class representatives*. We class the remaining runs into one of those classes if their cores have at least 6 nodes in common with the core of the class representative. We dismiss classes that contain fewer than 10 runs. See Table 8.1 for the distribution of the number of cliques in different networks.

When networks have multiple cores, about 80% of runs on average are included in the classes. The precision of the simulated-annealing algorithm[2] affects how many cores arise in the 100 runs, and the number of distinct cores decreases with increasing precision. At sufficiently low precision, nodes end up in the core almost uniformly at random. In this scenario no particular core is well-represented. At sufficiently high precision, all runs of one network contain the same core. Our results show that many of these networks clearly exhibit disjoint but well-represented cores. Hence, we observe multiple high-scoring cores, rather than fluctuations in the results that are merely due to the imprecise nature of the simulated-annealing algorithm.

We investigate how individual nodes anti-correlate with each other across different runs and even across different layers of a network. When a certain node has a high core score, another node might be more likely to have a low score. For each participant, we create a network that consists of the correlation coefficients (the Pearson product-moment correlation) between pairs of nodes. This indicates more than what the single edge between two nodes in one layer of a network indicates, as it takes into account the relationship of nodes

[2]For these runs, we use the code from [270], with the following parameters: 10000 maximum number of consecutive rejections, 200 Maximum number of successes within one temperature, 3000 maximum number of tries within one temperature, 10^{-9} temperature at which to stop.

to the rest of the network at different points in time. We construct graphs G_i for each participant $i \in \{1, \ldots, 20\}$ by recording these correlations between every pair of brain regions. From these graphs, we record only the edges that are very negative. In this case, we record those edges whose weight is less than $-.2$ (this is about .5% of all the edges). We then construct an "anti-cooperation graph" G^* from these edges. This graph has 85 nodes (one for each brain region, but with the brain regions of degree 0 left out), and there is an edge between two regions if they have a correlation below $-.2$ in a network G_i. We observe that G^* has 360 edges.

We want to colour this anti-cooperation graph with a proper k-colouring, because an independent set in the anti-cooperation graph could indicate a set of regions that cooperate strongly (because the set contains no anti-cooperation edges). We use the greedy colouring algorithm described in Section 3.5.1: we colour the nodes of $V(G^*)$ in some order, and assign to each node the lowest colour not yet taken by any of its neighbours. We order the nodes from high degree to low degree and we colour them in that order. So, we label the nodes $1, \ldots, 85$ such that $d(1) \geq d(2) \geq \ldots \geq d(85)$. This gives $\chi(G^*) = 3$. To find out whether these colour classes are meaningful, we investigate the graph G^* over the course of the colouring process and describe the intermediate subgraphs induced by the set of coloured vertices: $G^*_{(1)}, \ldots, G^*_{(85)}$. We also alter the process slightly after the first 17 nodes have been coloured, such that we can throw away nodes that might not have a meaningful colour-class assignment.

The first interesting subgraph is $G^*_{(17)}$. This is a bipartite graph (*i.e.*, $\chi(G^*_{(17)}) = 2$) on two sets of sizes 9 and 8. This graph is nearly complete: $|E(G^*_{(17)})| = 70$, so only two edges are missing. We will call these sets the *nuclei*[3] of the first two colour classes C_1 and C_2. We denote the nuclei by C^*_1 and C^*_2, and we will make special use of them when classifying the rest of the vertices, as we have a high degree of certainty that the nodes in the nuclei have been classified correctly.

[3] We are avoiding yet another definition of the word *core*.

After those first 17 nodes –*i.e.*, for nodes $18, \ldots, 85$– we change the greedy colouring algorithm slightly. The entire algorithm is given by GREEDY-BRAINS (Algorithm 8.1). Let $c(i) : V(G^*) \to C_1, C_2, C_3, C_R$, where C_R is the class for nodes that we decide not to classify.

Algorithm 8.1 GREEDY-BRAINS

for $i = 1$ to 17 **do**
  if $|\Gamma(i) \cap C_1| = 0$; **then**
    add i to C_1^*;
  else
    add i to C_2^*;
  end if
end for
for $i = 18$ to 85 **do**
  if $|\Gamma(i) \cap C_1| = 0$ **and** $|\Gamma(i) \cap C_2^*| \geq 2$; **then**
    add i to C_1;
  else if $|\Gamma(i) \cap C_2| = 0$ **and** $|\Gamma(i) \cap C_1^*| \geq 2$; **then**
    add i to C_2;
  else if $|\Gamma(i) \cap C_3| = 0$ **and** $|\Gamma(i) \cap C_1^*| \geq 2$ **and** $|\Gamma(i) \cap C_2^*| \geq 2$; **then**
    add i to C_3;
  else
    add i to C_R;
  end if
end for

After completion of GREEDY-BRAINS, a total of 63 nodes have been coloured. We list the nodes of each colour class in Tables 8.2, 8.3, and 8.4. For a full list of the brain regions, see Table 8 in the Appendix.

Community 1 is centred around nodes 7, 17, 18, and 55, which are the left precentral gyrus, the postcentral gyrus, the superior parietal lobule, and the right precentral gyrus. The precentral gyrus is the main motor centre of the brain and initiates muscle movement. The postcentral gyrus is the main sensory area that processes sensory information from the skin (touch, pressure, *etc.*) and position information from the spindle organs in the joints. The superior parietal lobule is involved in spatial orientation [281]. Community 2 is centred around nodes 95, 72, 47, and 24 which are the right supercalcarine cortex, the right intracalcarine cortex, the left supercalcarine cortex, and the left intracalcarine cortex. These

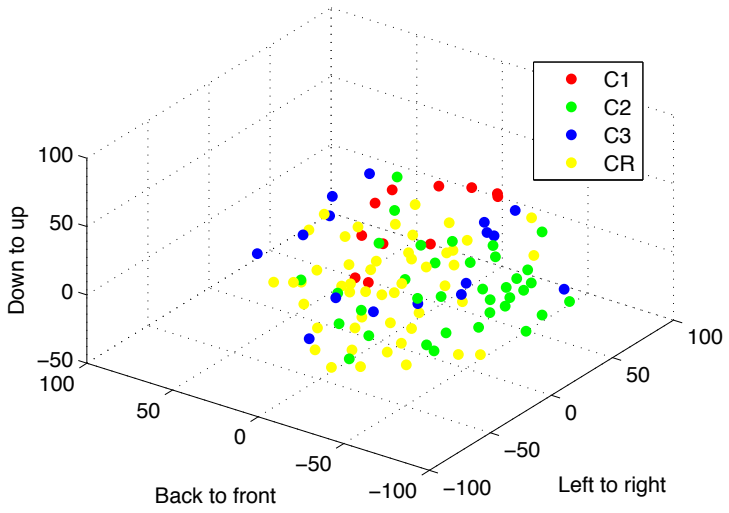


Figure 8.10: The three communities of the learning brain. C1: "motor control", C2: "visual processing", C3: "awareness", and CR: unclassified nodes. The locations of the points are their three-dimensional coordinates in the brain.

form the calcarine sulcus, where the primary visual cortex is located [281]. Community 3 is centred around nodes 1, 3, 68, and 2 which are the frontal pole, the superior frontal gyrus, the posterior supramarginal gyrus, and the insular cortex. The frontal pole has been linked to cognition [268] and decision making and evaluating [267]. The superior frontal gyrus has been linked to self-awareness [129]. The posterior supramarginal gyrus is involved in spatial orientation and semantic representation [281]. The insular cortex is involved in homeostasis and consciousness.

In Figure 8.10 we show a 3-dimensional plot of the locations of the regions in the brain, colour-coded by their community assignments. We name the three communities that we have found "motor control", "visual processing", and "awareness", but we note that these are crude labels, as they are based only on a first impression of the function of the nodes in those communities. We hope that this example opens the door to new methods of interpreting brain networks constructed from fMRI data sets, and that this will attract the attention of brain specialists who are equipped to interpret partitions such as this one. We will discuss this further in Section 9.5.

Table 8.2: Community 1: "Motor control". The columns indicate the region number, the hemisphere of the brain (left or right), the name of the region, and the degree of the node in the graph G^*.

R #	**HS**	**Region**	**Degree in G^***
7	L	Precentral gyrus	57
17	L	Postcentral gyrus	53
18	L	Superior parietal lobule	42
55	R	Precentral gyrus	39
74	R	Supplemental motor area	18
19	L	Supramarginal gyrus, anterior	15
66	R	Superior parietal lobule	13
67	R	Supramarginal gyrus, anterior	13
65	R	Postcentral gyrus	12
26	L	Supplemental motor area	11
75	R	Subcallosal cortex	3

Table 8.3: Community 2: "Visual processing". The columns indicate the region number, the hemisphere of the brain (left or right), the name of the region, and the degree of the node in the graph G^*.

