Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Monte Carlo Methods on Complex Networks

by

Neal Mc Bride

A thesis presented for the degree of
Doctor of Philosophy

April 6, 2017

Supervised by:

Dr. John Bulava
School of Mathematics
Trinity College Dublin
Dublin 2
Ireland

Dr. Nicola Marchetti
CONNECT Centre
Trinity College Dublin
Dublin 2
Ireland

Declaration of Authorship

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

_____

Neal Mc Bride

Summary

In this thesis, we perform Markov Chain Monte Carlo simulations involving complex networks. The work is separated into three main themes; the ferromagnetic Ising model on complex networks, the antiferromagnetic Potts model on complex networks and rewiring of spanning trees of complete graphs. This work was performed as part of a collaboration between the School of Mathematics and CONNECT (the Science Foundation Ireland Research Centre for Future Networks and Communications) in Trinity College, Dublin.

In the first main part of this work, a combination of the Wolff cluster and Metropolis algorithms are used to sample from the Boltzmann distribution of the ferromagnetic Ising model on complex networks. The complex networks in question represent the interference between basestations in a mobile phone network in Dublin, Ireland. The system exhibits a crossover between an ordered phase at low temperature to a disordered phase at high temperature as signalled by the loss of magnetisation. We find that the critical behaviour of this model resembles that of the Watts-Strogatz model. The higher the level of interconnectedness between nodes in the complex network, the less sharp the transition from ordered to disordered phases and the higher the critical temperature. The critical point is determined using the peaks of the specific heat and susceptibility.

The antiferromagnetic q-state Potts model on complex networks is studied using the Wang-Landau multicanonical Monte Carlo method in the second part of this thesis. The Potts model energy is used to represent interference in a wireless network with q orthogonal broadcasting channels. The density of states is estimated for $q = 2$, 3, 4, 5 & 6 and the interference value which bounds 90% of spin configurations is calculated. This critical interference value falls as q increases and can be thought of as the interference value below which the system lies 90% of the time if the spin values are uniformly distributed. Additionally, the estimate of the density of states allows the direct calculation of thermodynamic observables of the Potts model. A crossover between a high temperature and low temperature phase is found to occur.

The temperature at which this happens decreases as q increases. However the error in the density of states estimate is too high to extract precision calculations of this critical temperature.

In the third part of this thesis, we introduce a new graph rewiring algorithm which samples from the space of spanning trees with N nodes and $N-1$ edges. The probability distribution of graphs of this kind is shown to depend on the symmetries of these graphs. We demonstrate that the algorithm is ergodic and proceed to estimate the probability distribution for small graph ensembles with exactly known probabilities, as a test of our algorithm. The autocorrelation time of the graph diameter observable demonstrates that the algorithm generates independent configurations efficiently as the system size increases. Finally, the mean graph diameter is estimated for spanning trees of sizes ranging over three orders of magnitude. The mean graph diameter results agree very closely with the analytical values.

Acknowledgements

First of all, I would like to thank my supervisors Dr. John Bulava and Dr. Nicola Marchetti. This work would never have been possible without their constant guidance, inspiration, support and willingness to test the water with both feet.

I am very grateful to Dr. Linda Doyle and Dr. Sinéad Ryan, for originally building the collaboration between the School of Mathematics and CTVR, the precursor to CONNECT.

I would like to thank Dr. Irene Macaluso and Dr. Mike Peardon for their invaluable contributions over the last three years, especially in the beginning of this project while we were still finding our way.

During the course of this work, I have been fortunate enough to be involved in two outstanding research groups; the Lattice QCD group in the School of Mathematics and the Wireless Networks group in CONNECT, both in Trinity College Dublin. There are far too many people to name individually, but thanks to everyone involved in these groups for fostering such a welcoming and productive environment.

I would like to thank my fellow postgraduate students and in particular my office-mates, Ben Hörz and Cian O'Hara for their friendship and help over the last few years. They were instrumental to my personal development and enjoyment of being a Ph.D. student and without them my juggling skills would be severely lacking.

Finally, the biggest thanks go to my family and friends for their unending love, support and encouragement over the last three years.

Calculations for this work were performed on the Lonsdale cluster maintained by Research IT in Trinity College Dublin. This cluster was funded through grants from Science Foundation Ireland. This work is supported by Science Foundation Ireland through the CTVR and CONNECT grants 10/CE/i1853 and 13/RC/2077, respectively.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis we use Markov Chain Monte Carlo methods to study problems on complex networks. Complex networks are a general method used to represent the connections between a set of objects with the aim of gaining some insight which would otherwise be obscured in the complexity. Classical examples of complex networks include the Internet, cellular networks, power grids, social networks and chemical reactions in the cell [1].

This work is the result of a collaboration between the School of Mathematics, Trinity College Dublin and CONNECT (The Science Foundation Ireland Research Centre for Future Networks and Communications), formed to study complex networks which arise in mobile telecommunication networks using a statistical physics approach. This thesis contains three main parts, each contained within a chapter. Of the following three chapters, the first two treat complex networks as a substrate for a dynamical process. The third involves the behaviour of the networks themselves. Chapters 2 and 3 are part of the work that stems directly from the program at CONNECT to study the properties of mobile networks as complex systems. The goal of that program is to contribute to the design of networks which perform consistently and optimally with a minimal amount of planning, coordination, and human intervention. Chapter 4 is motivated by the need to describe networks that can change over time.

In Chapter 2, the topic of the ferromagnetic Ising model on complex networks is studied. This flavour of the Ising model favours ordered states at low temperatures and disordered states at higher temperatures. The crossover between these phases and the temperature at which it occurs is shown to depend on the nature of the complex network. A greater amount of interconnections in the network cause the critical behaviour to becomes less sharp and the ordered phase to persist to much higher

temperatures. This behaviour is compared with some exact and numerical results on other well-known complex networks. To study this system, a combination of local and global spin flip algorithms is used. The Metropolis single spin flip algorithm on its own is found to break down on these complex networks. We show that this failure is due to the heterogeneous environment of each spin. The failure stems from exploring only a small region of the state space of system configurations which is visually represented in Fig. 2.3. When used in combination, we demonstrate that the Metropolis and Wolff algorithm can efficiently sample the set of spin configurations. The critical temperature which separates the ordered and disordered phase is estimated for the three complex networks by calculating thermodynamic observables which have a peak at the critical temperature.

In Chapter 3, the antiferromagnetic Potts model on complex networks is the subject of investigation. In our early work on this topic, we found that both local and cluster spin-flip algorithms fail to effectively sample the space of spin configurations of this model, especially at low temperature. This drew our attention to the Wang-Landau algorithm [2]; a multicanonical Monte Carlo method. By estimating the density of states independently of the temperature, we avoid the problems associated with geometric frustration and random edges. The antiferromagnetic q-state Potts model is then used to model interference in wireless networks using q orthogonal frequency channels. In a dynamic, distributed spectrum allocation model the density of states allows us to calculate interference values below which the system lies 90% of the time. Furthermore, with access to the partition function, all thermodynamic potentials of the system can be estimated. In particular, the peaks in the specific heat can be used to indicate the temperature at which the system crosses over from a disordered phase to a quenched state. This critical temperature is shown to decrease as q increases.

In Chapter 4, we present a new Monte Carlo algorithm which samples the space of graphs with a given number of nodes and edges. To the best of our knowledge, this is the first algorithm of its kind and we anticipate that it will be very useful to study graph ensembles in complex networks. We show that the probability of a given graph is proportional to its symmetry and even go as far as exactly calculating the graph probability distribution for small ensembles. In order to demonstrate that our algorithm can perform this kind of sampling, we first show that it is ergodic. Next, we numerically demonstrate that it samples independent graphs efficiently, even for ensembles of large graphs. Finally, the mean diameter of these graph ensembles is estimated and compared with an analytical model. Our algorithm produces results which compare well with the exact values. Finally, we close this thesis in Chapter 5

with some concluding remarks.

Chapter 2

Ferromagnetic Ising Model

This chapter focuses on the behaviour of the Ising model on complex networks which have been generated during a study of interference in wireless mobile phone networks in Dublin, Ireland [3]. The Ising model is a simple mathematical model with many degrees of freedom which can exhibit an order-disorder phase transition. The nature of this phase transition depends both on the type of lattice and the number of spatial dimensions of this lattice. It should be of no surprise then that embedding the Ising and Potts models on complex networks gives rise to some unique and unexpected behaviours. Using Monte Carlo methods, we study this critical behaviour and the effect of the network substrate on the Ising model. An excellent review of the Potts and Ising models on regular lattices can be found in Ref. [4] and we will often refer to this review when comparing lattice and complex network differences and similarities.

Over the last two decades, complex networks have been extensively studied using exact methods, mean-field approximations and Monte Carlo simulations [5]. Both network features themselves and their effect on dynamic models are extremely interesting. Some canonical examples of complex networks which have been extensively studied include small-world networks, scale-free networks and random graphs. Due to the empirical nature of our interference graphs, we draw comparisons between their structure and critical behaviour and that of more well-known complex networks. Of particular interest is the observation of ordered and disordered states of the system driven by temperature. The temperature dependence of this transition, the efficiency of Monte Carlo simulations near this temperature and the failure of some of these Monte Carlo methods will be studied in detail here.

This chapter contains four main sections. In Sec. 2.1, we provide some background on the Potts and Ising models and complex networks. Sec. 2.2 lays out the technology

that we use to perform Monte Carlo simulations and to perform a statistical analysis of the results. Sec. 2.3 discusses the results of these Monte Carlo simulations and this chapter ends with our conclusions and suggestions for future work in Sec. 2.4.

2.1 Background

The Ising model on complex interference networks is the topic of this chapter and the Potts model is studied in the next. The following section introduces the more general Potts model and discusses the case where the Ising model is equivalent to the Potts model. Complex networks are described using graph theory. The notation introduced in this section will be used throughout this thesis. Furthermore, a discussion of the structure and behaviour of canonical network models is discussed. Our interference networks are then introduced and compared to these canonical networks.

2.1.1 Potts and Ising Model

The Potts model is typically defined as a system of interacting spins which are placed at the vertices of a lattice. A spin configuration, σ, is the result of each spin in the system taking a value $\sigma_i \in [1, q]$. The energy of a configuration in the standard Potts model is

$$E(\sigma) = -J \sum_{\langle i,j \rangle} \delta_{\sigma_i \sigma_j} + h \sum_i \delta_{\sigma_i q}, \tag{2.1}$$

where $\langle i, j \rangle$ denotes a sum over the nearest neighbours (or neighbourhood) of each site i and the external field strength h attempts to align spins in the q direction. For the rest of this work, we consider the zero-field Potts model and the second term in Eq. 2.1 vanishes. The coupling constant J, which determines the strength of the interaction between neighbouring spins, distinguishes two flavours of the model:

$$\text{(Ferromagnetic)} \qquad J > 0 \quad \text{and} \tag{2.2}$$

$$\text{(Antiferromagnetic)} \qquad J < 0. \tag{2.3}$$

The spin configurations are Boltzmann distributed with probabilities,

$$P(\sigma, \beta) = Z^{-1} \exp\left[-\beta E(\sigma)\right], \tag{2.4}$$

where $\beta = 1/k_B T$ which is known as the inverse temperature. For convenience, we choose a system of units such that the Boltzmann constant k_B is equal to one. The partition function,

$$Z(\beta) = \sum_{\{\sigma\}} \exp\left[-\beta E(\sigma)\right], \tag{2.5}$$

is of utmost importance. With perfect knowledge of the partition function the model can be solved exactly. The Ising model is equivalent to the $q = 2$ Potts model, in which each spin can take one of two possible values, $\sigma_i \in [-1, +1]$. These models have identical critical behaviour once the interaction strength is related by,

$$2J_{\text{Ising}} = J_{\text{Potts}}. \tag{2.6}$$

On a lattice, the Potts model exhibits a phase transition driven by temperature from an ordered, low temperature phase to a disordered, higher temperature phase. A phenomenon known as spontaneous symmetry breaking occurs below the critical Curie temperature (T_c) and is associated with a decrease in the entropy of the system. Depending on the spatial dimension and q, a phase transition in the Potts model can either be discontinuous (first order) or continuous (second order).

All thermodynamic potentials can be expressed as derivatives of the Helmholtz free energy,

$$F = -k_B T \log Z. \tag{2.7}$$

The internal energy of the system is given by,

$$\langle E \rangle_\beta = \frac{\partial(\beta F)}{\partial \beta} = Z^{-1} \sum_{\{\sigma\}} E(\sigma) \exp\left[-\beta E(\sigma)\right] \tag{2.8}$$

and the average magnetisation is defined as,

$$\langle M \rangle_\beta = -\frac{\partial F}{\partial h}\Big|_{h=0} = Z^{-1} \sum_{\{\sigma\}} M(\sigma) \exp\left[-\beta E(\sigma)\right], \tag{2.9}$$

where the magnetisation is the sum over the volume of spins,

$$M(\sigma) = \sum_{i=1}^{V} \sigma_i. \tag{2.10}$$

In the thermodynamic limit, the magnetisation of the ferromagnetic Potts model is an indicator of an ordered or disordered system. At low temperature, the spins align in an ordered fashion and the magnetisation tends to infinity. Above the Curie temperature, the magnetisation vanishes as the system becomes disordered, with spins pointing in each direction with uniform probability.

A discontinuous phase transition emits or absorbs a large amount of latent heat which results in a discontinuity in the internal energy of the system in infinite volume. The absolute value of the magnetisation is also discontinuous and falls from one to zero around the critical point. These first derivatives of F are discontinuous in first order phase transitions.

In a continuous phase transition, the second derivatives of F are discontinuous. In the Potts model, these derivatives are known as the magnetic susceptibility

$$\chi = -\frac{\partial^2 F}{\partial h^2}\Big|_{h=0} = \frac{\partial M}{\partial h} = \frac{1}{T}\left(\langle M^2\rangle - \langle M\rangle^2\right), \tag{2.11}$$

and the specific heat,

$$C_v = \frac{\partial \langle E\rangle}{\partial T} = \frac{1}{T^2}\left(\langle E^2\rangle - \langle E\rangle^2\right). \tag{2.12}$$

In finite volume these discontinuities can be smeared out and appear continuous. The susceptibilities do not diverge, however a maximum is observed around the critical point in both first and second order phase transitions. In both Monte Carlo simulations and exact solutions, the critical point can be identified where these susceptibilities diverge. However, the temperature at which the susceptibilities peak can be shifted from their infinite volume values and the peaks themselves appear more rounded in finite volume.

Intuitively this smoothing out of the divergent functions in finite volume can be understood as follows. The partition function is a sum of exponentials depending on $\exp[-\beta E]$. The exponential function is analytic everywhere as a function of β. The sum of analytic functions is an analytic function. Analytic real functions are infinitely differentiable, thus the derivatives (susceptibilities) are also analytic (not divergent). In the thermodynamic limit, the infinite sum over analytic functions can diverge which leads to a phase transition.

2.1.2 Graph Notation

This section introduces the graph theory notation necessary to define the Potts model in a more general way. Instead of placing spins at lattice sites, we use the sites of a graph, G, which allows us to study the model on a complex network. Since graph theory notation tends to differ between sources, we use the conventions from Ref. [6].

A graph is an object consisting of a vertex set, V, and an edge set, E. The elements of each set are called vertices (or nodes) and edges (or links) respectively. The number of elements in each set is given as $|V|$ and $|E|$. If an edge $\{u, v\}$ exists, then vertices u and v are adjacent. The edge $\{u, v\}$ is also incident to both vertices u and v. For the purpose of this thesis, the ordering of the vertices in the edge $\{u, v\}$ does not hold any special meaning since the edges in our graphs are undirected and each edge holds the same weight.

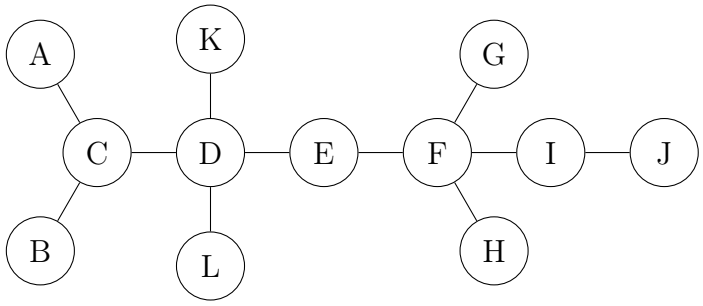


Figure 2.1: An example of a graph diagram representing a spanning tree. The edge $\{A, C\}$ connects vertices A and C. The edges of this graph are unweighted and undirected. This graph is connected since there is a walk between all vertices. Graphs of this type will be studied further in Chapter 4.

A walk or path is a sequence of vertices, $a_1\ a_2\ a_3\ \ldots\ a_n$, with an edge joining adjacent elements of the sequence. A connected graph has a walk which joins all pairs of vertices. All graphs investigated in this work are undirected, unweighted and connected. A tree is an undirected graph which contains a unique path between any two vertices. The diameter of a graph, $d(G)$, is the maximum pairwise distance between any two vertices. In a tree, the diameter is the longest walk of unique vertices. The average path length, $\langle l \rangle$ is the average number of steps along the shortest path for all possible pairs of nodes.

Any two graphs are considered equal if they have equal edge sets and vertex sets. However a stronger idea of equality, isomorphism, exists between two graphs with the same structure regardless of vertex labelling. Two graphs are isomorphic if there exists a one-to-one correspondence having the property that whenever two vertices

are adjacent in either graph, the corresponding two vertices are adjacent in the other graph. In practice an isomorphism preserves the adjacency and non-adjacency in the graphs. This is discussed further in Chapter 4.

A graph H is a subgraph of a graph G if the vertex and edge set of H are subsets of the vertex and edge sets of G. The neighbourhood, $N(u)$ of a vertex u is the subgraph induced by all vertices adjacent to u. The degree $k(u)$ of a vertex u is the number of edges incident to u and depends on the size of its neighbourhood. Consequently, the degree distribution is the probability of a vertex having a given degree.

The local clustering coefficient, C_i, is defined as the fraction of neighbours of a given node which themselves are interconnected over the number of all possible edges between these neighbours. The overall clustering coefficient, C, is the mean over all local coefficients;

$$C = \frac{1}{|V|} \sum_{i=1}^{|V|} C_i \tag{2.13}$$

A clustering coefficient, $C = 1$, describes a graph in which every node is connected to every other node. On the other hand, $C = 0$ describes a graph such as the square lattice.

2.1.3 Complex Networks

While no formal definition currently exists, a working definition for a complex network is as follows: a complex network is a graph that describes the set of connections in systems with many constituents which may interact with each other. Both real-world systems and appropriate graph models can display complex network behaviour. Real-world complex networks include chemical reactions in a cell, the internet of computers, the World Wide Web of hyper-linked websites and friends in social networks. From the empirical study of these large real-world networks, some common features unique to complex networks have been identified; small-world behaviour, clustering of nodes and power-law degree distributions [1].

Two canonical complex network models are the Watts-Strogatz (WS) [7] and Barabási-Albert (BA) [1] models. The WS model generates complex networks from regular lattices with a fraction p of the total number of edges randomly rewired. To generate these networks;

- start with a set of $|V|$ nodes in a ring, each with K neighbours.

- Each node is connected to the $K/2$ nodes immediately to its right and left on the ring.
- Iterate over each node, rewiring each of their edges with probability p to a random neighbour chosen with uniform probability (ensuring to avoid self-connections and duplicate edges.).

Fully ordered ring lattices remain after this process using $p = 0$ and completely random graphs result using $p = 1$. Some very interesting behaviour occurs for intermediate values of p. For low p, the average path length scales linearly with the number of nodes as you would expect in a lattice, $\langle l \rangle \sim |V|$. At higher p, the average path length starts to grow like $\langle l \rangle \sim \ln |V|$. The average path length does not increase significantly with system size. In a WS network, the mean degree of a network generated from a lattice with coordination number K is $\langle k \rangle = K$ which is independent of p. With $p = 0$, the degree distribution is a delta function and for $p = 1$ the degree is Poisson distributed.

The clustering coefficient depends strongly on the rewiring probability p as

$$C(p) \sim C(0)(1 - p)^3. \tag{2.14}$$

The WS model exhibits the small-world property where most nodes are not neighbours of each other but any node is only ever a few hops away from any other node. This is a fundamental feature of complex networks. A small average path length $\langle l \rangle$ and a large clustering coefficient are necessary to achieve the small-world property.

Although displaying the small-world phenomenon, the degree distribution of the WS model does not match that of many real-world complex networks. Empirically many large complex networks are scale-free. That is to say that their degree distribution follows a power-law,

$$P(k) \sim k^{-\gamma}. \tag{2.15}$$

The BA model produces complex networks with power-law degree distribution as a result of assembling a network using the principles of growth and preferential attachment. This means that nodes are added one-by-one to the graph during construction rather than being present from the start. Also, the probability of attaching new nodes to the existing graph depends on the degree of each node. Nodes with a larger degree have a higher probability of being connected to than low degree nodes. To generate a BA model graph:

- start with a small core of (m_0) nodes connected to each other.

- At each growth step, add a new node to the graph with $m \leq m_0$ edges to older nodes.

- The probability of attaching each new edge to an existing node, i, of degree k_i at each growth step is,

$$\Pi_i(k_i) = \frac{k_i}{\sum_{j=1}^{|V|-1} k_j}. \tag{2.16}$$

The sum in Eq. 2.16 is over the degree of all nodes in the graph excluding the new node being added in this round of growth. This method of growth and preferential attachment results in a network which is scale-free, with exponent $\gamma = 3$. These scale-free networks also exhibit the small-world property, with the average path length $\langle l \rangle \sim \ln |V|$. The clustering coefficient is lower than the WS model at $C \sim |V|^{-0.75}$, however this decay is slower than that of a random graph $C = \langle k \rangle |V|^{-1}$.

The behaviour of spin models on complex networks has been intensely studied over the last two decades. An excellent review of critical phenomena of complex networks is available in Ref. [5]. Here we summarise some results relating to the BA and WS models to later compare with our Monte Carlo results on complex interference networks.

The increased heterogeneity of the nodes and edges in complex networks changes the critical behaviour when compared with a lattice. The ferromagnetic phase transition becomes less sharp and the critical temperature is increased. At high temperature, far from criticality, the thermodynamic quantities of WS networks are close to those of the starting lattice. Near the critical temperature, the mean-field approximation holds. This should not come as a surprise since complex networks are high dimensional structures with dimension that far exceeds the critical dimension above which the mean-field approximation is obeyed. An exact solution of the Ising model in small-world networks is presented in Ref. [8] using a 1-dimensional ring Ising model with random long range bonds, similar to the WS model. The critical temperature is found to scale with rewiring probability p,

$$T_c(p) \sim J/|\ln p|. \tag{2.17}$$

Their solution confirms the mean-field character of the transition. Furthermore, they found that the specific heat has a peak at criticality but remains finite. In the disordered

phase, near criticality, the specific heat scales like,

$$C_v \propto \left(\frac{T - T_c}{T_c}\right)^{-\alpha}, \tag{2.18}$$

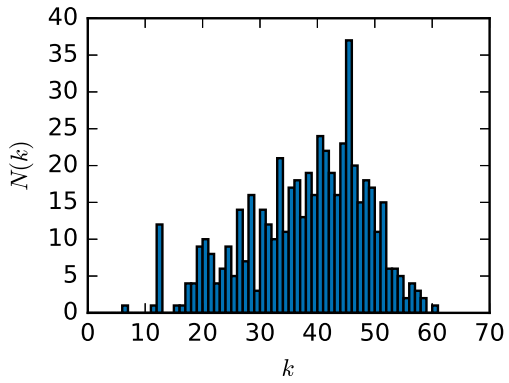
where, T, is the temperature of the system and, α, the critical exponent of specific heat. The non-divergent specific heat is a result of the mean-field Ising model predicting that in four spatial dimensions or greater, $\alpha = 0$.

A further important contribution on small-world network models is Ref. [9]. With $p < 1$, a transition occurs between the high temperature region dominated by the one dimensional structure and the low temperature mean-field region. This work also confirms the existence of a finite temperature ferromagnetic phase transition. Their methods involved both analytical approximation using the replica trick and a Monte Carlo simulation using the Swendsen and Wang [10] cluster algorithm.

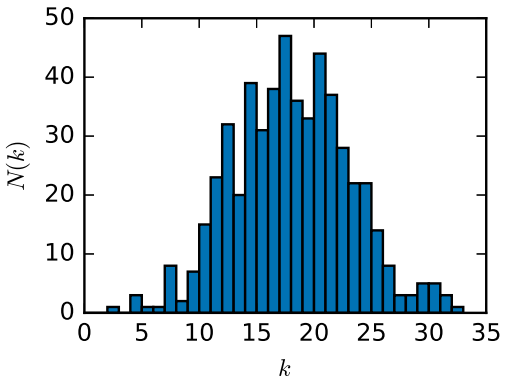
The exact critical behaviour of the ferromagnetic Ising model in scale-free graphs with arbitrary degree distribution was found in Refs. [11] and [12]. In networks with $\gamma = 3$, the critical temperature is infinite and the system is ferromagnetic at all temperatures. A Monte Carlo study of the ferromagnetic phase transition of these scale-free networks was performed in Ref. [13]. Using the heat-bath Metropolis algorithm, they found that on a scale-free network with $\gamma = 3$, the critical temperature $T_c \sim \ln|V|$. This confirms the findings of the exact studies which claim that no phase transition occurs in the thermodynamic limit and the system remains in the ordered phase. Furthermore, using a mean-field approximation in Ref. [14], the critical temperature is again found to scale like $T_c \sim \ln|V|$.

2.1.4 Interference Networks

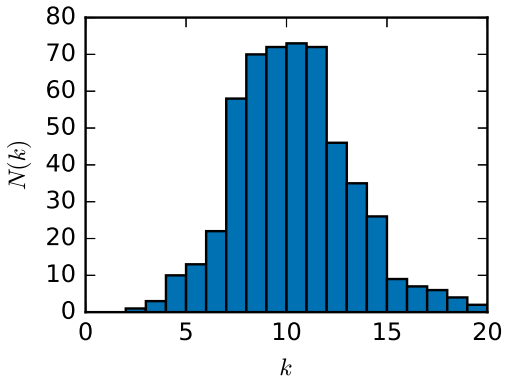
Having discussed the structural features of both the WS and BA model networks, here we describe the interference graphs which we perform our Monte Carlo simulation on. An interference graph is a common representation of interference in wireless networks, typically used to describe interference in channel allocation schemes. Usually, each transmitter/receiver corresponds to a vertex in the interference graph, and an edge connects two vertices if and only if the two cannot use the same channel simultaneously without degrading the signal quality below a given threshold.



(a) This sub-figure shows the number of vertices in the $\theta = 1\%$ interference-graph with k nearest neighbours (degree k). The $\theta = 1\%$ graph contains $|V| = 532$ vertices and $|E| = 10009$ edges.



(b) This sub-figure shows the number of vertices in the $\theta = 3\%$ interference-graph with k nearest neighbours (degree k). The $\theta = 3\%$ graph contains $|V| = 532$ vertices and $|E| = 4675$ edges.



(c) This sub-figure shows the number of vertices in the $\theta = 5\%$ interference-graph with k nearest neighbours (degree k). The $\theta = 5\%$ graph contains $|V| = 529$ vertices and $|E| = 2607$ edges.

Figure 2.2: This figure shows the distribution of the number of vertices with k nearest neighbours (degree k) in the three interference-networks that we will use both in this and the next chapter. In the interference-networks, the vertices represent mobile phone basestations in Dublin city centre and the edges connect vertices which can potentially cause interference to neighbouring basestations. In these networks, a pair of basestations are connected if a fraction, θ, of the users of either basestation experiences interference from the other basestation. This results in nodes with higher degree in the $\theta = 1\%$ network than the $\theta = 5\%$ network, since causing interference for 5% of the users means that 1% of the users must also experience interference, but the inverse is not in general true.

The networks studied in this work were constructed in Ref. [3] by considering the radiation pattern from mobile phone base-stations in the city centre of Dublin, Ireland.

A threshold percentage, θ, of users which may experience high levels of interference before considering two base-stations to interfere with each other is chosen. This value of θ is constant for each interference graph.