R #	**HS**	**Region**	**Degree in G^***
95	R	Supercalcarine cortex	28
72	R	Intracalcarine cortex	25
47	L	Supercalcarine cortex	22
24	L	Intracalcarine cortex	17
80	R	Cuneus cortex	17
84	R	Lingual gyrus	16
32	L	Cuneus cortex	15
36	L	Lingual gyrus	13
79	R	Precuneus cortex	11
96	R	Occipital pole	11
48	L	Occipital pole	10
52	R	Middle frontal gyrus	6
31	L	Precuneus cortex	5
104	L	Brainstem	5
39	L	Temporal occipital fusiform cortex	4
45	L	Heschl's gyrus	4
50	R	Insular cortex	4
63	R	Inferior temporal gyrus, posterior	4
112	R	Brainstem	4
21	L	Angular gyrus	3
30	L	Cingulate gyrus, posterior	3
34	L	Parahippocampal gyrus, anterior	3
42	L	Central opercular cortex	3
44	L	Planum polare	3
105	R	Caudate	3
25	L	Frontal medial cortex	2
37	L	Temporal fusiform cortex, anterior	2
53	R	Inferior frontal gyrus, pars triangularis	2
56	R	Temporal pole	2
57	R	Superior temporal gyrus, anterior	2
60	R	Middle temporal gyrus, posterior	2
62	R	Inferior temporal gyrus, anterior	2
64	R	Inferior temporal gyrus, temporooccipital	2
70	R	Lateral occipital cortex, superior	2
87	R	Temporal occipital fusiform cortex	2
88	R	Occipital fusiform gyrus	2

Table 8.4: Community 3: "Awareness". The columns indicate the region number, the hemisphere of the brain (left or right), the name of the region, and the degree of the node in the graph G^*.

R #	**HS**	**Region**	**Degree in G^***
1	L	Frontal pole	12
3	L	Superior frontal gyrus	11
68	R	Supramarginal gyrus, posterior	11
2	L	Insular cortex	10
4	L	Middle frontal gyrus	10
51	R	Superior frontal gyrus	10
94	R	Planum temporale	10
58	R	Superior temporal gyrus, posterior	9
71	R	Lateral occipital cortex, inferior	9
22	L	Lateral occipital cortex, superior	8
8	L	Temporal pole	6
46	L	Planum temporale	6
82	R	Parahippocampal gyrus, anterior	6
83	R	Parahippocampal gyrus, posterior	6
91	R	Parietal operculum cortex	6
76	R	Paracingulate gyrus	5

Chapter 9

Conclusions and Future Work

9.1 Contributions

We use the categories of contributions to the field of network science set out in the introductory chapter of this text to summarise its contributions.

(A) Direct observations by measuring properties of empirical networks. Although we examined several empirical networks in Chapter 7–the Zachary Karate Club, the London Tube, two networks of network scientists, and the US Senate roll call voting network– these networks were mainly tools to analyse the method that we introduced and to make comparisons with existing methods. We expected each of these networks to contain a core-periphery organisation and we did discover one, but we did not use this explicitly to gain an understanding of these networks. Chapter 8, however, describes an experiment and an analysis of empirical networks – functional human brain networks – aimed to both introduce a set of new methods and gain an understanding of the networks. We found that reconfiguration patterns of dynamic communities can be used to classify nodes into a stiff core, a flexible periphery, and a bulk. The separation between this stiff core and flexible periphery changes as a person learns a simple motor skill and, importantly, it is a good predictor of how

successful the person is at learning the simple motor task. This temporally defined core-periphery organisation corresponds with the core-periphery structure detected by the method that we developed in Chapter 7 and apply to the static networks created by averaging over the subjects' dynamic functional brain networks.

(B) Suggestions of new (random) models motivated by (A); (C) Observations of computer simulations of the models proposed in (B); (G) Observations of the measures proposed in (F) on the models proposed in (B). In Chapter 7, we proposed a stochastic block model $CP(N, d, p, k)$ with an artificial core-periphery structure to use as computer-generated benchmark for core-periphery detection methods. We then use it to test several methods and as evidence that our new method finds the structure that we intend it to find.

(D) Heuristic investigations of these new models to predict their properties. We did not contribute to this category. The $CP(N, d, p, k)$ model is a straightforward stochastic block model and was only introduced as a testing ground for the different measures of core-periphery structure. It is not intended as a realistic model of core-periphery structure in empirical networks, instead it is a highly idealised and simplified model that is easy to understand and generate.

(E) Rigorous theorems and proofs about these models. We prove a rigorous theorem about $G(n, p)$ in Chapter 4. Namely that, we proved the following theorem.

Theorem 4.1.1. *If $n^{-1+\epsilon_1} < p < 1 - \epsilon_2$, for $\epsilon_1, \epsilon_2 > 0$, then almost surely*

$$\chi(G(n,p)) \leq \chi_{=}(G(n,p)) \leq \chi_{=}^{*}(G(n,p)) = (1 + o(1))\chi(G(n,p))$$

holds. (The first two inequalities are true by definition.)

(F) Suggestions of new ways to define, detect, and measure properties of networks. In Chapter 7, we introduced a new method to study core-periphery structure in weighted, undirected networks that was motivated by the continuous formulation of Borgatti and Everett [50]. However, our method takes cores of different size and shapes into account. The motivation for this method is, in part, the coexistence of community structure and core-periphery structure, and we provide examples of empirical networks that exhibit such a coexistence.

(H) Rigorous theorems and proofs about these measures. In Chapter 6, we attempt to gain some insight into several centrality measures from the perspective of deterministic (non-probabilistic) graph theory. We do this by searching for minimal examples of graphs on which various centrality measures fail to distinguish any of the vertices from one another, even though the graphs are not vertex-transitive. We examined the following two conjectures posed by Estrada.

Conjecture 6.7.1. *[100]* $\operatorname{var}(\mathbf{sc}_G) = 0 \Rightarrow$,
$\operatorname{var}(\mathbf{d}_G) = \operatorname{var}(\mathbf{ec}_G) = \operatorname{var}(\mathbf{cc}_G) = \operatorname{var}(\mathbf{bc}_G) = 0$.

Conjecture 6.7.2. *[97]* $\operatorname{var}(\mathbf{sc}_G) = 0 \Leftrightarrow$ *G is walk-regular.*

We show that Conjecture 6.7.1 is at least partially false by giving a counterexample for betweenness centrality and closeness centrality.

9.2 Equitable colourings

In Chapter 4, we resolved the conjecture by Krivelevich and Patkós posed in [170] for a wide range of p: namely, $n^{-1+\epsilon_1} < p < 1 - \epsilon_2$ for $\epsilon_1, \epsilon_2 > 0$. The next step is to bring p down to c/n for a positive constant c. At that lower limit, we will probably be able to use the

properties of extremely sparse graphs around the threshold of connectedness, but it will be a harder problem to resolve the conjecture for $p = \frac{f(n)}{n}$, where $f(n) \to \infty$ slowly.

9.3 Centrality

In Section 6.7, we resolved part of the conjecture posed by Estrada in [100]. Part of it remains open, as we pose in a modified conjecture.

Conjecture 6.7.6. $\mathrm{var}(\mathbf{sc}_G) = 0 \Leftrightarrow$ *G is walk-regular, and*
$\mathrm{var}(\mathbf{sc}_G) = \mathrm{var}(\mathbf{bc}_G) = \mathrm{var}(\mathbf{cc}_G) = 0 \Leftrightarrow$ *G is distance-regular.*

Additionally, we hope that this problem can provide an example of strengths and weaknesses of various centrality measures on different types of graphs. Hopefully, this adds understanding to the theoretical foundations of such measures and the importance of developing them alongside the understanding that one gains from applying them to empirical networks and synthetic benchmark networks.

9.4 Core-periphery structure

The purpose of the method presented in Chapter 7 is conceptual development, and the current implementation of the method is slow because we use simulated annealing (see Section 5.6.2 and 7.8). Additionally, when using two-parameter transition functions, we often used 10000 different (and uniformly-spaced) values of (α, β). One can improve speed considerably by considering fewer parameter values, designing schemes to sample values of α and β intelligently, or employing a one-parameter transition function.

Methods discussed in this chapter could be extended to multilayer networks, as was done for community structure (see Section 5.7). Although most of the methods in our thesis can be extended to directed networks, we did not pay any special attention to them.

Using modularity to detect core-periphery structure, analogously to how it is used to detect community structure, has not been investigated sufficiently. Defining it is not so hard, and it is also not too difficult to implement algorithms, as they could be very similar to existing ones for community structure. However, one will likely face the same obstacles as network scientists did (and still do) when developing modularity for community structure. Modularity optimization is very difficult to study rigorously, and the landscape of solutions for core-periphery is likely to show many near-optimal solutions in the same way as it does for community structure. Additionally, there might be problems with resolution limits (see [109] for a discussion of resolution in community structure). Core-periphery structure is a less developed concept than community structure, and we predict that it will follow a similar path to community structure. We hope that the knowledge that has been gained from studying community structure and the progress of network science as a whole in the meantime will allow ideas for core-periphery structure to be developed much faster. There is already a rapid increase in the frequency of publications on the subject and knowledge on the subject, to which we hope that our work will contribute.

9.5 Brain networks

The results described in Sections 8.5, 8.6 and 8.7 show that the mesoscopic structure of a functional brain network changes with learning of a new motor skill, and that the way in which it changes is a predictor for how fast an individual improves at performing such a skill. These results are promising, and we hope that they will spark future research into mesoscopic structures in the brain. In particular, we hope that this will increase the popularity of studying core-periphery structure in the brain.

The purpose of the results presented in Section 8.9 is to provide new ideas for methods of analysing brain networks created from imaging time series. This work is something to discuss with the neuroscientist collaborators to gain more insight from their perspective.

This method seems to find a pronounced structure, but it is not clear how surprising this structure is and what we can learn from it. This is the main part of the project to develop further in the short term.

Appendix

1 London underground

Station	Degree Centrality
King's Cross St. Pancras	12
Baker Street	10
Paddington	9
West Ham	8
Moorgate	8
Liverpool Street	8
Embankment	8
Willesden Junction	7
Waterloo	7
Stratford	7

Table 1: The ten top nodes in the London Underground network ranked by their degrees (number of lines running out of them; so, if a line goes in both directions it is counted twice).

Station	Betweenness Centrality
Willesden Junction	33432.4484
Stratford	28876.1757
Euston	27751.3842
Gospel Oak	26419.6174
Queen's Park	23187.1882
Dalston	22344.9587
Baker Street	22014.2198
Finchley Road	17604.9521
Green Park	17275.4481
King's Cross St. Pancras	16984.4017

Table 2: The ten top nodes in the London Underground network ranked by their betweenness centrality. In this case we did not consider multiple edges in the network.

Station	Closeness Centrality
Euston	0.0003873
Queen's Park	0.00038595
Willesden Junction	0.00038344
Oxford Circus	0.00037793
Warren Street	0.00037779
Green Park	0.00037779
King's Cross St. Pancras	0.00037383
Gospel Oak	0.00036819
Bond Street	0.0003667
Westminster	0.00036443

Table 3: The ten top nodes in the London Underground network ranked by their closeness centrality. In this case we did not consider multiple edges in the network.

Station	PageRank Centrality
King's Cross St. Pancras	1
Farringdon	0.80167
Euston Square	0.76244
Barbican	0.64627
Great Portland Street	0.56572
Moorgate	0.52549
Baker Street	0.3993
Euston	0.39272
Liverpool Street	0.34063
Edgware Road	0.19718

Table 4: The ten top nodes in the London Underground network ranked by their PageRank centrality.

2 Network of Network Scientists

In Table 5, we list the names and aggregate core scores (7.11) of the top 30 nodes for both the 2006 and 2010 network of network scientists. To compute the values in this table, we used the core quality (7.10) and the transition function (7.9).

In Table 6 in the Appendix, we list the top 30 aggregate core scores for the 2010 network using three variant computations: (left) using the one-parameter transition function (7.14) with the product form (7.8) for the core-matrix elements, (middle) using the smooth transition function (7.13) with the product form (7.8) and (right) using the usual transition function (7.9) with the p-norm (7.12) with $p = 2$ for the core-matrix elements.

NNS2006 Node	Core Score	NNS2010 Node	Core Score
Barabási, A.-L.	1.00	Barabási, A.-L.	1.00
Oltvai, Z. N.	0.97	Newman, M. E. J.	0.94
Jeong, H.	0.96	Pastor-Satorras, R.	0.93
Vicsek, T.	0.95	Latora, V.	0.93
Kurths, J.	0.88	Arenas, A.	0.93
Neda, Z.	0.87	Moreno, Y.	0.92
Ravasz, E.	0.86	Jeong, H.	0.92
Newman, M. E. J.	0.86	Vespignani, A.	0.91
Pastor-Satorras, R.	0.85	Díaz-Guilera, A.	0.90
Schubert, A.	0.85	Guimerà, R.	0.90
Boccaletti, S.	0.85	Watts, D. J.	0.89
Vespignani, A.	0.84	Vazquez, A.	0.89
Farkas, I.	0.84	Viczek, T.	0.89
Derenyi, I.	0.83	Amaral, L. A. N.	0.89
Holme, P.	0.82	Solé, R. V.	0.88
Crucitti, P.	0.81	Albert, R.	0.87
Albert, R.	0.80	Kahng, B.	0.87
Schnitzler, A.	0.80	Boccaletti, S.	0.86
Solé, R.	0.80	Oltvai, Z. N.	0.86
Rosenblum, M.	0.79	Barthélémy, M.	0.85
Tomkins, A.	0.79	Kurths, J.	0.84
Moreno, Y.	0.78	Fortunato, S.	0.84
Latora, V.	0.78	Marchiori, M.	0.83
Rajagopalan, S.	0.78	Kertesz, J.	0.83
Raghavan, P.	0.77	Caldarelli, G.	0.82
Pikovsky, A.	0.76	Dorogovtsev, S. N.	0.81
Kahng, B.	0.75	Boguñá, M.	0.80
Diazguilera, A.	0.74	Goh, K. I.	0.80
Vazquez, A.	0.74	Crucitti, P.	0.80
Kim, B.	0.74	Strogatz, S. H.	0.80

Table 5: The 30 nodes with the top aggregate core scores (7.11) for the (left) 2006 and (right) 2010 networks of network scientists. We used the core quality (7.10) and the transition function (7.9).

NNS2010 Node	SP& PN	NNS2010 Node	SmF & PN	NNS2010 Node	ShF & 2N
Barabási, A.-L.	1.0000	Barabási, A.-L.	1.0000	Barabási, A.-L.	1.0000
Jeong, H.	.9868	Moreno, Y.	.9702	Newman, M. E. J.	.9954
Vespignani, A.	.9859	Vespignani, A.	.9536	Pastor-Satorras, R.	.9932
Pastor-Satorras, R.	.9851	Jeong, H.	.9361	Jeong, H.	.9910
Newman, M. E. J.	.9788	Newman, M. E. J.	.9176	Vespignani, A.	.9888
Arenas, A.	.9765	Arenas, A.	.9129	Moreno, Y.	.9888
Moreno, Y.	.9762	Guimerà, R.	.8942	Díaz-Guilera, A.	.9862
Latora, V.	.9649	Díaz-Guilera, A.	.8809	Latora, V.	.9829
Guimerà, R.	.9638	Pastor-Satorras, R.	.8755	Arenas, A.	.9819
Vazquez, A.	.9616	Boccaletti, S.	.8686	Solé, R.V.	.9812
Díaz-Guilera, A.	.9604	Vicsek, T.	.8355	Amaral, L. A. N.	.9773
Vicsek, T.	.9491	Amaral, L. A. N.	.8341	Boccaletti, S.	.9768
Amaral, L. A. N.	.9470	Latora, V.	.8130	Vicsek, T.	.9737
Albert, R.	.9415	Barthélémy, M.	.8107	Guimerà, R.	.9712
Boccaletti, S.	.9379	Vazquez, A.	.8069	Vazquez, A.	.9689
Watts, D. J.	.9346	Kurths, J.	.7714	Kahng, B.	.9679
Solé, R. V.	.9321	Kahng, B.	.7633	Kurths, J.	.9676
Kahng, B.	.9309	Oltvai, Z. N.	.7616	Kertesz, J.	.9624
Kurths, J.	.9241	Caldarelli, G.	.7462	Bornholdt, S.	.9577
Oltvai, Z. N.	.9197	Kertesz, J.	.7096	Dorogovtsev, S. N.	.9554
Barthélémy, M.	.9183	Albert, R.	.7023	Marchiori, M.	.9549
Marchiori, M.	.9167	Watts, D. J.	.6861	Watts, D. J.	.9526
Caldarelli, G.	.9022	Porter, M. A.	.6842	Albert, R.	.9493
Kertesz, J.	.8914	Solé, R. V.	.6823	Barthélémy, M.	.9488
Fortunato, S.	.8883	Fortunato, S.	.6761	Oltvai, Z. N.	.9478
Goh, K. I.	.8852	Kaski, K.	.6752	Caldarelli, G.	.9474
Kim, D.	.8836	Tomkins, A. S.	.6648	Havlin, S.	.9458
Danon, L.	.8773	Boguñá, M.	.6584	Mendes, J.F.F.	.9443
Boguñá, M.	.8747	Goh, K. I.	.6458	Stauffer, D.	.9408
Strogatz, S. H.	.8690	Kim, D.	.6411	Tomkins, A. S.	.9401

Table 6: The 30 nodes with the top aggregate core scores from the 2010 networks of network scientists. From left to right, we computed these scores using the single-parameter transition function (7.14) and the product normalization (7.8) (using the parameter values $\alpha = [0.0001 : 0.0001 : 1]$ in MATLAB notation), the smooth two-parameter function (7.13) and the product normalization (using the parameter values $\alpha = \beta = [0.01 : 0.01 : 1]$ in MATLAB notation), and the sharp two-parameter function (7.9) and the 2-norm normalization [i.e., (7.12) with $p = 2$] (again using $\alpha = \beta = [0.01 : 0.01 : 1]$ in MATLAB notation). Note that the second column in Table 5 uses the sharp two-parameter function and the product norm.