The degree distributions of the θ-networks are shown in Fig. 2.2. The number of edges decrease as the threshold (θ) increases. The mean and variance of the degree distributions also decrease as θ increases. All three graphs are connected components, therefore there exists a finite walk between all vertices. To make sure that these graphs are connected, the $\theta = 5\%$ graph has three less vertices than the others. These missing vertices are located at the edge of the physical area of interest. They do not meet the condition that 5% of their users experience a high level of interference from other base-stations. Very few edges connect these three vertices to the rest of the graph in the $\theta = 1\%$ and 3% graphs. Therefore, omitting these vertices from the $\theta = 5\%$ graph does not greatly affect our results.

The average path length and clustering coefficient of each of these graphs is shown in Tab. 2.1. The global clustering coefficient is roughly constant over each of the graphs with an average 60% of the neighbours of each node being interconnected. The average path length however decreases as θ decreases. This behaviour is anticipated, since a low value of θ allows a greater number of connections between adjacent vertices.

$\boldsymbol{\theta}$	**1**%	**3**%	**5**%
C	0.596	0.609	0.624
$\langle l \rangle$	3.8	6.2	9.7

Table 2.1: Clustering coefficient (C) and average path length $\langle l \rangle$ of the θ interference networks. The clustering coefficient of each graph is quite high and the average path length is small compared to the number of nodes.

Due to the high level of clustering and small average path length, the θ interference graphs seem to possess small-world behaviour and we expect the critical behaviour of the Ising model on these graphs to be qualitatively similar to that of the WS model.

2.2 Monte Carlo Methods and Statistical Analysis

In this section, we introduce the methods necessary to perform a high quality Monte Carlo investigation into the ferromagnetic Ising model on complex networks. The aim of this section is to present to the reader a high-level discussion of the procedure we

found to be necessary to produce reliable results.

2.2.1 Markov Chain Monte Carlo

The integrals in Sec. 2.1.1 depend on 'many degrees of freedom, far from the perturbative regime' [15]. Problems of this kind are natural candidates for treatment using Markov Chain Monte Carlo, since the errors in numerical integration techniques scale poorly in large dimensions. Also, exact solutions of empirical complex networks are often unworkable.

Markov Chain Monte Carlo involves a configuration space S and a probability measure of these states, π. A Markov process is used which generates configurations from S with π as its unique, universal stationary distribution. Consecutive configurations form a Markov chain with a transition probability,

$$p_{xy} = P(X_t = x | X_{t+1} = y), \tag{2.19}$$

between two states ($X_t, X_{t+1} \in S$). Ideally, each successive state depends only on its predecessor. A probability measure, $\pi = \{\pi_x\}_{x \in S}$ is called a stationary or equilibrium distribution in the case

$$\sum_x \pi_x p_{xy} = \pi_y. \tag{2.20}$$

For an irreducible Markov chain, if a stationary distribution π exists then it is unique. If the Markov chain is also aperiodic with a finite state space it is ergodic. An ergodic (irreducible, aperiodic and positive-recurrent) Markov chain has a limiting distribution

$$\lim_{n \to \infty} p_{xy}^{(n)} = \pi_y. \tag{2.21}$$

If we consider some function $f = \{f(X_t)\}$ on the state space S, which is square-integrable with respect to π, then it has a mean,

$$\mu_f \equiv \langle f \rangle = \sum_x \pi_x f(x). \tag{2.22}$$

In practice in Markov Chain Monte Carlo simulations, we start with an initial configuration of our system. This is often a high temperature configuration of random spins with uniform probability. We perform a thermalisation phase, in which many configurations X_i are generated using our Markov process. Any functions of these

thermalisation configurations are either thrown away or kept to later estimate the number of steps necessary to equilibrate the system.

After thermalisation, we use the Markov chain configurations to estimate ensemble averages, such as the estimate for mean energy $\tilde{E}$ in the measurement phase,

$$\langle E \rangle \approx \tilde{E} = \sum_{i=1}^{N} E(X_i), \tag{2.23}$$

where N is the number of configuration samples. For efficiency, the configurations themselves are not stored for the Potts and Ising models, but the functions calculated from these configurations are.

2.2.2 Autocorrelation

Successive configurations X_i generated by a Monte Carlo algorithm are correlated, which effectively increases the variance of our estimated averages. An estimator of a mean, $\bar{f}$ is defined as

$$\bar{f} = \frac{1}{N} \sum_{i=1}^{N} f_i. \tag{2.24}$$

The naive unbiased estimate for the variance of uncorrelated samples on this mean, σ^2_{naive}, is related to the variance of the correlated samples by the integrated autocorrelation,

$$\tau_{\text{int}} \approx \frac{\sigma^2(\bar{f})}{\sigma^2_{\text{naive}}(\bar{f})}. \tag{2.25}$$

Having performed a Monte Carlo simulation, it is possible to statistically estimate the amount of correlation between successive samples in order to more precisely estimate the error in our means. The autocovariance between samples with a separation of t Monte Carlo steps is

$$C_{ff}(t) = \langle (f_s - \bar{f})(f_{s+t} - \bar{f}) \rangle \tag{2.26}$$

The normalised autocorrelation function is given by

$$\rho(t) = \frac{C_{ff}(t)}{C_{ff}(0)}. \tag{2.27}$$

The integrated autocorrelation time can be estimated by

$$\tau_{\text{int}} \approx \frac{1}{2} + \sum_{t=1}^{\infty} \rho(t). \tag{2.28}$$

In practice, the summation in Eq. 2.28 can be performed over a finite t-window, since $\rho(t)$ tends towards zero for $t > \tau_{\text{int}}$ and therefore summing any further predominantly adds noise. We must choose t^* as the size of the window of summation carefully. If we choose $t^* > \tau_{\text{int}}$, we introduce little bias by truncating the sum in Eq. 2.28. The value of t^* should also be small enough so as not to unnecessarily amplify the variance of τ_{int} in Eq. 2.29. Under this constraint, the truncated sum in Eq. 2.28 tends to a plateau. The value of τ_{int} at this plateau is taken to estimate the value in Eq. 2.25. and its error is calculated using the covariance of Eq. 2.26, which after some algebraic manipulation is given by,

$$\text{var}(\tau_{\text{int}}) \approx \frac{2(2t^* + 1)}{N} \tau_{\text{int}}^2. \tag{2.29}$$

Standard errors in means estimated using Monte Carlo will usually be of the order of $(\tau/N)^{\frac{1}{2}}$, therefore we aim to run simulations for at least 10000τ to ensure an accuracy in the error bars of around 1%. The definition of integrated autocorrelation times and how to treat the relaxation of our system to equilibrium during the Monte Carlo simulation is taken from Ref. [15]. This is an excellent guide for anyone looking to perform Monte Carlo simulations. We estimate τ_{int} using a modification of the MATLAB script provided in Ref. [16]. Another excellent resource for the statistical analysis of Monte Carlo data is Ref. [17], which takes a more data oriented approach to controlling the errors caused by autocorrelation.

2.2.3 Metropolis Algorithm

We use a combination of the Metropolis and Wolff cluster algorithms to perform our Monte Carlo simulations. These algorithms have very different motivations but both generate Markov chains of Ising states with a limiting distribution that approaches the unique, universal stationary distribution of this ensemble. Both the Metropolis and Wolff cluster algorithms are very well known in the field of statistical physics. However, in Chapter 4.3.1 we propose a new Monte Carlo algorithm of our own and in light of this the formulation of the Metropolis and Wolff algorithms are both interesting and of relevance.

The Metropolis algorithm is a local update algorithm in that small changes are made to each configuration in a proposal step that must then be accepted or rejected in a further step. The algorithm itself is motivated by the detailed balance condition, which is stronger than that in Eq. 2.20 and is sufficient to satisfy ergodicity. The detailed balance condition is shown in Eq. 2.30, where the stationary distribution (π) of a given configuration (x) and the transition probability to a new state p_{xy} is balanced by the reverse,

$$\pi_x p_{xy} = \pi_y p_{yx}. \tag{2.30}$$

The ratio of transition probabilities is proportional to the ratio of stationary distribution,

$$\frac{p_{xy}}{p_{yx}} = \frac{\pi_y}{\pi_x}, \tag{2.31}$$

and the transition probabilities can be factorised into a proposal (g) and acceptance step (A),

$$p_{xy} = g_{xy} A_{xy}. \tag{2.32}$$

The ratio of the stationary distribution of states is the most important factor here. Accept states with the correct relative probability,

$$\frac{A_{xy}}{A_{yx}} = \frac{\pi_y}{\pi_x} \frac{g_{yx}}{g_{xy}}. \tag{2.33}$$

To satisfy Eq. 2.33, we can choose the probability of accepting a new Ising state y given some change to state x to be,

$$A_{xy} = \min\left(1, \frac{\pi_y}{\pi_x} \frac{g_{yx}}{g_{xy}}\right). \tag{2.34}$$

In the case of the Ising model, the proposal probability from state, x to y and back is symmetric as they differ by only a single spin flip which leads to,

$$g_{yx} = g_{xy}. \tag{2.35}$$

The acceptance probability is then reduced to the simpler form,

$$A_{xy} = \min\left(1, \frac{\pi_y}{\pi_x}\right) \tag{2.36}$$

For the Ising model in particular, the Metropolis algorithm is useful when we can calculate the energy of a given configuration, σ. We can numerically calculate the numerator of the Ising probability distribution, but the partition function denominator still eludes us,

$$\pi(\sigma, \beta) = Z^{-1} \exp[-\beta E(\sigma)], \tag{2.37}$$

with

$$Z = \sum_{\{\sigma\}} \exp[-\beta E(\sigma)]. \tag{2.38}$$

However, the Metropolis algorithm is only concerned with the ratio of the stationary distribution,

$$A_{xy} = \frac{\pi_y}{\pi_x} = \exp[-\beta\,(\Delta E)], \tag{2.39}$$

where,

$$\Delta E = E(y) - E(x). \tag{2.40}$$

Due to the local spin flip nature of the algorithm, the change in energy (ΔE) depends only on the interaction of the spin being flipped and its nearest neighbours. All other contributions to the energy cancel out,

$$\Delta E = -J\left(\sum_{j\in N(i)} \delta_{y_i\sigma_j} - \sum_{j\in N(i)} \delta_{x_i\sigma_j}\right) \tag{2.41}$$

The simplicity of the Metropolis algorithm is its main strength. Unfortunately it suffers from a phenomenon known as critical slowing down near a second order phase transition. In this situation, the correlation length between individual spins diverges, leading to an extremely high autocorrelation between samples and very large errors.

In addition to this, we have found that the Metropolis algorithm suffers from more

problems on complex networks. Due to the heterogeneous nature of the degree of each node in a complex network, the difference in energies between successive states in Eq. 2.41 separated by spin flips can be very large. This leads to some spins being flipped very infrequently and a loss in ergodicity in the system that is no longer irreducible since not all states can be accessed. Therefore, we have found that our Monte Carlo simulations benefit from a combination of Wolff cluster and Metropolis updates.

2.2.4 Wolff Algorithm

The Wolff cluster algorithm [18] has been shown to improve the effects of critical slowing down, with autocorrelation times as low as one or two steps per spin for the two dimensional $O(n)$ models, including the Ising model.

The Wolff cluster algorithm is an adaptation of the earlier Swendsen and Wang cluster algorithm [10] which iterates over each spin in the lattice, clustering sites with equal spin values into existing clusters or new clusters. The Wolff adaptation creates only one cluster in the graph. The edge, $\{x, y\}$ is activated to add node y into the cluster with probability,

$$P(\{x, y\}) = 1 - \exp[-2\beta]. \tag{2.42}$$

The Wolff algorithm was chosen over the Swendsen and Wang implementation since the autocorrelation times are lower and the update involves only a subset of the entire graph. The necessary steps for one Wolff cluster sweep of the Ising model are given below.

- Choose an initial site x on the graph G as the first element of the cluster c.
- Visit all edges connecting $x \in c$ to its nearest neighbours y.
- If the spin at site y equals that at site x, $\sigma_x = \sigma_y$, activate edge $\{x, y\}$ with probability $P(\{x, y\})$, to include y in cluster c.
- Iterate over all untested edges incident to the cluster until all edges have been tested and fail, leaving the cluster no more growth opportunity.
- Flip the spins at all sites $x \in c$, giving $\sigma_x \to r\sigma_x = -\sigma_x$.

Ergodicity can be proven using a simple and intuitive argument. It is possible to choose any site initially and there is a non-zero possibility of the cluster containing

only one spin. There is also a non-zero possibility of choosing the same initial site twice, negating the spin flip. Therefore, at most there are $|V|$ spin flips between any two spin configurations and the algorithm is irreducible. Aperiodicity results from the ability of the algorithm to flip clusters of different sizes. A number of adjacent spins may be flipped individually in consecutive sweeps and flipped back in a single sweep. Positive recurrence results from an irreducible, aperiodic Markov chain in a finite state space.

2.2.5 Bootstrap Resampling

Following our Monte Carlo simulations, we estimate the mean of the magnetisation and energy of our ensemble. As we have already seen, the standard error of these means depends on the number of samples taken. Resampling methods have become the standard technique for estimating the error in ensemble means calculated using Monte Carlo. The most popular resampling methods are jackknife and bootstrap resampling [19, 20]. The jackknife has been shown to be a linear approximation to the bootstrap resampling method, but has the advantage of being computationally less intensive. The jackknife estimate of the error also does not change when repeated on the same data set.

Our Monte Carlo algorithm produces a sequence of random variables,

$$y = \{x_1, x_2, x_3, \ldots, x_N\}. \tag{2.43}$$

To calculate the bootstrap estimate of standard error $\hat{\sigma}_B$, draw N samples from y with replacement to give,

$$y^* = \{x_1^*, x_2^*, x_3^*, \ldots, x_N^*\}. \tag{2.44}$$

Repeat this sampling to draw a large number of bootstrap samples,

$$y^*(1), y^*(2), \ldots, y^*(B). \tag{2.45}$$

For each bootstrap sample $y^*(b)$, evaluate the statistic of interest (sample mean),

$$\hat{F}^*(b) = \hat{F}(y^*(b)). \tag{2.46}$$

Calculate the standard deviation of $\hat{F}^*(B)$,

$$\hat{\sigma}_B = \sqrt{\frac{\sum_{b=1}^{B}\{\hat{F}^*(b) - \langle\hat{F}^*\rangle\}^2}{B-1}} \tag{2.47}$$

where

$$\langle\hat{F}^*\rangle = B^{-1}\sum_{b=1}^{B}\hat{F}^*(b). \tag{2.48}$$

When dealing with correlated data, such as that produced during a Monte Carlo run, it can be useful to use a blocked bootstrap method. In this variation of the algorithm, the data is divided into M blocks of length $m = N/M \gg \tau_{\text{int}}$, which are resampled instead of individual samples [21, 22]. Following this treatment of the Monte Carlo data, we end up with a sample mean and the standard error of the mean. The standard error will be reported in this thesis in parentheses adjacent to estimates of the mean.