3 Voting Similarities in the United States Senate

In Table 7, we show aggregate core scores for Senators in the 108th Congress. We calculated these core scores using the core quality (7.10) and the transition function (7.9).

Table 7: Senators in the 108th Congress along with their aggregate core scores (7.11) and the percentage of bills for which they voted in line with their political parties. We determined the core scores using the core quality (7.10) and the transition function (7.9). The party affiliations are indicated by ■ for Republican and ■ for Democrat.

Node	Core Score	Party Vote
■ Chuck Grassley [R - IA]	1	97%
■ Thad Cochran [R - MS]	0.9864	98%
■ Mitch McConnell [R - KY]	0.9628	98%
■ Pete Domenici [R - NM]	0.9476	96%
■ Bill Frist [R - TN]	0.8943	97%
■ Pat Roberts [R - KS]	0.8712	97%
■ Conrad Burns [R - MT]	0.8595	96%
■ Jim Bunning [R - KY]	0.8472	97%
■ Saxby Chambliss [R - GA]	0.8132	97%
■ Orrin Hatch [R - UT]	0.7969	97%
■ Bob Bennett [R - UT]	0.7966	97%
■ Jim Talent [R - MO]	0.7625	97%
■ Kit Bond [R - MO]	0.7481	96%
■ Ted Stevens [R - AK]	0.7177	96%
■ John Cornyn [R - TX]	0.6890	96%
■ Mike Crapo [R - ID]	0.6819	96%
■ Liddy Dole [R - NC]	0.6739	96%
■ Sam Brownback [R - KS]	0.6736	96%
■ Lamar Alexander [R - TN]	0.6676	97%
■ Larry Craig [R - ID]	0.6540	96%
■ George Allen [R - VA]	0.6323	96%
■ Richard Shelby [R - AL]	0.6094	95%
■ James Inhofe [R - OK]	0.5977	96%
■ Richard Lugar [R - IN]	0.5918	96%
■ Trent Lott [R - MS]	0.5822	95%
■ Chuck Hagel [R - NE]	0.5732	95%
■ Craig Thomas [R - WY]	0.5525	95%
■ Wayne Allard [R - CO]	0.5357	95%
■ Zell Miller [D - GA]	0.5327	38%
■ Gordon Smith [R - OR]	0.5324	94%
■ Lisa Murkowski [R - AK]	0.5277	95%

Node	Core Score	Party Vote
■ Norm Coleman [R - MN]	0.5189	94%
■ John Warner [R - VA]	0.5145	94%
■ Lindsey Graham [R - SC]	0.5055	94%
■ Jeff Sessions [R - AL]	0.5009	94%
■ Mike Enzi [R - WY]	0.4885	94%
■ Rick Santorum [R - PA]	0.4741	94%
■ Ben Campbell [R - CO]	0.4658	93%
■ Peter Fitzgerald [R - IL]	0.4492	93%
■ Donald Nickles [R - OK]	0.4420	93%
■ Kay Bailey Hutchison [R - TX]	0.4387	92%
■ George Voinovich [R - OH]	0.4305	92%
■ Mike DeWine [R - OH]	0.4290	91%
■ Jon Kyl [R - AZ]	0.4169	93%
■ John Sununu [R - NH]	0.3996	91%
■ John Ensign [R - NV]	0.3923	90%
■ Arlen Specter [R - PA]	0.3898	85%
■ Judd Gregg [R - NH]	0.3821	90%
■ Susan Collins [R - ME]	0.3789	84%
■ John McCain [R - AZ]	0.3687	84%
■ Olympia Snowe [R - ME]	0.3667	82%
■ Lincoln Chafee [R - RI]	0.3580	78%
■ Ben Nelson [D - NE]	0.3512	72%
■ John Breaux [D - LA]	0.3448	74%
■ Max Baucus [D - MT]	0.3378	82%
■ Patty Murray [D - WA]	0.3339	95%
■ Mary Landrieu [D - LA]	0.3322	85%
■ Blanche Lincoln [D - AR]	0.3274	87%
■ Tim Johnson [D - SD]	0.3192	94%
■ Mark Pryor [D - AR]	0.3172	89%
■ Evan Bayh [D - IN]	0.3169	86%
■ Kent Conrad [D - ND]	0.3038	88%
■ Byron Dorgan [D - ND]	0.3036	91%
■ Debbie Stabenow [D - MI]	0.3023	96%
■ Tom Carper [D - DE]	0.3021	86%
■ Barbara Mikulski [D - MD]	0.2994	96%
■ Harry Reid [D - NV]	0.2960	93%
■ Tom Daschle [D - SD]	0.2927	94%
■ Ron Wyden [D - OR]	0.2904	93%
■ Bill Nelson [D - FL]	0.2899	93%
■ Maria Cantwell [D - WA]	0.2836	95%
■ Chuck Schumer [D - NY]	0.2774	94%
■ Jeff Bingaman [D - NM]	0.2732	92%

Node	Core Score	Party Vote
■ Herb Kohl [D - WI]	0.2704	94%
■ Dianne Feinstein [D - CA]	0.2616	92%
■ Mark Dayton [D - MN]	0.2522	93%
■ Hillary Clinton [D - NY]	0.2261	95%
■ Jay Rockefeller [D - WV]	0.2254	93%
■ Chris Dodd [D - CT]	0.2209	94%
■ Carl Levin [D - MI]	0.2181	95%
■ Joseph Lieberman [D - CT]	0.2154	93%
■ Joe Biden [D - DE]	0.2140	92%
■ Patrick Leahy [D - VT]	0.2028	94%
■ James Jeffords [I - VT]	0.1964	88%
■ Daniel Inouye [D - HI]	0.1921	93%
■ Paul Sarbanes [D - MD]	0.1765	96%
■ Dick Durbin [D - IL]	0.1732	95%
■ Barbara Boxer [D - CA]	0.1718	95%
■ Jon Corzine [D - NJ]	0.1686	95%
■ Edward Kennedy [D - MA]	0.1643	94%
■ Daniel Akaka [D - HI]	0.1593	94%
■ Russ Feingold [D - WI]	0.1505	91%
■ John Edwards [D - NC]	0.1386	96%
■ Jack Reed [D - RI]	0.1378	95%
■ John Kerry [D - MA]	0.1246	98%
■ Tom Harkin [D - IA]	0.1138	94%
■ Fritz Hollings [D - SC]	0.1097	88%
■ Frank Lautenberg [D - NJ]	0.1006	94%
■ Robert Byrd [D - WV]	0.0997	90%
■ Bob Graham [D - FL]	0.0559	93%

4 Brain regions of the Harvard-Oxford Brain Atlas

Table 8: Brain regions in the Harvard-Oxford Cortical and Subcortical Parcellation Scheme provided by FSL [252, 284]. The columns indicate the region number, the hemisphere of the brain (left or right), and the name of the region.

R #	HS	Region Name
1	L	Frontal pole
2	L	Insular cortex
3	L	Superior frontal gyrus
4	L	Middle frontal gyrus
5	L	Inferior frontal gyrus, pars triangularis
6	L	Inferior frontal gyrus, pars opercularis
7	L	Precentral gyrus
8	L	Temporal pole
9	L	Superior temporal gyrus, anterior
10	L	Superior temporal gyrus, posterior
11	L	Middle temporal gyrus, anterior
12	L	Middle temporal gyrus, posterior
13	L	Middle temporal gyrus, temporooccipital
14	L	Inferior temporal gyrus, anterior
15	L	Inferior temporal gyrus, posterior
16	L	Inferior temporal gyrus, temporooccipital
17	L	Postcentral gyrus
18	L	Superior parietal lobule
19	L	Supramarginal gyrus, anterior
20	L	Supramarginal gyrus, posterior
21	L	Angular gyrus
22	L	Lateral occipital cortex, superior
23	L	Lateral occipital cortex, inferior
24	L	Intracalcarine cortex
25	L	Frontal medial cortex
26	L	Supplemental motor area
27	L	Subcallosal cortex
28	L	Paracingulate gyrus
29	L	Cingulate gyrus, anterior
30	L	Cingulate gyrus, posterior
31	L	Precuneus cortex
32	L	Cuneus cortex
33	L	Orbital frontal cortex
34	L	Parahippocampal gyrus, anterior
35	L	Parahippocampal gyrus, posterior