2.2.6 Reweighting

Monte Carlo simulations of the canonical ensemble are usually run over a range of T values centred on T_c. Shifts in T_c due to finite size effects can make it difficult to predict the location of the finite volume critical temperature, $T_c(V)$. This discretisation error can be rather large compared to the standard error and in turn leads to inaccuracies in determining the critical temperature using the peak in susceptibilities, overlap in Binder cumulants or finite size scaling.

Reweighting has become the standard tool for locating T_c in systems displaying a second order phase transition [23]. It uses the Monte Carlo observable histories at a given inverse temperature β_0 and reweights them across different inverse temperatures close to β [24]. In this way, the maxima of susceptibilities and the crossing of cumulants of different volumes can be found at a finer T resolution.

The standard estimate of the expectation value at inverse temperature β_0 is

$$\langle O\rangle_{\beta_0} = Z_{\beta_0}^{-1}\sum_{\{\sigma\}} O(\sigma)\exp[-\beta_0 E(\sigma)] \approx N^{-1}\sum_{i=1}^{N} O_i \tag{2.49}$$

where the set of N Monte Carlo measurements of observable O, labelled O_i, are sampled from the Boltzmann distribution. Similarly to Eq. 2.49, the expectation value for a

general inverse temperature β can be given as,

$$\langle O\rangle_\beta = Z_\beta^{-1}\sum_{\{\sigma\}} O(\sigma)\exp[-\beta E(\sigma)], \tag{2.50}$$

$$= Z_\beta^{-1}\sum_{\{\sigma\}} O(\sigma)\exp[-(\beta-\beta_0)E(\sigma)]\exp[-\beta_0 E(\sigma)]. \tag{2.51}$$

Given that the probability of a spin configuration, σ, at temperature β_0 is,

$$P(\sigma,\beta_0) = Z_{\beta_0}^{-1}\exp[-\beta_0 E(\sigma)], \tag{2.52}$$

Eq. 2.51 can be expressed in terms of an expectation value calculated using the Boltzmann distribution with β_0,

$$\langle O\rangle_\beta = \frac{Z_{\beta_0}}{Z_\beta}\sum_{\{\sigma\}} O(\sigma)\exp[-(\beta-\beta_0)E(\sigma)]P(\sigma,\beta_0), \tag{2.53}$$

$$= \frac{Z_{\beta_0}}{Z_\beta}\langle O(\sigma)\exp[-(\beta-\beta_0)E(\sigma)]\rangle_{\beta_0}. \tag{2.54}$$

The sum over the probability distribution must add to one, therefore, for a general inverse temperature β,

$$P(\sigma,\beta) = Z_\beta^{-1}\exp[-\beta E(\sigma)], \tag{2.55}$$

$$= Z_\beta^{-1}\exp[-(\beta-\beta_0)E(\sigma)]\exp[-\beta_0 E(\sigma)], \tag{2.56}$$

$$= \frac{Z_{\beta_0}}{Z_\beta}\exp[-(\beta-\beta_0)E(\sigma)]P(\sigma,\beta_0), \tag{2.57}$$

and,

$$\sum_{\{\sigma\}} P(\sigma,\beta) = \sum_{\{\sigma\}}\frac{Z_{\beta_0}}{Z_\beta}\exp[-(\beta-\beta_0)E(\sigma)]P(\sigma,\beta_0), \tag{2.58}$$

$$= \frac{Z_{\beta_0}}{Z_\beta}\langle\exp[-(\beta-\beta_0)E(\sigma)]\rangle_{\beta_0} = 1. \tag{2.59}$$

Therefore, the prefactor in Eq. 2.54 can be expressed solely in terms of the energy observables $E(\sigma)$ and the inverse temperatures β and β_0,

$$\langle\exp[-(\beta-\beta_0)E(\sigma)]\rangle_{\beta_0} = \frac{Z_\beta}{Z_{\beta_0}}. \tag{2.60}$$

Using the N Monte Carlo measurements, O_i and E_i which come from the same

configuration σ_i,

$$\langle O \rangle_\beta = \frac{\langle O(\sigma) \exp[-(\beta - \beta_0)E(\sigma)] \rangle_{\beta_0}}{\langle \exp[-(\beta - \beta_0)E(\sigma)] \rangle_{\beta_0}} \tag{2.61}$$

$$\approx \frac{\sum_{i=1}^{N} O_i \exp[-(\beta - \beta_0)E_i]}{\sum_{i=1}^{N} \exp[-(\beta - \beta_0)E_i]}. \tag{2.62}$$

Using this treatment, we can reweight any ensemble mean or higher moment, such as the susceptibility

$$\chi = \beta V \langle (|m| - \langle |m| \rangle)^2 \rangle = \beta V (\langle m^2 \rangle - \langle |m| \rangle^2), \tag{2.63}$$

where

$$m = M/V, \tag{2.64}$$

with M defined in Eq. 2.10 and specific heat

$$C_v = \frac{\beta^2}{V} \langle (E - \langle E \rangle)^2 \rangle = \frac{\beta^2}{V} (\langle E^2 \rangle - \langle E \rangle^2). \tag{2.65}$$

In practice, estimating $\langle E \rangle_\beta$ and $\langle m \rangle_\beta$ using reweighting are both straightforward numerically, for example;

$$\langle E \rangle_\beta = \frac{\sum_{i=1}^{N} E_i \exp[-(\beta - \beta_0)E_i]}{\sum_{i=1}^{N} \exp[-(\beta - \beta_0)E_i]}. \tag{2.66}$$

The appropriate susceptibilities depend on the resampled average energy which can be reused from above,

$$C_v(\beta) = \frac{\beta^2}{V} \frac{\sum_{i=1}^{N} (E_i - \langle E \rangle_\beta)^2 \exp[-(\beta - \beta_0)E_i]}{\sum_{i=1}^{N} \exp[-(\beta - \beta_0)E_i]}. \tag{2.67}$$

Note that the average energy in the numerator is not simply the estimator of the sample mean at β_0, but is the reweighted mean $\langle E \rangle_\beta$ from above.

This reweighting treatment is used to both estimate the susceptibilities of our observables and the critical temperature at which they peak. The errors of both of these are estimated using blocked bootstrap resampling. The N samples of observable data O_i are blocked into M blocks of length $m \gg \tau_{\text{int}}$. M blocks are randomly chosen with replacement, according to the bootstrap methods and the bootstrap sample means are

calculated,

$$\langle O\rangle^*_\beta = \frac{\sum_{i=1}^{N} O_i \exp[-(\beta-\beta_0)E_i]}{\sum_{i=1}^{N} \exp[-(\beta-\beta_0)E_i]} \tag{2.68}$$

The standard form of the bootstrap estimate of the standard error is then used.

Estimating the error in the critical temperature is somewhat more involved. We must divide the data into M blocks of length $m \gg \tau_{\rm int}$. Each block of data is reweighted over a small range of β around the β_c where the susceptibility peaks. The temperatures $\beta^*_{\rm max}$, at which the susceptibility peaks for each block are recorded. The jackknife estimate of the standard error for $\beta_{\rm max}$ is then calculated as

$$\delta\beta^*_{\rm max} = \sqrt{\frac{M-1}{M}\sum_m (\beta^*_{\rm max} - \langle\beta^*_{\rm max}\rangle)^2} \tag{2.69}$$

where

$$\beta^*_{\rm max} = M^{-1}\sum_{i=1}^{M} \beta^*_{i,\rm max}. \tag{2.70}$$

Some care should be taken not to reweight too far from the original sampled temperature β_0 since the reweighting introduces a large error for states which are not abundant at this temperature. For the energy reweighting, the distance between the reweighted mean and the original sample mean should not exceed the fluctuation of the samples themselves,

$$|\delta\langle E\rangle| \leq \langle (E - \langle E\rangle_{\beta_0})^2\rangle^{1/2}. \tag{2.71}$$

For our purposes, we choose data at a temperature β_0 as close to T_c as possible to estimate the reweighted critical temperature β_c using susceptibility peaks.

2.3 Results and Discussion

In this section, we present the results of our Monte Carlo simulations of the Ising model on complex interference networks. The results have been arranged in three subsections for convenience. Sec. 2.3.1 explores the loss of ergodicity which may occur in these simulations. In Sec. 2.3.2 the efficiency of the Monte Carlo sampling in producing

independent configurations is presented and finally, in Sec. 2.3.3, the Monte Carlo estimates of thermodynamic observables are shown.

2.3.1 Metropolis Algorithm: Loss of Ergodicity

Our experience with these Monte Carlo simulations on complex networks has shown that we must be very careful with inspecting the observables prior to performing any statistical analysis.

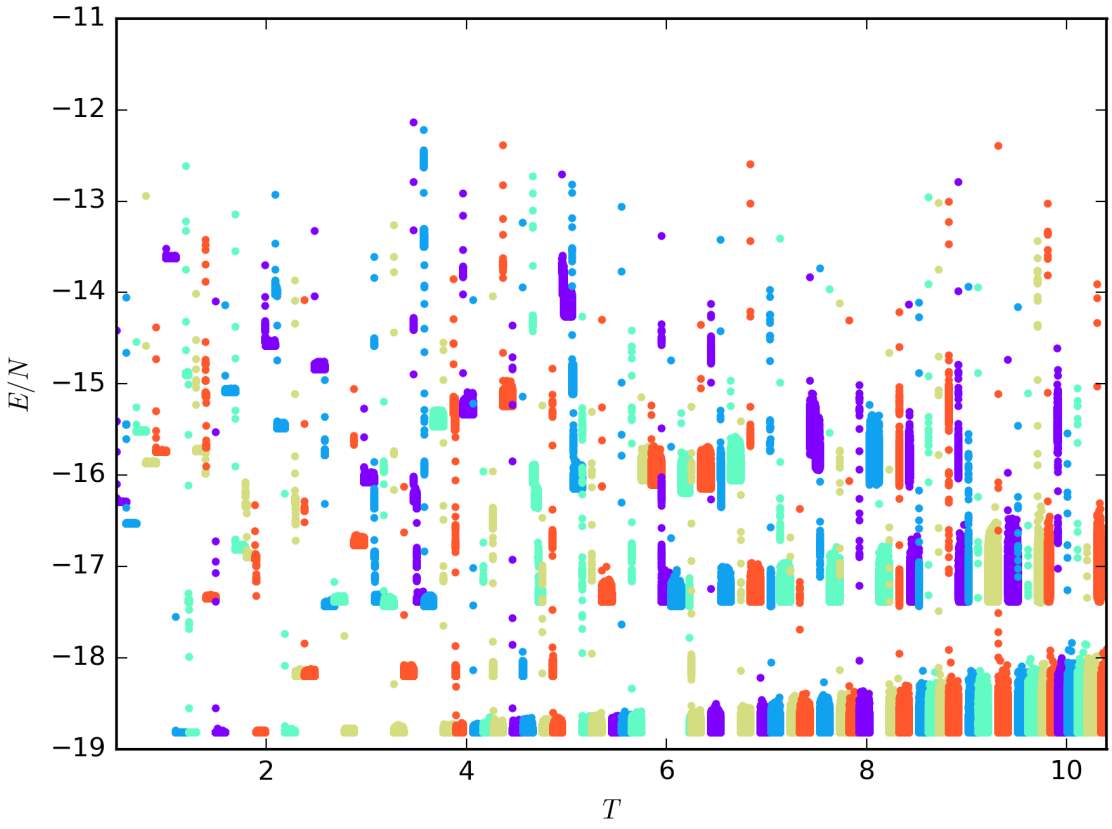


Figure 2.3: Plot of Ising model Energy observable thermalisation histories produced by the Metropolis algorithm. The Ising model is embedded on the $\theta = 1\%$ graph. From low to high T, a colour change indicates moving from the history of one independent run to the next, with consecutive runs performed at higher temperature. 10^4 iterations at each T run are shown, from $T_{\min} = 0.5$ to $T_{\max} = 10.4$. The grouped nature of the energy histories, which do not display a clear mean which increases with temperature is due to the loss of ergodicity. For reference, the lowest energy state occurs at $E/N = -18.81$ and is the fully ordered state.

As we previously mentioned in Sec. 2.2.3, the increased heterogeneity in degree can cause unusually large ΔE values to occur during Metropolis algorithm sweeps through

the complex network. These large values can suppress the transition probabilities to higher energy states. Following each sweep of the Metropolis algorithm, the energy of that configuration is calculated and stored. These energy histories are then used to estimate the average energy and specific heat as shown in Sec. 2.1.1. Ideally, these energies fluctuate around a well defined mean with no large jumps; a sign of short autocorrelation times.

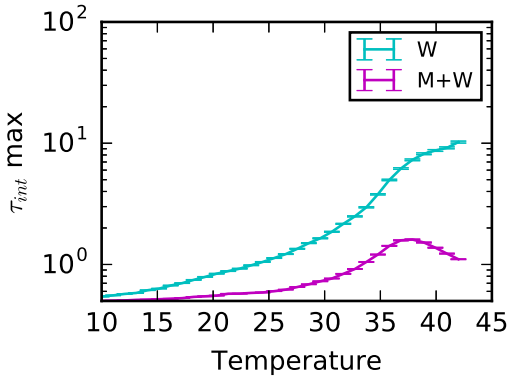
An example of a set of energy histories from an exclusively Metropolis algorithm simulation which loses ergodicity is shown in Fig. 2.3. The system was started in a hot/random state to begin with and the initial transient to lower energy states can be seen in many cases as a column of points of the same colour. These columns are the energies due to the thermalisation phase of the runs.

The stratification of energy histories clearly shows that the system gets stuck in local energy minima which take a long time to leave. For this reason, we deem the Metropolis algorithm alone to be unsuitable for our Monte Carlo simulations. For the rest of the chapter we use both Wolff and consecutive Metropolis and Wolff cluster sweeps to generate new configurations. Both of these approaches were able to access the same range of energies but with different autocorrelation times.

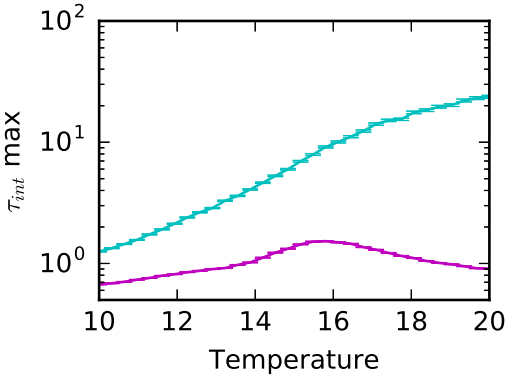
2.3.2 Integrated Autocorrelation

The Monte Carlo energy histories resulting from both exclusive Wolff updates and combined Metropolis and Wolff algorithm simulations were found to be well-behaved on these networks. Both displayed a well defined mean and the energy histories fluctuated around this mean.

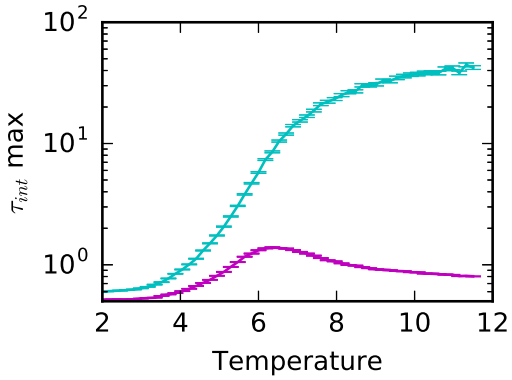
The integrated autocorrelation times for both are shown in Fig. 2.4. The peak that occurs in the Wolff and Metropolis & Wolff autocorrelation times can be used as an early indication of the presence of critical behaviour in the region of this peak.



(a) $\theta = 1\%$



(b) $\theta = 3\%$



(c) $\theta = 5\%$

Figure 2.4: Plot of the plateau values of the integrated autocorrelation time, τ_{int} estimated for the Energy observable. These simulations were performed on our θ complex networks. The Wolff algorithm and Metropolis & Wolff algorithm results are displayed. Due to loss of ergodicity, the Metropolis algorithm results are not considered here.

The autocorrelation is under control at both high and low temperatures using the combined Wolff and Metropolis simulations. This low autocorrelation results in a very small contribution to the naive standard error of the observables shown in the next section.

2.3.3 Critical Behaviour

In this section, we present the thermodynamic observables estimated using the combined Wolff and Metropolis Monte Carlo simulations. Fig. 2.5 shows that the average magnetisation falls as the temperature increases. This drop in magnetisation indicates that the system is transitioning from a fully ordered phase to a more disordered one.

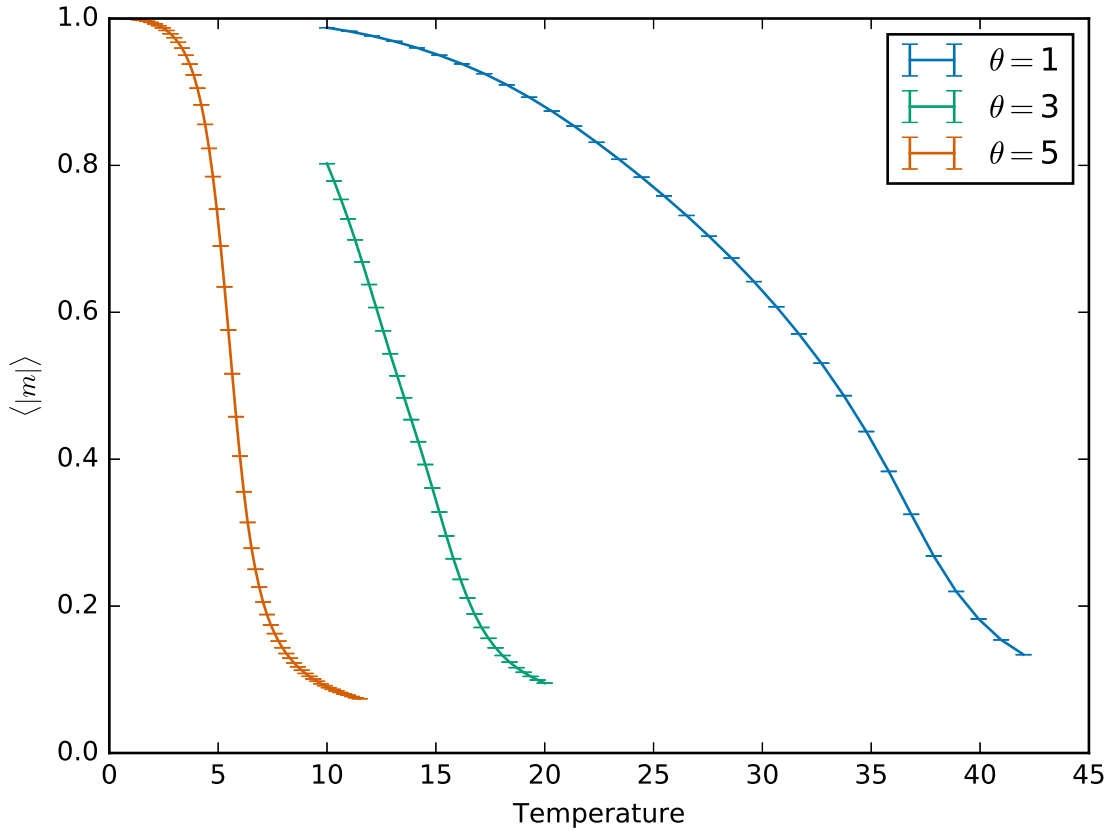


Figure 2.5: Mean magnetisation per site on the θ complex network, with bootstrap re-sampled errors. The spontaneous magnetisation shown by our Monte Carlo simulations indicates the presence of a crossover between ordered and disordered spin phases. Each data point is the mean over energy samples from 10^6 consecutive combined Metropolis & Wolff Cluster Monte Carlo sweeps after thermalising for 10^5 sweeps. Standard error of the mean estimates are calculated using 10^3 bootstrap resamples. Unless otherwise stated, all other results in this chapter were calculated with the same level of statistics.

The magnetisation curve itself becomes more rounded and drawn out for the graphs with lower θ values. The $\theta = 1\%$ graph has the smallest average path length $\langle l \rangle$ of the three graphs shown here and they all possess a similar clustering coefficient, $C \sim 0.6$. The $\theta = 1\%$ graph also has the most heterogeneous degree distribution. The tendency for the magnetisation to fall over this larger temperature range is comparable to that of the behaviour in the WS model small-world network.

The mean energy per site in Fig. 2.6 shows the accompanying increase in the internal energy in the system as temperature increases. The ground state energies per site are listed in the caption. It is clear then that the Monte Carlo method has no problem sampling both high and low energy states of our system. The significantly lower ground

state energy for the $\theta = 1\%$ graph is due to the far larger number of edges than the others.

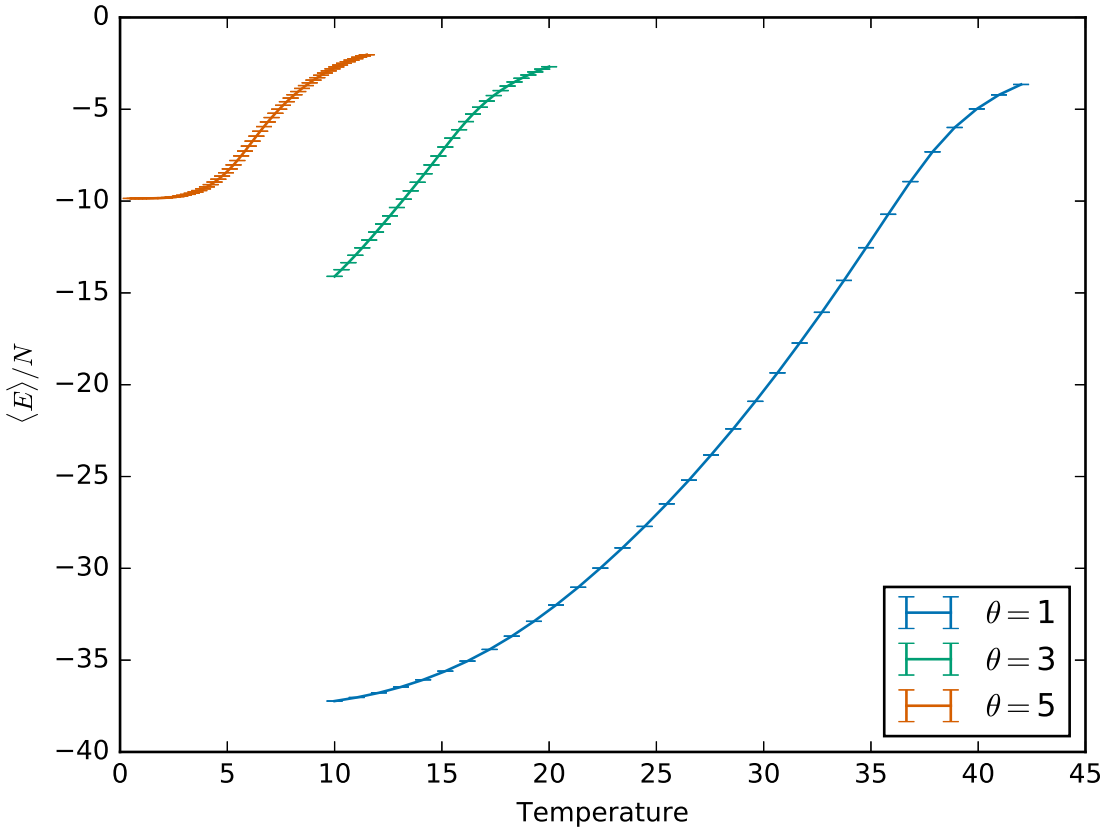


Figure 2.6: The mean energy per site on our complex network, with bootstrap resampled errors. In the temperature region where the magnetisation changes rapidly, there is a corresponding rise in the internal energy of the system. The ground state energies per site for the $\theta = 1\%$, 3% and 5% graphs are $-37.62\ J/k_B$, $-17.58\ J/k_B$ and $-9.86\ J/k_B$ respectively.

A peak occurs in the specific heat curves in Fig. 2.7 around the temperature range where the average magnetisation occurs. Again, the $\theta = 1\%$ peak is significantly less sharp and occurs over a much larger temperature range than the other graphs. None of these curves appear to be symmetric about the maximum and the $\theta = 3\%$ peak is almost bimodal.

We use the maxima in the specific heat curves to estimate the critical temperatures on the networks. However, it is not often clear where the true peak of these curves actually lies since we perform our simulations at a range of temperatures chosen prior to simulation.

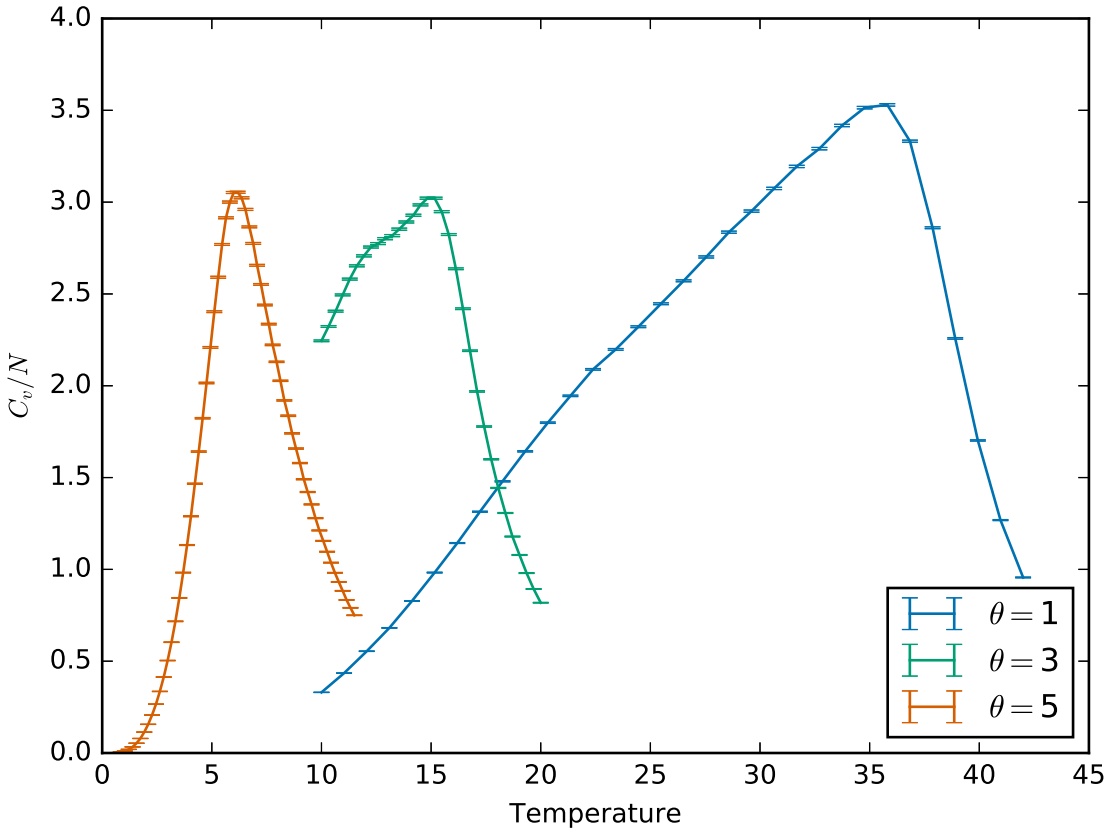
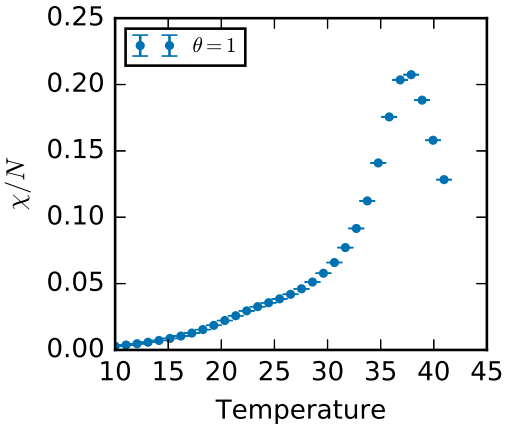


Figure 2.7: The specific heat per site on the complex networks, with bootstrap resampled errors. A peak appears in the specific heat in the temperature range where the spontaneous magnetisation occurs. The peak is much more narrow and symmetric for the $\theta = 5\%$ graph, which has the least edges. As θ increases, the peaks are shifted toward higher temperatures and become broader and less symmetric.