R #	HS	Region Name
36	L	Lingual gyrus
37	L	Temporal fusiform cortex, anterior
38	L	Temporal fusiform cortex, posterior
39	L	Temporal occipital fusiform cortex
40	L	Occipital fusiform gyrus
41	L	Fronal operculum cortex
42	L	Central opercular cortex
43	L	Parietal operculum cortex
44	L	Planum polare
45	L	Heschl's gyrus
46	L	Planum temporale
47	L	Supercalcarine cortex
48	L	Occipital pole
49	R	Frontal pole
50	R	Insular cortex
51	R	Superior frontal gyrus
52	R	Middle frontal gyrus
53	R	Inferior frontal gyrus, pars triangularis
54	R	Inferior frontal gyrus, pars opercularis
55	R	Precentral gyrus
56	R	Temporal pole
57	R	Superior temporal gyrus, anterior
58	R	Superior temporal gyrus, posterior
59	R	Middle temporal gyrus, anterior
60	R	Middle temporal gyrus, posterior
61	R	Middle temporal gyrus, temporooccipital
62	R	Inferior temporal gyrus, anterior
63	R	Inferior temporal gyrus, posterior
64	R	Inferior temporal gyrus, temporooccipital
65	R	Postcentral gyrus
66	R	Superior parietal lobule
67	R	Supramarginal gyrus, anterior
68	R	Supramarginal gyrus, posterior
69	R	Angular gyrus
70	R	Lateral occipital cortex, superior
71	R	Lateral occipital cortex, inferior
72	R	Intracalcarine cortex
73	R	Frontal medial cortex
74	R	Supplemental motor area
75	R	Subcallosal cortex
76	R	Paracingulate gyrus
77	R	Cingulate gyrus, anterior
78	R	Cingulate gyrus, posterior

R #	HS	Region Name
79	R	Precuneus cortex
80	R	Cuneus cortex
81	R	Orbital frontal cortex
82	R	Parahippocampal gyrus, anterior
83	R	Parahippocampal gyrus, posterior
84	R	Lingual gyrus
85	R	Temporal fusiform cortex, anterior
86	R	Temporal fusiform cortex, posterior
87	R	Temporal occipital fusiform cortex
88	R	Occipital fusiform gyrus
89	R	Fronal operculum cortex
90	R	Central opercular cortex
91	R	Parietal operculum cortex
92	R	Planum polare
93	R	Heschl's gyrus
94	R	Planum temporale
95	R	Supercalcarine cortex
96	R	Occipital pole
97	L	Caudate
98	L	Putamen
99	L	Globus pallidus
100	L	Thalamus
101	L	Nucleus Accumbens
102	L	Parahippocampal gyrus (superior to ROIs 34,35)
103	L	Hippocampus
104	L	Brainstem
105	R	Caudate
106	R	Putamen
107	R	Globus pallidus
108	R	Thalamus
109	R	Nucleus Accumbens
110	R	Parahippocampal gyrus (superior to ROIs 34,35)
111	R	Hippocampus
112	R	Brainstem

Bibliography

[1] S. Achard and E. T. Bullmore. Efficiency and cost of economical brain functional networks. *PLoS Computational Biology*, 3(2):e17, 2007.

[2] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Communities and hierarchical organization of links in complex networks. *arXiv:0903.3178*, 2009.

[3] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180, 2000.

[4] N. Alon. The maximum number of Hamiltonian paths in tournaments. *Combinatorica*, 10(4):319–324, 1990.

[5] N. Alon, M. Capalbo, Y. Kohayakawa, V. Rodl, A. Ruciński, and E. Szemerédi. Universality and tolerance. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 14–21. IEEE, 2000.

[6] N. Alon, O. Schwartz, and A. Shapira. An elementary construction of constant-degree expanders. *Combinatorics, Probability and Computing*, 17(03):319–327, 2008.

[7] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience, third edition, 2008.

[8] J. M. Anthonisse. The rush in a directed graph. *Stichting Mathematisch Centrum voor de Mathematische Besliskunde*, 9:1–10, 1971.

[9] K. Appel and W. Haken. Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, 1977.

[10] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, volume 1. World Scientific Publishing Company, 2011.

[11] K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967.

[12] J. P. Bagrow and E. M. Bollt. Local method for detecting communities. *Physical Review E*, 72(4):046108, 2005.

[13] P. Bak and K. Sneppen. Punctuated equilibrium and criticality in a simple model of evolution. *Physical Review Letters*, 71(24):4083–4086, 1993.

[14] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[15] A. D. Barbour. Poisson convergence and random graphs. *Math. Proc. Cambridge Philos. Soc*, 92(2):349–359, 1982.

[16] S. Barnard and H. Simon. Fast implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and Experience*, 6(2):101–117, 1994.

[17] G. Baron and W. Imrich. Asymmetrische reguläre graphen. *Acta Mathematica Academiae Scientiarum Hungaricae*, 20:135—142, 2009.

[18] D. S. Bassett, J. A. Brown, V. Deshpande, J. M. Carlson, and S. T. Grafton. Conserved and variable architecture of human white matter connectivity. *NeuroImage*, 54(2):1262–1279, 2011.

[19] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton. Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences of the United States of America*, 108(18):7641–7646, 2011.

[20] D. S. Bassett, N. F. Wymbs, M. P. Rombach, M. A. Porter, P. J. Mucha, and S. T. Grafton. Task-based core-periphery organization of human brain dynamics. *PLOS Computational Biology*, in press 2013.

[21] A. Békéssy, P. Bekessy, and J. Komlós. Asymptotic enumeration of regular matrices. *Studia Scientiarum Mathematicarum Hungarica*, 7:343–353, 1972.

[22] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, 1978.

[23] E. A. Bender and H. S. Wilf. A theoretical analysis of backtracking in the graph coloring problem. *Journal of Algorithms*, 6(2):275–282, 1985.

[24] C. T. Benson. Minimal regular graphs of girth 8 and 12. *Canadian Journal of Mathematics*, 18:1091–1094, 1966.

[25] D. Bertsimas and J. Tsitsiklis. Simulated annealing. *Statistical Science*, 8(1):10–15, 1993.

[26] M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005.

[27] G. Bianconi. The entropy of randomized network ensembles. *EPL (Europhysics Letters)*, 81(2):28005, 2008.

[28] G. Bianconi, A. C. C. Coolen, and C. J. P. Vicente. Entropies of complex networks with hierarchically constrained topologies. *Physical Review E*, 78(1):016114, 2008.

[29] J. Blazewicz. *Scheduling computer and manufacturing processes*. Springer, 2001.

[30] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, , and E. Lefebvre. Fast unfolding of communities in large network. *Journal of Statistical Mechanics: Theory and Experiment*, (P10008), 2008.

[31] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[32] S. Boettcher and A. G. Percus. Extremal optimization for graph partitioning. *Physical Review E*, 64(2):026114, 2001.

[33] B. Bollobás. The distribution of the maximum degree of a random graph. *Discrete Mathematics*, 32(2):201–203, 1980.

[34] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1:311–316, 1980.

[35] B. Bollobás. Threshold functions for small subgraphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 90, pages 197–206. Cambridge Univ Press, 1980.

[36] B. Bollobás. Degree sequences of random graphs. *Discrete Mathematics*, 33(1):1–19, 1981.

[37] B. Bollobás. The diameter of random graphs. *Transactions of the American Mathematical Society*, 267(1):41–52, 1981.

[38] B. Bollobás. Vertices of given degree in a random graph. *Journal of Graph Theory*, 6(2):147–155, 1982.

[39] B. Bollobás. The chromatic number of random graphs. *Combinatorica*, 8(1):49–55, 1988.

[40] B. Bollobás. *Modern Graph Theory*. Springer, 1998.

[41] B. Bollobás. *Random Graphs*. Number 73 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2001.

[42] B. Bollobás and P. Erdös. Cliques in random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 80(3):419–427, 1976.

[43] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18(3):279–290, 2001.

[44] B. Bollobás and O. M. Riordan. Linearized chord diagrams and an upper bound for vassiliev invariants. *Journal of Knot Theory and its Ramifications*, 9(7):847–853, 2000.

[45] B. Bollobás and O. M. Riordan. Mathematical results on scale-free random graphs. In *Handbook of graphs and networks*, pages 1–34. 2003.

[46] B. Bollobás and O. M. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34, 2004.

[47] L. Boltzmann. *Lectures on Gas Theory*, 1896.

[48] P. Bonacich. Technique for analyzing overlapping memberships. *Sociological Methodology*, 4:176–185, 1972.

[49] P. Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[50] S. P. Borgatti and M. G. Everett. Models of core / periphery structures. *Social Networks*, 21:375–395, 1999.

[51] J. P. Boyd, W. J. Fitzgerald, M. C. Mahutga, and D. A. Smith. Computing continuous core/periphery structures for social relations data with MINRES/SVD. *Social Networks*, 32(2):125–137, 2010.

[52] M. Brammer. Multidimensional wavelet analysis of functional magnetic resonance images. *Human Brain Mapping*, 6:378–382, 1998.

[53] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[54] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20:172–188, 2008.

[55] R. L. Brieger. Career attributes and network structure: A blockmodel study of a biomedical research specialty. *American Sociological Review*, 41(1):117–135, 1976.

[56] A. E. Brouwer and W. H. Haemers. *Distance-Regular Graphs*. Springer, 2012.

[57] F. Buckley and F. Harary. *Distance in Graphs*. Addison-Wesley Redwood City, 1990.

[58] P. G. Buckley and D. Osthus. Popularity based random graph models leading to a scale-free degree sequence. *Discrete Mathematics*, 282(1):53–68, 2004.

[59] E. T. Bullmore and D. S. Bassett. Brain graphs: Graphical models of the human brain connectome. *Annual Review of Clinical Psychology*, 7:113–140, 2011.