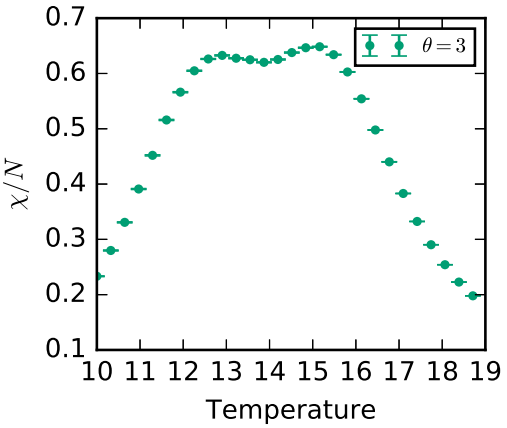
In Fig. 2.8, this problem becomes quite clear. For example, in Subfigs. 2.8a and 2.8c two adjacent values of the susceptibility per node occur close to the peak. Especially in the Subfig. 2.8a, the temperature difference between these points is quite substantial and the true peak presumably lies between these points.

The bimodal character of the Subfig. 2.8b for the $\theta = 3\%$ graph is quite striking. For this reason, the Monte Carlo data from both Metropolis & Wolff simulations and Wolff-only simulations were closely inspected to determine if this two-peak structure is robust. The magnetisation histories, integrated autocorrelation times and susceptibility calculations themselves were repeatedly inspected and tested for either human or simulation error. Following this close scrutiny, we found the magnetisation histories to be well behaved. As expected, at low temperature the histogram of the magnetisation histories shows two symmetric peaks of equal height close to the magnetisation values

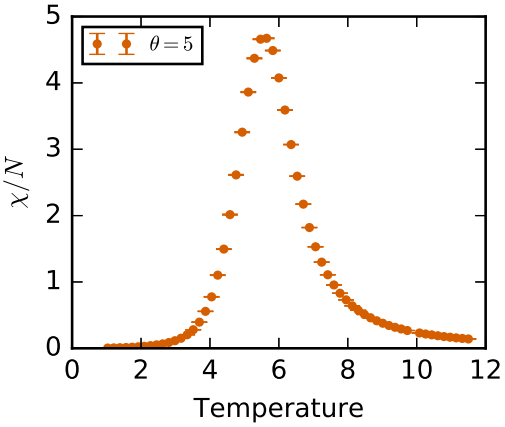
of +1 and −1. No visible signs of autocorrelation in the magnetisation histograms were present, such as unequal height in the two peaks or more than two peaks. This verifies that the Monte Carlo algorithm was successful in tunnelling between the spin-up (+1) and spin-down (−1) aligned states. This is a strong indication that the Markov Chain is ergodic. For simulations at higher temperatures, the two peaks in the magnetisation histogram are shifted towards mean magnetisation values of 0 and eventually join to form a single peak, indicating a crossover to a disordered phase. As shown already in Fig. 2.2.2, the autocorrelation times are quite small and as a result of our testing and verification, we are confident that we have not made an error in calculating the susceptibility curves.



(a) $\theta = 1\%$



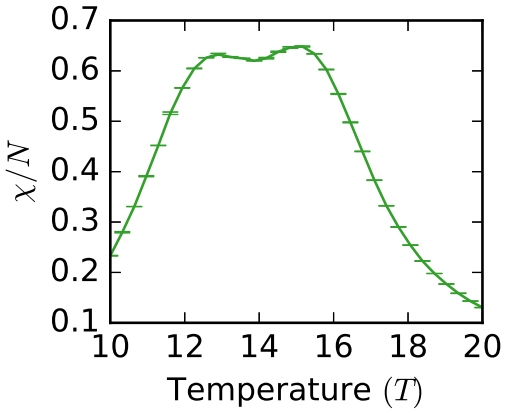
(b) $\theta = 3\%$



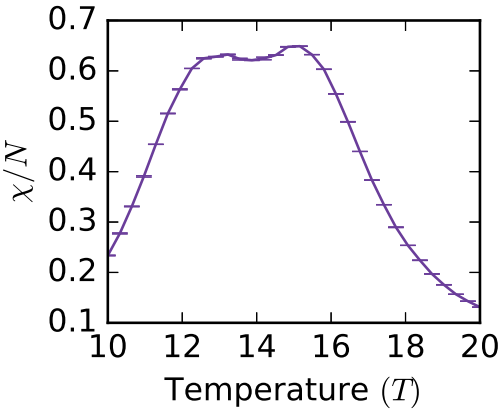
(c) $\theta = 5\%$

Figure 2.8: The susceptibility per site on our complex networks, with bootstrap resampled errors.

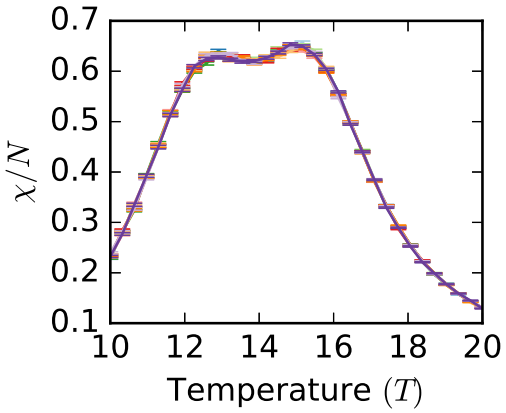
In Fig. 2.9, the susceptibility curves for the Ising model on the $\theta = 3\%$ networks are compared. In Subfigs. 2.9a and 2.9b, the susceptibility curves are produced from Metropolis & Wolff and Wolff-only simulation data respectively. The twin-peak structure seen in Fig. 2.8 is also visible in the independently obtained Wolff data.



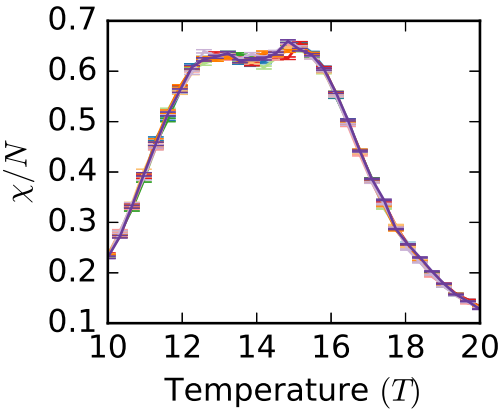
(a) Metropolis and Wolff sampling (10^6 samples)



(b) Wolff only sampling (10^6 samples)



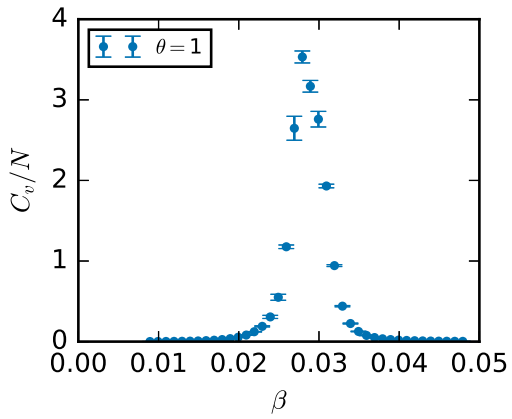
(c) Metro and Wolff sampling (10^5 samples each)



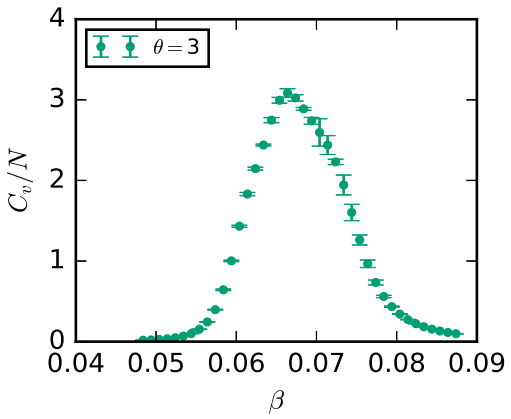
(d) Wolff only sampling (10^5 samples each)

Figure 2.9: This plot compares the susceptibility per spin for the $\theta = 3\%$ networks using (Subfig. 2.9a) Metropolis and Wolff and (Subfig. 2.9b) Wolff-only Monte Carlo data. The curves in Subfigs. 2.9a and 2.9b were produced using 10^6 samples per temperature value with bootstrap resampled standard errors. Subfigs. 2.9c and 2.9d show the susceptibilities per node calculated using subsets of the Monte Carlo data. For both of these Subfigs., the 10^6 sample histories were split into 10 subsets of 10^5 consecutive magnetisation values. The twin-peak structure, separated by a trough, is present in all curves in this figure, with a difference between peak and trough susceptibility values of at least two standard errors and up to more than ten.

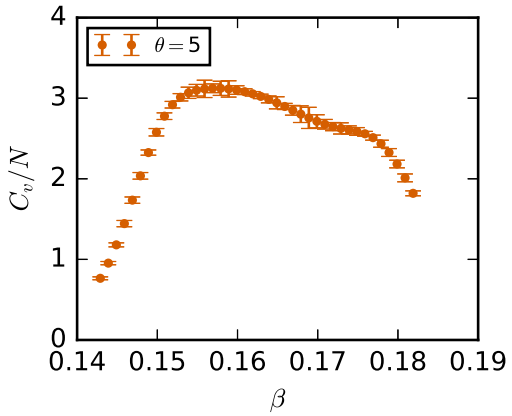
The curves in both sub-figures are produced using 10^6 samples per temperature value and the standard errors are calculated using bootstrap resampling. In both sub-figures, a peak is visible near temperature, $T = 13.2$, with a minimum near $T = 13.9$ and a second peak near $T = 15.2$. The susceptibility values of the peaks are over ten standard errors larger than those at the troughs between them.



(a) $\theta = 1\%$



(b) $\theta = 3\%$

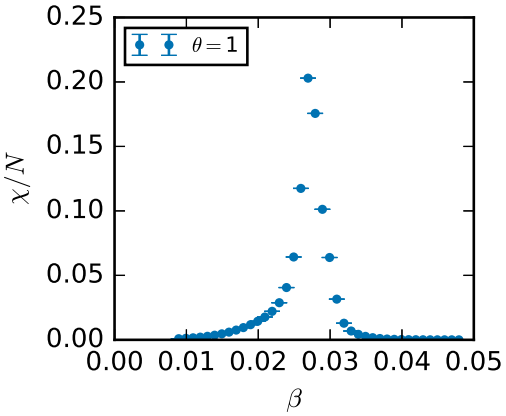


(c) $\theta = 5\%$

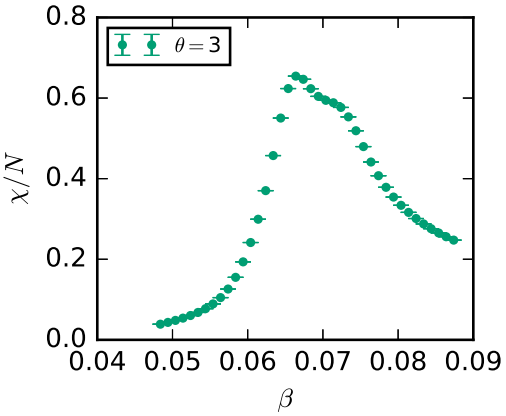
Figure 2.10: Reweighted specific heat curves with bootstrap errorbars. The reweighted data was obtained from Monte Carlo simulation performed at the temperature where the specific heat peaks, $T = \{35.806\ J/k_B,\ 14.839\ J/k_B,\ 6.177\ J/k_B\}$ for each of the $\theta = 1\%, 3\%$ and 5% graphs respectively.

Additionally, due to the low autocorrelation times and large number of samples, the long magnetisation histories at each temperature point can be divided into ten separate and essentially independent histories. Using these ten histories for both the W and

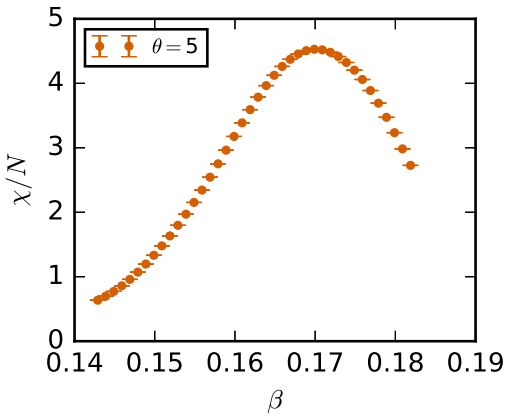
M&W data, the curves in Subfigs. 2.9c and 2.9d were produced. In each of the ten curves, for both M and M&W, the peaks and separating trough are present. Further, the height of the peaks are at least two standard errors greater than the troughs in all ten curves, with differences of up to 10 standard errors. This data strongly supports our conclusion that the twin-peak structure is a true feature of this system. A more thorough study of this behaviour is warranted to find the underlying cause, however this was beyond the original scope of this work.



(a) $\theta = 1\%$



(b) $\theta = 3\%$



(c) $\theta = 5\%$

Figure 2.11: Reweighted susceptibility curves with bootstrap errorbars. The reweighted data was obtained from Monte Carlo simulation performed at the temperature where the susceptibility peaks.

In Fig. 2.10, we improve on the finite T separations by reweighting the energy history using the method set out in Sec. 2.2.6. The reweighting was performed on

the Monte Carlo energy history which corresponded to the maximum of the peak in Fig. 2.7. The much finer resolution in T allows us to judge more precisely the peak in specific heat and consequently the critical temperature. This procedure was repeated for the susceptibility shown in Fig. 2.11.