[60] E. T. Bullmore, J. Fadili, M. Breakspear, R. Salvador, J. Suckling, and M. Brammer. Wavelets and statistical analysis of functional magnetic resonance images of the human brain. *Statistical Methods in Medical Research*, 12(5):375–399, 2003.

[61] E. T. Bullmore, J. Fadili, V. Maxim, L. endur, B. Whitcher, J. Suckling, M. Brammer, and M. Breakspear. Wavelets and functional magnetic resonance imaging of the human brain. *NeuroImage*, 23:234–249, 2004.

[62] E. T. Bullmore and O. Sporns. Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10:186–198, 2009.

[63] J. D. Burtin. Asymptotic estimates of the diameter and the independence and domination numbers of a random graph. In *Doklady Akademii Nauk SSSR*, volume 209, pages 765–768, 1973.

[64] P. J. Cameron. Random strongly regular graphs? *Discrete mathematics*, 273(1):103–114, 2003.

[65] A. L. Cauchy. *Mémoire sur l'intégration des équations linéaires aux différences partielles et à coefficiens constans*. 1823.

[66] V. Černỳ. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

[67] L.-C. Chang. The uniqueness and non-uniqueness of the triangular association scheme. *Scientific Record of the Peking Mathematical Society*, 3:604–613, 1959.

[68] C. Chase-Dunn. *Global Formation: Structures of the World-Economy*. Basil Blackwell, 1989.

[69] J. Chen and B. Yuan. Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, 22(18):2283–2290, 2006.

[70] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[71] F. Chung and L. Lu. The diameter of sparse random graphs. *Advances in Applied Mathematics*, 26(4):257–279, 2001.

[72] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences of the United States of America*, 99:15879–15882, 2002.

[73] F. Chung and L. Lu. Connected components in random graphs with given degree sequence. *Annals of Combinatorics*, 6:125–145, 2002.

[74] F. Chung and L. Lu. Concentration inequalities and martingale inequalities: A survey. *Internet Mathematics*, 3(1):79–127, 2006.

[75] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 2nd edition, 1997.

[76] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[77] A. L. Comrey. The minimum residual method of factor analysis. *Psychological Reports*, 11:15–18, 1962.

[78] C. Cooper and A. Frieze. A general model of web graphs. *Random Structures & Algorithms*, 22(3):311–335, 2003.

[79] C. Corradino. Proximity structure in a captive colony of Japanese monkeys (macaca fuscata fuscata): An application of multidimensional scaling. *Primates*, 31(3):351–362, 1990.

[80] H. Cramér. Sur un nouveau théorème-limite de la théorie des probabilités. *Actualités scientifiques et industrielles*, 736(115):5–23, 1938.

[81] R. B. Darlington, S. L. Weinberg, and H. J. Walberg. Canonical variate analysis and related techniques. *Review of Educational Research*, 43(4):433–454, 1973.

[82] F. Della Rossa, F. Dercole, and C. Piccardi. Profiling core-periphery network structure by random walkers. *Scientific Reports*, 3:1467, 2013.

[83] I. Derényi, G. Palla, and T. Vicsek. Clique percolation in random networks. *Physical review letters*, 94(16):160202, 2005.

[84] G. A. Dirac. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, 3(1):69–81, 1952.

[85] P. Doreian. Structural equivalence in a psychology journal network. *Journal of the American Society for Information Science*, 36(6):411–417, 1985.

[86] P. Doreian, V. Batagelj, and A. Ferligoj. *Generalized Blockmodeling*. Cambridge University Press, 2004.

[87] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633, 2000.

[88] E. Drinea, M. Enachescu, and M. Mitzenmacher. Variations on random graph models for the web. Technical report, Harvard University, Department of Computer Science, 2001.

[89] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72:027104, 2005.

[90] D. Edler and M. Rosvall. *The Map Generator software package*, 2010. `http://www.mapequation.org`.

[91] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and Finite Sets*, 10:609–627, 1975.

[92] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290—297, 1959.

[93] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[94] P. Erdős and A. Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Hungarica*, 12(1):261–267, 1961.

[95] P. Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(292-294):2, 1947.

[96] P. Erdös. Graph theory and probability II. *Canadian Journal of Mathematics*, 13:346–352, 1961.

[97] E. Estrada. About the discriminant power of the subgraph centrality and other centrality measures (working paper). *arXiv:1305.6836*, 2013.

[98] E. Estrada and N. Hatano. Communicability in complex networks. *Physical Review E*, 77(3):036111, 2008.

[99] E. Estrada, N. Hatano, and M. Benzi. The physics of communicability in complex networks. *Physics Reports*, 514(3):89–119, 2012.

[100] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.

[101] L. Euler. *The Seven Bridges of Königsberg*, 1736.

[102] T. S. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1):016105, 2009.

[103] M. G. Everett and S. B. Borgatti. Regular equivalence: General theory. *Journal of the Mathematical Society*, 19(1):29–52, 1994.

[104] I. J. Farkas, I. Derényi, A.-L. Barabási, and T. Vicsek. Spectra of "real-world graphs: Beyond the semicircle law. *Physical Review E*, 64(2):026704, 2001.

[105] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998.

[106] W. Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

[107] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. *Computer*, 35(3):66–70, 2002.

[108] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, 2010.

[109] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1):36–41, 2007.

[110] S. Fortunato, V. Latora, and M. Marchiori. Method to find community structures based on information centrality. *Physical Review E*, (70):056104, 2004.

[111] J. E. Fowler. The redundant discrete wavelet transform and additive noise. *IEEE Signal Processing Letters*, 12(9):629–632, 2005.

[112] J. H. Fowler. Connecting the Congress: A study of legislative cosponsorship networks. *Political Analysis*, 14(4):454–465, 2006.

[113] J. H. Fowler. Legislative cosponsorship networks in the U.S. House and Senate. *Social Networks*, 28(4):456–487, 2006.

[114] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[115] L. C. Freeman. The impact of computer based communication on the social structure of an emerging scientific specialty. *Social networks*, 6(3):201–221, 1984.

[116] L. C. Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press, 2004.

[117] A. M. Frieze. On the independence number of random graphs. *Discrete Mathematics*, 81(2):171–175, 1990.

[118] S. Gago Álvarez, J. Hurajová, and T. Madaras. Betweenness-selfcentric graphs. 2012. `http://hdl.handle.net/2117/15768`.

[119] P. Galinier, J.-P. Hamiez, J.-K. Hao, and D. Porumbel. Recent advances in graph vertex coloring. In *Handbook of Optimization*, pages 505–528. Springer, 2013.

[120] A. Garas, F. Schweitzer, and S. Havlin. A k-shell decomposition method for weighted networks. *New Journal of Physics*, 14:083030, 2012.

[121] M. R. Garey and D. S. Johnson. The complexity of near-optimal graph coloring. *Journal of the Association for Computing Machinery*, 23(1):43–49, 1976.

[122] A. Georgakopoulos. On walk-regular graphs and graphs with symmetric hitting times. *arXiv:1211.5689*, 2012.

[123] R. Ghosh and K. Lerman. Rethinking centrality: The role of dynamical processes in social network analysis. *arXiv:1209.4616*, 2012.

[124] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space*, pages 225–234. 1998.

[125] E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[126] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.

[127] D. F. Gleich. *Models and Algorithms for PageRank Sensitivity*. PhD thesis, Stanford University, 2009. Chapter 7 on MatlabBGL.

[128] C. D. Godsil and B. D. McKay. Feasibility conditions for the existence of walk-regular graphs. *Linear Algebra and its Applications*, 30:51–61, 1980.

[129] I. I. Goldberg, M. Harel, and R. Malach. When the brain loses its self: Prefrontal inactivation during sensorimotor processing. *Neuron*, 50(2):329–339, 2006.

[130] D. Gómez, J. Montero, and J. Yáñez. A coloring fuzzy graph approach for image classification. *Information Sciences*, 176(24):3645–3657, 2006.

[131] B. H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[132] B. H. Good, Y.-A. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.

[133] M. Granovetter. The strength of weak ties: A network theory revisited. *Sociological Theory*, 1(1):201–233, 1983.

[134] G. R. Grimmett and C. J. H. McDiarmid. On colouring random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 77(02):313–324, 1975.

[135] B. Grünbaum. A result on graph-coloring. *The Michigan Mathematical Journal*, 15(3):381–383, 1968.

[136] S. Guattery and G. L. Miller. On the performance of spectral graph partitioning methods. *Proceedings of the 6th ACM-Society for Industrial and Applied Mathematics Symposium on Discrete Algorithms*, pages 233–242, 1995.

[137] M. Gurevich. *The Social Structure of Acquaintanceship Networks*. PhD thesis, Massachusetts Institute of Technology, 1961.

[138] V. Guruswami and S. Khanna. On the hardness of 4-coloring a 3-collorable graph. In *Proceedings of the 15th Annual IEEE Conference on Computational Complexity*, pages 188–197. IEEE, 2000.

[139] P. Hage and F. Harary. Eccentricity and centrality in networks. *Social Networks*, 17(1):57–63, 1995.

[140] A. Hajnal and E. Szemerédi. Proof of a conjecture of Erdös. *Combinatorial theory and its applications*, 2:601–623, 1970.

[141] P. Hansen and M. Delattre. Complete-link cluster analysis by graph coloring. *Journal of the American Statistical Association*, 73(362):397–403, 1978.

[142] F. Harary. Status and contrastatus. *Sociometry*, 22(1):23–43, 1959.

[143] H. A. Harrington, M. Beguerisse Díaz, M. P. Rombach, L. M. Keating, and M. A. Porter. Teach network science to teenagers. *arXiv:1302.6567*, 2013.

[144] M. P. van den Heuvel, C. J. Stam, M. Boersma, and H. E. Hulshoff Pol. Small-world and scale-free organization of voxel-based resting-state functional connectivity in the human brain. *NeuroImage*, 33:528–39, 2008.

[145] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[146] P. Holme. Core-periphery organization of complex networks. *Physical Review E*, 72(4):046111, 2005.

[147] G. I. Ivchenko. On the asymptotic behavior of degrees of vertices in a random graph. *Theory of Probability & Its Applications*, 18(1):188–195, 1973.

[148] S. Janson, D. E. Knuth, T. Łuczak, and B. Pittel. The birth of the giant component. *Random Structures & Algorithms*, 4(3):233–358, 1993.

[149] S. Janson, T. Łuczak, and Ruciński. *Random graphs*. Cambridge University Press, 2000.

[150] T. R. Jensen and B. Toft. *Graph coloring problems*, volume 39. John Wiley & Sons, 2011.

[151] M. Jerrum and G. B. Sorkin. Simulated annealing for graph bisection. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 94–103. IEEE, 1993.

[152] C. Jordan. Sur les assemblages de lignes. *Journal für die reine und angewandte Mathematik*, 70:185–190, 1869.

[153] I. S. Jutla, L. G. S. Jeub, and P. J. Mucha. A generalized Louvain method for community detection implemented in MATLAB, 2011–2012. `http://netwiki.amath.unc.edu/GenLouvain`.

[154] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1988.

[155] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *Journal of the Association for Computing Machinery*, 45(2):246–265, 1998.

[156] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[157] L. Katz. A new status index derived from sociometric index. *Psychometrika*, pages 39–43, 1953.

[158] M. G. Kendall and B. Babington-Smith. On the method of paired comparisons. *Biometrika*, 31(3/4):324–345, 1940.

[159] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[160] S. Khanna, N. Linial, and S. Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000.

[161] H. A. Kierstead. Coloring graphs on-line. In *Online Algorithms*, pages 281–305. Springer, 1998.

[162] H. A. Kierstead and A. V. Kostochka. A short proof of the hajnal-szemerédi theorem on equitable colouring. *Combinatorics, Probability & Computing*, 17(2):265, 2008.

[163] S. Kirkpatrick, D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[164] V. Klee and D. Larman. Diameters of random graphs. *Canadian Journal of Mathematics*, 33:618–640, 1981.

[165] W. Klotz. Graph coloring algorithms. *Mathematics Report*, pages 1–9, 2002.

[166] M. Kochen. *The small world*. Ablex Norwood, NJ, 1989.

[167] A. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Julius Springer, 1933.

[168] A. V. Kostochka. Equitable colorings of outerplanar graphs. *Discrete Mathematics*, 258(1):373–377, 2002.

[169] M. Krivelevich. Coloring random graphsan algorithmic perspective. In *Mathematics and Computer Science II*, pages 175–195. Springer, 2002.

[170] M. Krivelevich and B. Patkós. Equitable coloring of random graphs. *Random Structures & Algorithms*, 35:83–99, 2009.

[171] M. Krivelevich, B. Sudakov, V. H. Vu, and N. C. Wormald. On the probability of independent sets in random graphs. *Random Structures & Algorithms*, 22:1–14, 2003.

[172] M. Krivelevich and V. H. Vu. Approximating the independence number and the chromatic number in expected polynomial time. *Journal of Combinatorial Optimization*, 6(2):143–155, 2002.

[173] Michael Krivelevich. Deciding k-colorability in expected polynomial time. *Information Processing Letters*, 81(1):1–6, 2002.

[174] M. Kubale. *Graph colorings*, volume 352. American Mathematical Society, 2004.

[175] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 57–65. IEEE, 2000.

[176] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Computer networks*, 31(11):1481–1493, 1999.

[177] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Physical Review E*, 84(6):066122, 2011.

[178] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87:198701, 2001.

[179] V. Latora and M. Marchiori. A measure of centrality based on network efficiency. *New Journal of Physics*, 9(6):188, 2007.

[180] K.-W. Lih. The equitable coloring of graphs. *Handbook of combinatorial optimization*, 3:543–566, 1998.

[181] T. Łuczak. Sparse random graphs with a given degree sequence. In *Proceedings of the Symposium on Random Graphs, Poznan*, pages 165–182, 1989.

[182] T. Łuczak. The chromatic number of random graphs. *Combinatorica*, 11:45–54, 1991.

[183] T. Łuczak. Size and connectivity of the k-core of a random graph. *Discrete Mathematics*, 91(1):61–68, 1991.

[184] K. T. Macon, P. J. Mucha, and M. A. Porter. Community structure in the United Nations General Assembly. *Physica A*, 391(1–2):343–361, 2012.

[185] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic press, 1998.

[186] A. V. Mantzaris, D. S. Bassett, N. F. Wymbs, E. Estrada, M. A. Porter, S. T. Grafton P. J. Mucha, and D. J. Higham. Dynamic network centrality summarizes learning in the human brain. *Journal of Complex Networks*, 1(1):83–92, 2013.

[187] Y. Matsuzaka, N. Picard, and P. L. Strick. Skill representation in the primary motor cortex after long-term practice. *Journal of Neurophysiology*, 97:1819–32, 2007.

[188] D. W. Matula. On the complete subgraphs of a random graph. In *Proceedings of the 2nd Chapel Hill Conference on Combinatorial Mathematics and its Applications (Chapel Hill, NC, 1970)*, pages 356–369, 1970.

[189] D. W. Matula. The employee party problem. *Notices AMS*, 19(2), 1972.

[190] D. W. Matula. *The largest clique size in a random graph*. Department of Computer Science, Southern Methodist University, 1976.

[191] D. W. Matula. Expose-and-merge exploration and the chromatic number of random graph. *Combinatorica*, 7(3):275–284, 1987.

[192] D. W. Matula and L. Kučera. An expose and merge algorithm and the chromatic number of a random graph. In *Random Graphs*, pages 175–187. Wiley, New York, 1990.

[193] C. McDiarmid. On the chromatic number of random graphs. *Random Structures & Algorithms*, 1(4):435–442, 1990.

[194] C. McDiarmid and B. Reed. Channel assignment and weighted coloring. *Networks*, 36(2):114–117, 2000.

[195] B. D. McKay and N. C. Wormald. The degree sequence of a random graph. *Random Structures and Algorithms*, 11(2):97–117, 1997.

[196] D. Meunier, R. Lambiotte, A. Fornito, K. D. Ersche, and E. T. Bullmore. Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3(37), 2009.

[197] S. Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[198] B. Mohar. The Laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications*, 2:871–898, 1991.

[199] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

[200] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2–3):161–180, 1995.

[201] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Combinatorics, Probability and Computing*, 7(3):295–305, 1998.

[202] M. Molloy and B. Reed. Graph coloring and the probabilistic method, 2002.

[203] J. Moody and D. R. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, 68(1):103–127, 2003.

[204] P. J. Mucha and M. A. Porter. Communities in multislice voting networks. *Chaos*, 20:041108, 2010.

[205] S. Munoz, M. T. Ortuño, J. Ramırez, and J. Yáñez. Coloring fuzzy graphs. *Omega*, 33(3):211–221, 2005.

[206] R. J. Nemeth and D. A. Smith. International trade and world-system structure: A multiple network analysis. *Review (Fernand Braudel Center)*, 8(4):517–560, 1985.

[207] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004.

[208] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005.

[209] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[210] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103:8577–8582, 2006.

[211] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[212] M. E. J. Newman and M. Girvan. Mixing patterns and community structure in networks. In R. Pastor-Satorras, J. Rubi, and A. Díaz-Guilera, editors, *Statistical Mechanics of Complex Networks*. Springer, 2003.

[213] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[214] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 1):2566–2572, 2002.

[215] M. R. da Silva, H. Ma, and A.-P. Zeng. Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. *Proceedings of the IEEE*, 96:1411–1420, 2008.

[216] A. C. Paauwe. Distance regular graphs. 2007. http://buzzard.ups.edu/courses/2007spring/projects/paauwe-paper-revised.pdf.

[217] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 66, Stanford InfoLab, 1999.

[218] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[219] V. Y. Pan and Z. Q. Chen. The complexity of the matrix eigenproblem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 507–516. ACM, 1999.

[220] C. H. Papadimitriou and K. Steiglitz. 6.1 the max-flow, min-cut theorem. In *Combinatorial Optimization: Algorithms and Complexity*, pages 117–135. Dover, 1998.

[221] E. Pegg Jr. and E. W. Weisstein. Tutte 12-cage. *MathWorld–A Wolfram Web Resource*. http://mathworld.wolfram.com/Tutte12-Cage.html.