The estimated critical temperatures themselves are shown in Tab. 2.2. In general, the critical temperature increases as θ decreases, meaning that as the nodes become more interconnected the critical temperature increases. This behaviour agrees qualitatively with the critical temperature of the WS model increasing as the fraction of interconnected nodes increase.

However, the critical temperatures due to the specific heat and susceptibility peaks in Tab. 2.2 do not closely agree. While these temperatures are qualitatively similar, there are many σ in the difference between them in all cases. It is not clear from the work performed here whether these differences are due to a finite size effect or if there is a difference in critical behaviour for different observables.

Some difficulties arise in performing larger volume simulations to suppress boundary and finite volume effects, since these networks are generated using real-world data. This is definitely an area that we believe would benefit from future work.

$\boldsymbol{\theta}$	$T_c(C_v\,\max)$	$T_c(\chi\,\max)$
1%	35.8(3)	37.1(2)
3%	15.06(2)	14.61(4)
5%	6.37(5)	5.88(1)

Table 2.2: Estimated critical temperature of the complex networks based on the temperature at which the peak in the specific heat and susceptibility occur. The errors are estimated using bootstrap resampling.

2.4 Conclusion and Future Work

In this chapter, we have performed Monte Carlo simulations of the ferromagnetic Ising model on complex networks. The critical behaviour of this model is known to differ from the usual lattice behaviour on complex networks and we show that the network itself is responsible for this change. The complex networks which we perform our simulations on are derived from the potential interference between basestations in real-world mobile phone networks.

The Monte Carlo energy histories resulting from our simulations indicate a loss of

ergodicity when using the Metropolis algorithm only. The loss of ergodicity is due to nodes in the network which have many more connections than average. The probability of flipping these highly connected nodes can become very low due to the number of connections.

Performing the simulations using consecutive sweeps of the Wolff cluster and Metropolis algorithm is sufficient to maintain ergodicity of the Markov chain. The correlation between successive configurations of the system using these algorithms is shown to be extremely low, as measured by the integrated autocorrelation time for the energy observable.

The average magnetisation observable is used to show that at low temperature an ordered phase occurs in which all spins are aligned in the same direction. At high temperature, a disordered phase is observed. The temperature at which this order/disorder crossover occurs is shown to rise as the number of interconnections increases in our network (i.e. low θ values). The critical behaviour also becomes less sharp and occurs over a larger temperature range for the more interconnected network. This behaviour is very similar to that expected in the Watts-Strogatz family of complex networks, where the critical temperature rises with increased interconnectedness and becomes less sharp.

The critical temperature itself is estimated to be the point at which the magnetic susceptibility and specific heat curves peak. To overcome the large error in temperature which results from simulating over an interpolated temperature range, the data is reweighted at a finer temperature resolution to determine the critical point.

Of the work performed for this chapter, we believe that two areas in particular would benefit from further study. The first is the bimodal susceptibility observed in the $\theta = 3\%$ graph. Despite in-depth statistical analysis of the Monte Carlo data, this bimodal peak is yet to be explained. We hypothesise that this is a real behaviour due to some high degree hubs becoming easier to flip at higher temperature and effecting the overall phase transition as a result.

In addition to this, the difference in the critical temperatures using the specific heat and susceptibility maxima should be further investigated. One possible avenue of investigation would be to repeat this work on interference networks of larger cities where finite size and boundary effects would be less influential.

Chapter 3

Antiferromagnetic Potts model

In this chapter, we study the antiferromagnetic q-state Potts model on the θ complex networks. Originally, these complex networks were created to model the potential interference between mobile phone basestations broadcasting on the same channel. We use these networks to investigate the situation where basestations have q channels to choose from. Limits on signalling capacity force these modern mobile networks to make local decisions based on their surrounding environment. This suggests that the nearest neighbour interactions in the Potts model can be used as a simple description of distributed channel allocation on these networks.

The critical behaviour of the antiferromagnetic Potts model involves an ordered low temperature phase of alternating values of adjacent spins. At higher temperature, a disordered phase with almost uniformly distributed spin values occurs. To minimise interference in a mobile phone network, an operator would be wise to ensure the system stays in the low temperature ordered phase so that adjacent base stations use different channels. On complex networks however, due to the random nature of adjacent nodes, it becomes difficult to achieve this ordered state and spin-glass characteristics may be exhibited. This spin-glass behaviour causes a non-zero ground state with a high degeneracy in the Potts model. Therefore, the nature of the degeneracy of low interference states then becomes very important for mobile network operators.

Performing conventional Monte Carlo simulations in models containing a spin-glass phase becomes almost impossible and for this reason we use the multicanonical Wang-Landau (WL) algorithm from Ref. [2]. The WL algorithm estimates the density of states (DOS) which is independent of temperature and therefore avoids spin-glass simulation problems. We consider the Potts energy [25] to be equivalent to the level of interference in the network. The degeneracy of these interference levels is numerically

estimated to find the density of states. With an estimate of the DOS, we can examine both the distribution of interference levels and the behaviour of thermodynamic observables of the Potts model.

This chapter contains a further four sections. Sec. 3.1 provides a brief overview of the antiferromagnetic Potts model, its previous use in network engineering and some known behaviour of this model on complex networks. Sec. 3.2 discusses the basis of the Wang-Landau Monte Carlo algorithm, how its errors scale and the derivation of canonical ensemble thermodynamic observables. Sec. 3.3 presents our numerical results and some discussion on the difficulty of performing Monte Carlo on complex networks. Finally, we conclude this chapter with Sec. 3.4 and a discussion of future work. The majority of the work performed in this chapter is found in Ref. [3].

3.1 Background

Treating mobile phone networks as complex systems has become quite popular, such as in Refs. [26, 27, 28, 29, 30] where the authors focus on human dynamics and social interaction via mobile phones. Critical phenomena in wired communication networks are studied in Refs. [31, 32, 33]. Transmission rate calculations and interference reduction in wireless networks was performed in Refs. [34, 35] using the replica method to evaluate an analogue of the free energy of the system. A great deal of attention has especially been given to the Ising and Potts models in wireless networks. Some applications of these spin models have been; calculating the transmission probability and throughput [36, 37], recreating network topology based on mutual information shared between nodes [38], distributed configuration management [39] and adaptive scheduling for wireless networks of sensors for energy efficiency [40]. The use of interference graphs to study interference in wireless networks is discussed in Ref. [41]. Alternative methods for statistical interference analysis among the wireless network community also include stochastic geometry of node placement [42] and a novel circular interference model as in Ref. [43]. Our treatment of interference networks using the q-state Potts model begins in the next section, where antiferromagnetism is further explained.

3.1.1 Potts Model

The Potts model representing interference in our complex networks is described here. Each vertex in our graphs (basestation sector[1]) $i \in V$ possesses a transmitter using a frequency channel (spin) which may take one of the discrete values $\sigma_i \in [1, q]$ representing the division of the spectrum into q orthogonal channels. Each spin interacts with its nearest neighbours only, interfering with all neighbours using the same channel. The local interference per basestation is summed to define the overall interference, which is equivalent to the Potts model energy. Several q values are considered in this chapter on three different graphs, derived from a real-world wireless network. The choice of these q-values is informed by the frequency reuse schemes described in Ref. [44].

The Potts model energy was discussed already in Chapter 2. The antiferromagnetic Potts model is relevant for interference and we henceforth set $J = -1$. For illustration, consider the interaction between a single basestation sector and its nearest neighbours in this set-up. Setting the frequency of sector i (denoted σ_i) and that of all its nearest neighbours to the same value adds n_i to the total interference, where n_i is the number of nearest neighbours of sector i. Our definition of interference assigns the largest interference to configurations in which each sector uses the same frequency channel. The negative interaction constant effectively punishes neighbouring aligned spins, representing neighbouring basestations using the same frequency channel, by assigning a larger interference.

Clearly, lower interference will result from configurations in which neighbouring basestations use different frequency channels. However, the topology of the network plays a crucial role in determining the lowest possible interference level. Consider the $q = 2$ Potts model (the Ising Model) on a triangular lattice [45]. Due to the presence of triangles in the network, it is not possible for three basestations situated on the vertices of a triangle to be completely non-interfering as they are all mutual nearest neighbours and there are only two frequency channels available. This phenomenon is known as 'geometric frustration' and increases both the lowest possible interference level as well as its degeneracy. For the same $q = 2$ Potts model on a square lattice, geometric frustration does not occur and zero-interference configurations are possible by alternating frequencies on the 'even' and 'odd' sites of the lattice. In this manner a particular sector is always using a different frequency than its neighbours and no interference occurs. We now consider the degeneracy of this zero-interference level.

[1] In cellular networks, a basestation covers a given area which is divided into sectors, each served by a separate transmitter.

There are exactly two configurations which give zero interference: the one in which the even sectors are assigned frequency one and the odd sectors frequency two, and the alternative configuration where this assignment is reversed. The resulting degeneracy of this zero interference state is two, regardless of the size of the square lattice. The triangular lattice possesses a substantially higher degeneracy for the lowest interference level. In fact, it has been shown [45] that the degeneracy of the minimum energy state for the $q = 2$ Potts model on a triangular lattice grows exponentially with the size of the lattice. This results in a non-zero specific ground state entropy (proportional to the natural logarithm of the degeneracy) in the thermodynamic limit.

The appearance of geometric frustration can be understood using the graph theory concept of cliques. A clique is a subset of vertices that are all connected to each other, i.e. a set of nodes that are all mutual nearest neighbours. For example, a triangle is a 3-vertex clique. As discussed above, the $q = 2$ Potts model experiences geometric frustration on the triangular lattice, which is composed of 3-cliques. Generally, an arbitrary q-state Potts model will experience frustration on networks which contain $(q + 1)$-cliques. It should be noted that the phenomenon of geometric frustration and its connection to the topology of the network and q are well-known in graph colouring.

The presence of geometric frustration and the randomness in the edges of a complex network has been found to cause spin-glass behaviour at low temperatures. Spin-glasses are not well understood on complex networks. In fact, only recently has the study of spin-glasses on scale-free and Watts-Strogatz networks started to draw attention. In regular lattices, spin-glasses are known to occur in the Ising model with long-range random couplings J_{ij} of ferromagnetic and antiferromagnetic interactions between spins [46]. In a spin-glass, the system is stuck in a metastable state above the ground state. It has a non-ergodic equilibrium state, a vanishing magnetisation and a non-vanishing correlation between spins at the same lattice points on different replica lattices.

Spin-glass behaviour of the antiferromagnetic Ising model has been found on a scale free network studied using Monte Carlo methods in Ref. [47]. The authors observed a transition between a paramagnetic and spin-glass phase and estimate the critical temperature $T_c \sim 4.0(1)$. The replica exchange method was used to avoid difficulties in simulation due to the spin-glass phase. The Binder forth order cumulant was used to find the critical temperature. Very interestingly, hubs with a high degree were found to freeze for many updates and this is considered a standard behaviour for spin-glasses. The high degree hubs are difficult to flip using standard Monte Carlo methods since the energy difference, $\Delta E_i = -2\sigma_i \sum_{j=1}^{k_i} \sigma_j$, can be extremely high. At lower temperatures, the system can get trapped in local minima of free energy and the time necessary to

move to another allowed region of the state space diverges as system size grows. In this paper, C_v peaks at $T \sim 2.0$ but the critical temperature is estimated to be $T_c \sim 4.0$ using a scaling hypothesis on the Binder parameter. In a similar study in Ref. [48], the authors used the Metropolis algorithm with a simulated annealing approach to tunnel through the complex energy landscape of their system. They found a similar critical temperature, $T_c \sim 4.8(2)$, which is close but still four standard deviations away from the last study. The cause of this discrepancy is unclear.

The critical behaviour of Erdős-Rényi random networks seems to be highly dependent on the proportion of rewired edges, p. No phase transition at any temperature exists in the antiferromagnetic Potts model if $p < 2(q-1)\ln(q-1) = p_1(q)$ [49, 50]. If $p \geq p_1(q)$, a phase transition occurs between the spin-glass phase and the paramagnetic phase above the temperature bound,

$$T_c(q) \geq \begin{cases} -1/\ln\left(1 - \frac{q}{1+\sqrt{p}}\right), & \text{if } q = 2 \\ -1/\ln\left(1 - \frac{q}{1+\sqrt{\frac{p(q-1)}{2\ln(q-1)}}}\right), & \text{otherwise.} \end{cases} \tag{3.1}$$

As discussed above, the stochastic nature of the edges in these graphs, combined with the presence of geometric frustration is likely to result in a spin-glass phase at low temperature as in Refs. [48] and [47]. To avoid loss of ergodicity of a classic Monte Carlo algorithm, we estimate the Potts model degeneracy using the multicanonical Wang-Landau algorithm. The temperature independent sampling avoids simulation difficulties arising at low temperatures such as loss of ergodicity due to a rough energy landscape. We observe interesting features in the distribution of interference levels (density of states) due to configurations of channel assignments. The degeneracy of a particular interference level E, denoted $g(E)$, is the number of frequency allocation configurations which result in this interference level:

$$g(E) = \sum_{\sigma} \delta_{E(\sigma)E}, \tag{3.2}$$

where the sum is taken over all possible frequency allocation configurations. We will use the Wang-Landau algorithm described in the next section to estimate $g(E)$ for a number of different interference networks and q values over the entire range of E.

3.2 The Wang-Landau Algorithm

We estimate the degeneracy of each interference level using the Wang-Landau algorithm. The state space of the q-state Potts model grows exponentially with system volume according to q^N, where N is the number of sectors in the system (nodes). While this behaviour makes naive DOS histogramming extremely inefficient, the Wang-Landau algorithm uses a different approach. It converges to a non-biased random walk in energy space to determine how often each energy state is visited. Since we use the Potts model energy to represent interference, this method lends itself naturally to our cause. The probability of transitioning from one energy state to the next is given by the ratio of the estimated degeneracy of the current energy state ($\tilde{g}(E_i)$) to the estimated degeneracy of the proposed new energy state ($\tilde{g}(E_j)$), i.e.

$$P(E_i \to E_j) = \min\left\{1, \frac{\tilde{g}(E_i)}{\tilde{g}(E_j)}\right\}. \tag{3.3}$$

When the estimated DOS, $\tilde{g}(E)$, approaches the true DOS, $g(E)$, each energy level will have been observed approximately equally. Although Ref. [2] provides a good description of the algorithm, greater detail is provided here on the necessary steps in our simulations. For this work, new energy states are proposed via a Metropolis-style spin-flip mechanism. We iterate through the network, proposing a new spin value at each node with equal probability. If the estimated degeneracy of the current spin configuration $\tilde{g}(E_i)$ is greater than that of the proposed spin configuration $\tilde{g}(E_j)$ then the spin flip is automatically accepted. Otherwise, the spin flip is accepted with the probability of the ratio of the current and proposed energy state degeneracies as in Eq. (3.3). Having performed a complete iteration through the entire network, one sweep has been completed.