[222] S. V. Pemmaraju, K. Nakprasit, and A. V. Kostochka. Equitable colorings with constant number of colors. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 458–459. Society for Industrial and Applied Mathematics, 2003.

[223] C. Piccardi. Finding and testing network communities by lumped Markov chains. *PloS ONE*, 6(11):e27028, 2011.

[224] M. S. Pinsker. On the complexity of a concentrator. In *7th International Telegraffic Conference*, volume 4, pages 1–318, 1973.

[225] B. Pittel and R. S. Weishaar. On-line coloring of sparse random graphs and random trees. *Journal of Algorithms*, 23(1):195–205, 1997.

[226] P. Pons and M. Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

[227] K. T. Poole. Voteview. http://voteview.com, 2013.

[228] K. T. Poole and H. Rosenthal. *Congress: A Political-Economic History of Roll Call Voting*. Oxford University Press, 1997.

[229] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082–1097, 1164–1166, 2009.

[230] R. B. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.

[231] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101:2658–2663, 2004.

[232] J. Reichardt and S. Bornholdt. Detecting fuzzy community structures in complex networks with a Potts model. *Physical Review Letters*, 93(21):218701, 2004.

[233] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74:016110, 2006.

[234] S. A. Rice. The identification of blocs in small political bodies. *The American Political Science Review*, 21(3):619–627, 1927.

[235] W. Richards and A. Seary. Convergence analysis of communication networks. In G. Barnett, editor, *Advances in Communication Science*. Ablex, 1995.

[236] T. Richardson, P. J. Mucha, and M. A. Porter. Spectral tripartitioning of networks. *Physical Review E*, 80:036111, September 2009.

[237] M. P. Rombach and M. A. Porter. Discriminating power of centrality measures. *arXiv:1305.3146*, 2013.

[238] M. P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha. Core-periphery structure in networks. *arXiv:1202.2684*, 2012.

[239] C. Roth, S. M. Kang, M. Batty, and M. Barthelemy. A long-time limit for world subway networks. *Journal of the Royal Society Interface*, In press:(doi: 10.1098/rsif.2012.0259), 2012.

[240] W. G. Roy. The unfolding of the interlocking directorate structure of the United States. *American Sociological Review*, 48(2):248–257, 1983.

[241] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31:581–603, 1966.

[242] G. H. Sasaki and B. Hajek. The time complexity of maximum matching by simulated annealing. *Journal of the ACM (JACM)*, 35(2):387–403, 1988.

[243] R. A. Schmidt and T. D. Lee. *Motor Control and Learning. A Behavioral Emphasis*. Human Kinetics, 4 edition, 2005.

[244] K. Schürger. Limit theorems for complete subgraphs of random graphs. *Periodica Mathematica Hungarica*, 10(1):47–53, 1979.

[245] E. Shamir and J. Spencer. Sharp concentration of the chromatic number on random graphs $g(n, p)$. *Combinatorica*, 7(1):121–129, 1987.

[246] A. Shimbel. Structural parameters of communication networks. *The Bulletin of Mathematical Biophysics*, 15(4):501–507, 1953.

[247] P. E. Simonds. Outcast males and social structure among bonnet macaques (macaca radiata). *American Journal of Physical Anthropology*, 38(2):599–604, 1973.

[248] A. Sinclair. Randomness & computation. University Lecture, `http://www.cs.berkeley.edu/ sinclair/cs271/f11.html`, 2011.

[249] P. J. Slater. On locating a facility to service areas within a network. *Operations Research*, 29(3):523–531, 1981.

[250] B. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 2004.

[251] D. A. Smith and D. R. White. Structure and dynamics of the global economy: Network analysis of international trade 1965–1980. *Social Forces*, 70(4):857–893, 1992.

[252] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. J. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney, R. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J. M. Brady, and P. M. Matthews. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 23(S1):208–219, 2004.

[253] D. Snyder and E. L. Kick. Structural position in the world system and economic growth, 1955-1970: A multiple-network analysis of transnational interactions. *American Journal of Sociology*, 84(5):1096–1126, 1979.

[254] D. de Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27(5):292–306, 1976.

[255] R. Solomonoff and A. Rapoport. Connectivity of random nets. *The Bulletin of Mathematical Biophysics*, 13(2):107–117, 1951.

[256] G. B. Sorkin. Efficient simulated annealing on fractal energy landscapes. *Algorithmica*, 6(1-6):367–418, 1991.

[257] J. Spencer. Asymptotic lower bounds for ramsey functions. *Discrete Mathematics*, 20:69–76, 1977.

[258] D. Spielman and S. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Proceedings of 37th Conference on Foundations of Computer Science*, pages 96–105.

[259] C. J. Stam, G. Nolte, and A. Daffertshofer. Phase lag index: Assessment of functional connectivity from multichannel EEG and meg with diminished bias from common sources. *Human Brain Mapping*, 28:1178–93, 2007.

[260] W. L. Steiger. Some kolmogoroff-type inequalities for bounded random variables. *Biometrika*, 54(3-4):641–647, 1967.

[261] T. Szele. Combinatorial investigations about the directed complete graph (written in hungarian). *Matematikai és Fizikai Lapok*, 50:223–254, 1943.

[262] J. Szymański. On a nonuniform random recursive tree. *North-Holland Mathematics Studies*, 144:297–306, 1987.

[263] J. Tits. Sur la trialité et certains groupes qui s'en déduisent. *Institut des Hautes Études Scientifiques Publications Mathématiques*, 2:14–60, 1959.

[264] F. Tönnies. Gemeinschaft und gesellschaft. *Harvard College Library*, 1887.

[265] A. L. Traud, C. Frost, P. J. Mucha, and M. A. Porter. Visualization of communities in networks. *Chaos*, 19(041104), 2009.

[266] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter. Comparing community structure to characteristics in online collegiate social networks. *Society for Industrial and Applied Mathematics Review*, 53:526–543, 2011.

[267] S. Tsujimoto, A. Genovesio, and S. P. Wise. Evaluating self-generated decisions in frontal pole cortex of monkeys. *Nature Neuroscience*, (1):120–126, 2009.

[268] S. Tsujimoto, A. Genovesio, and S. P. Wise. Frontal pole cortex: Encoding ends at the end of the endbrain. *Trends in Cognitive Sciences*, 15(4):169–176, 2011.

[269] A. Tucker. Perfect graphs and an application to optimizing municipal services. *Siam Review*, 15(3):585–590, 1973.

[270] J. Vandekerckhove. General simulated annealing algorithm. 2008. http://www.mathworks.de/matlabcentral/fileexchange/10548.

[271] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[272] D. J. Watts and S. H. Strogatz. Collective dynamics of small-worldnetworks. *Nature*, 393(6684):440–442, 1998.

[273] A. S. Waugh, L. Pei, J. H. Fowler, P. J. Mucha, and M. A. Porter. Party polarization in Congress: A network science approach. arXiv:0907.3509, 2012.

[274] I. Wegener. *Simulated annealing beats metropolis in combinatorial optimization*. Springer, 2005.

[275] S. Wegener, T. Droste, and I. Jansen. Dynamic parameter control in simple evolutionary algorithms. *Foundations of Genetic Algorithms 2001 (FOGA 6)*, 6:275, 2001.

[276] R. S. Weiss and E. Jacobson. A method for the analysis of the structure of complex organizations. *American Sociological Review*, 20:661—668, 1955.

[277] E. W. Weisstein. Chang graphs. *MathWorld–A Wolfram Web Resource*. http://mathworld.wolfram.com/ChangGraphs.html.

[278] D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, 1985.

[279] H. C. White, S. A. Boorman, and R. L. Breiger. Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology*, 81(4):730–780, 1976.

[280] H. S. Wilf. Backtrack: An $O(1)$ expected time algorithm for the graph coloring problem. *Information Processing Letters*, 18(3):119–121, 1984.

[281] S. M. Williams. *Sylvius*, pyramis studios, inc., 2010. http://www.sylvius.com.

[282] S. W. Williams. Internal colonialism, core-periphery contrasts and devolution: An integrative comment. *Area*, 9, 1977.

[283] C. L. Willis and S. J. McNamee. Social networks of science and patterns of publication in leading sociology journals, 1960 to 1985. *Science Communication*, 11(4):363–381, 1990.

[284] M. W. Woolrich, W. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, and S. M. Smith. Bayesian analysis of neuroimaging data in FSL. *NeuroImage*, 45:S173–S186, 2009.

[285] F. Y. Wu. The Potts model. *Review of Modern Physics*, 54(1):235–268, 1982.

[286] J. Yang and J. Leskovec. Structure and overlaps of communities in networks. arXiv:1205.6228, 2012.

[287] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[288] B. Zelinka. On the number of independent complete subgraphs. *Publ. Math. Debrecen*, 13:95–97, 1966.

[289] Y. Zhang, A. J. Friend, A. L. Traud, M. A. Porter, J. H. Fowler, and P. J. Mucha. Community structure in Congressional cosponsorship networks. *Physica A*, 387:1705–1712, 2008.