On observing energy state E, the estimated DOS is updated according to $\tilde{g}(E) \to \tilde{g}(E) \times f$. The modification factor f is initialised to $f = e^1$ at the start of the run. Since the state space grows so quickly, for efficiency it is more practical to store the natural log of the DOS and perform updates according to $\ln \tilde{g}(E) \to \ln \tilde{g}(E) + \ln f$. Additionally, the observed energy states histogram must be incremented by $h(E) \to h(E) + 1$. As the simulation proceeds, $h(E)$ is tested for flatness. We consider $h(E)$ to be flat when $(1.0 - \epsilon)\bar{h}(E) \leq h(E_i) \leq (1.0 + \epsilon)\bar{h}(E) \; \forall \, i$, where $\bar{h}(E)$ is the mean over all histogram bins. Typically, the Ising model on a square lattice can be run with $0.05 \leq \epsilon \leq 0.2$. On our real-world networks, for computational efficiency the wider tolerance of $\epsilon = 0.2$ is used. Once $h(E)$ is deemed flat, convergence has been achieved for the current value

of f. While the modification factor f is greater than the chosen cut-off, $f > f_{\text{cut}}$, the simulation continues and the modification factor is updated by $f_{i+1} = \sqrt{f_i}$. The histogram $h(E)$ is reset to zero but $\tilde{g}(E)$ is kept. The modification factor, f, must monotonically decrease towards 1 to ensure convergence of the DOS. We denote the final modification factor used in our simulation as f_{final}. A typical f_{cut} value for the Ising model on a square lattice is $f_{\text{cut}} = \exp[10^{-8}]$. In this work, we use $f_{\text{cut}} = \exp[10^{-4}]$ on graphs which converge slowly and $f_{\text{cut}} = \exp[10^{-8}]$ otherwise, if possible. A summary of the f_{cut} values used in our simulations and a comparison of the number of sweeps performed to achieve convergence is shown in Tab. 3.2.

In systems such as these, with a non-trivial energy spectrum, the energy levels are not known initially. In this case, the algorithm must be run for many sweeps before any measurement phase of the simulation, in order to mark accessible energies. Our choice in Hamiltonian makes this burn-in phase somewhat easier, by restricting the possible accessible states to the interval of energy levels, $[0, -J|L|]$, where $|L|$ is the number of edges in the graph. Care should be taken that the observed energy levels include all possible energy levels. It may however be extremely difficult to access the lowest energy levels in antiferromagnetic systems which exhibit frustration. The Wang-Landau algorithm is convergent on the condition that each energy state has been visited approximately evenly. As a result, the run times vary as shown in Tab. 3.3.

Since the Wang-Landau algorithm is a Monte Carlo method, the DOS is estimated up to a constant; the unnormalised $g'(E)$ is proportional to $g(E)$. Since we estimate the natural log of the DOS for numerical reasons, $\ln g'(E) = \ln g(E) + \ln c$. The normalisation constant c can be calculated by demanding that the total number of states equals q^N, therefore

$$\ln c = \ln \left(\sum_i \exp[\ln g'(E_i)] \right) - N \ln q. \tag{3.4}$$

Even with the use of extended precision numerical libraries, it is undesirable to calculate the exponential sum inside the parentheses in Eq. 3.4. For this reason, we restrict the terms in the sum to the interval $[0, 1)$ by recognising that

$$\sum_i \exp[\ln g'(E_i)] = \exp[\ln g'_{\max}] \cdot \sum_i \exp[\ln g'(E_i) - \ln g'_{\max}], \tag{3.5}$$

where

$$\ln g'_{\max} \coloneqq \max_i \{\ln g'(E_i)\}. \tag{3.6}$$

On the RHS of Eq. 3.5, only the pre-factor is difficult to compute. It is now straightforward to extract the normalised DOS using

$$\ln c = \ln \left(\sum_i \exp[\ln g'(E_i) - \ln g'_{\max}] \right) + \ln g'_{\max} - N \ln q. \tag{3.7}$$

As a first look at degeneracy in wireless networks, it is of practical use to find the energy level E_0 that bounds 90% of interference configurations resulting from uniformly distributed spins on the network. The cumulative distribution function of interference states is

$$P(E_{\text{obs}} \leq E) = \sum_{E_{\text{obs}} \leq E} g(E_{\text{obs}}) \cdot q^{-N}, \tag{3.8}$$

where E_{obs} is the energy of a configuration randomly chosen with uniformly distributed spins. In terms of our numerical data,

$$P(E_{\text{obs}} \leq E) = \sum_{E_{\text{obs}} \leq E} \exp[\ln g(E_{\text{obs}}) - N \ln q], \tag{3.9}$$

where the terms in the sum are again restricted to the interval $[0, 1)$. The critical values E_0, defined as the largest values of E such that

$$P(E_{\text{obs}} \leq E) \leq 0.9, \tag{3.10}$$

are calculated using Eq. 3.9 for different graphs and q-state Potts models. The results are presented in Sec. 3.3.1.

It should be noted that there is some subtlety in controlling the errors of the Wang-Landau algorithm. According to Ref. [51], the statistical error in $\tilde{g}(E)$ scales like $\sqrt{\ln f_{\text{final}}}$ when f_{final} is close to one. It is also not necessary for the histogram, $h(E)$, to be perfectly flat to ensure convergence to the true DOS. The systematic errors due to autocorrelation between successive histogram updates can be reduced by using an $f_{\text{final}} \to 1$ and performing a number of updates between histogram measurements. We demand that our final modification factor is no larger than $f_{\text{cut}} = \exp[10^{-4}]$, that our

histogram $h(E)$ is approximately flat to achieve convergence and that we perform at least $|V|$ measurements between successive histogram and DOS updates in order to reduce the systematic errors.

To quantify the errors in our implementation of the Wang-Landau algorithm, we can compare results from toy models with some exact values from the literature. In the case of the square lattice, the density of states and ground state entropy of the $q = 2$ Potts model is exactly known. On the triangular lattice, the exact infinite volume ground state entropy is known. After performing simulations on the $|V| = 60^2$ square lattice using $f_{\text{cut}} = \exp[10^{-8}]$ and by comparing with the exactly known values, we found an average relative error of 0.02% in the DOS, averaged over the entire energy range. The exact DOS for this system is given by a Mathematica script from Ref. [52]. Estimates for the Wang-Landau errors were also found on the triangular lattice. The lowest energy state on the triangular lattice is composed of a large number of different frustrated spin microstates [45]. Due to the frustration, this system has a non-zero ground state entropy in the thermodynamic limit, resembling that of the interference networks. For the $q = 2$ Potts model on the $|V| = 60^2$ triangular lattice, the ground state specific entropy, $N^{-1} \ln \tilde{g}(E_{\min})$, was found to differ relatively by 0.072% compared with the exact value reported in Ref. [53], on an infinite volume lattice. In this case, we used $f_{\text{cut}} = \exp[10^{-8}]$.

f_{cut}	$\ln \tilde{g}(E_{\min})$	$\sigma(\ln \tilde{g}(E_{\min}))$	$N^{-1} \ln \tilde{g}(E_{\min})$	Relative Error (%)
$\exp[10^{-8}]$	170.69	0.01	0.3227	0.13
$\exp[10^{-7}]$	170.68	0.02	0.3227	0.13
$\exp[10^{-6}]$	170.29	0.09	0.3219	0.36
$\exp[10^{-5}]$	169.6	0.3	0.321	0.78
$\exp[10^{-4}]$	168.4	0.4	0.318	1.5

Table 3.1: This table shows the estimated ground state entropy of the $q = 2$ Potts model on a triangular lattice with $|V| = 529$ nodes. The ground state entropy, $\ln \tilde{g}(E_{\min})$, is averaged over four independent simulations and the standard error of the mean, $\sigma(\ln \tilde{g}(E_{\min}))$, is estimated using 100 bootstrap resamples. The exact specific ground state entropy, $\ln g(E_{\min}) = 0.323066$, is known from Ref. [53] so the relative error in our estimates in shown in percent.

For simulations performed on a triangular lattice with $|V| = 529$, similar to that of the interference networks, the estimated DOS and relative percentage error in ground state entropy as compared to the exact value is shown in Tab. 3.1. The table shows results from averaging over four independent DOS simulations with the standard error

of the mean, $\sigma(\ln \tilde{g}(E_{\min}))$, found using 100 bootstrap resamples. The variance in the ground state entropy is shown to be quite low for all values of f_{cut}. The relative error in the ground state entropy does increase for larger f_{cut} values, however, this relative error does not grow beyond 1.5%. While these errors can not be directly compared to those of the ground state entropy of the Potts model on our interference networks, since those exact values are unknown, we do expect similar behaviour. For this reason, errors are not quoted on our DOS estimates.

3.2.1 Thermodynamic Observables

The real strength of the Wang-Landau algorithm is that the multicanonical approach avoids some serious difficulties which arise in classic Monte Carlo sampling such as critical slowing down and loss of ergodicity due to high degree hubs. Fortunately, there are few disadvantages to using this robust method and almost all thermodynamic observables are accessible using the density of states. In addition, an estimate of the partition function can be directly calculated,

$$\tilde{Z} = \sum_{E} \tilde{g}(E). \tag{3.11}$$

This provides us with access to the Gibbs free energy [54], which was discussed in Chapter 2,

$$F(T) = -k_B T \ln \left(\sum_{E} \tilde{g}(E) \exp[-\beta E] \right). \tag{3.12}$$

Using the free energy, all thermodynamic potentials can be derived as in Sec. 2.2.

The internal energy, $\langle E \rangle$ and specific heat, C_v, can be calculated as follows;

$$\langle E \rangle(T) = \tilde{Z}^{-1} \sum_{E} E \tilde{g}(E) \exp[-\beta E] \tag{3.13}$$

and

$$C_v(T) = \frac{1}{T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right). \tag{3.14}$$

In this chapter, we use the peak in the specific heat to indicate a crossover from a spin-glass type phase to the paramagnetic phase. From Eq. 3.14, we see that the specific heat can be estimated to arbitrary precision in temperature and only the error

in estimating the density of states needs to be controlled.

3.3 Results and Discussion

This sections presents the results of our Monte Carlo simulations of the q-state Potts model on the θ complex networks. The simulations were performed using the Wang-Landau algorithm for values $q = 2$, 3, 4, 5 and 6 on these graphs.

3.3.1 Wang-Landau Results

This section examines the Wang-Landau simulations run-time and the resulting DOS estimates for the q-state Potts model embedded on the graphs described in Sec. 2.1.4. Independent simulations were performed using progressively lower f_{cut} values to reduce systematic errors. The absolute value of exponents of the lowest f_{cut} to converge are noted in parentheses in Tab. 3.2 as a function of system parameters θ and q. A larger exponent indicates a higher precision simulation. There is no clear trend in Tab. 3.2 to the number of sweeps taken to converge, but there are a few points of interest.

		Potts Model q-value				
		2	**3**	**4**	**5**	**6**
	1%	1.5 (5)	92.0 (5)	13.0 (5)	8.2 (5)	23.0 (5)
$\boldsymbol{\theta}$	**3**%	3.7 (8)	17.0 (8)	2.9 (4)	7.8 (4)	110.0 (8)
	5%	8.2 (8)	6.8 (5)	11.0 (5)	140.0 (5)	650.0 (4)

Table 3.2: Number of sweeps ($\times 10^7$) performed to reach convergence (magnitude of f_{cut} exponent in brackets).

The $\theta = 1\%$ graph simulations were slow to converge in general and $f_{\text{cut}} = \exp[10^{-5}]$ in the highest precision successful simulations. Convergence with a higher precision was possible for the $\theta = 3\%$ graph, with runtime generally increasing as q increases as shown in Tab. 3.3. However, convergence was very difficult for the $q = 4$ and 5 simulations. The runtime of many repeated simulations exceeded a week without converging for $f_{\text{cut}} = \exp[10^{-4}]$. Finally, for $\theta = 5\%$, the runtime increases as q increases. Due to this effect, a lower precision f_{cut} is used as q increases. The $\theta = 5\%$ and $q = 6$ simulations demonstrated poor convergence and very large runtime. Interestingly, the variance in runtime increases as θ increases and the largest overall runtime was recorded in the graph with least edges.

		Potts Model q-value				
		2	**3**	**4**	**5**	**6**
$\boldsymbol{\theta}$	**1**%	2.0	110.4	15.6	9.8	27.6
	3%	3.0	13.9	2.4	6.4	90.2
	5%	5.3	4.4	7.2	91.0	422.5

Table 3.3: Wang-Landau algorithm convergence time ($\times 10^3$ seconds).

The DOS of interference levels have been estimated for interference graphs constructed with θ thresholds of 1%, 3% and 5%. On each graph, the DOS of the q = 2, 3, 4, 5, and 6 Potts models are presented. The θ-dependence of the DOS of interference levels, (E/N), is shown in Fig. 3.1 for two values of available channels ($q = 3$ and 5). The corresponding CDF of interference, as defined in Eq. 3.9, is also shown. Additionally, Fig. 3.2 shows the q-dependence of the DOS and corresponding CDFs on the interference graphs with θ = 1% and 5%.

We first examine how the estimated DOS is affected by varying θ, for fixed q. Modifying θ does not affect the total number of configurations of the system, since both q and N are constant for a given graph. As shown in Figs. 3.1a and 3.1b, increasing θ contracts and shifts the degeneracy curve towards lower interference (E/N). Increasing θ also results in a less connected interference graph, which in turn decreases the interference (E/N) for any configuration. This is seen in the figures and in Tab. 3.4; the minimum interference decreases when θ increases for a given number of available channels (q). It should be noted that the true interference minima may be lower than these levels, due to the difficulty in finding the lowest energy state in frustrated systems.

We also observe an increase in the degeneracy of the minimum interference levels as θ increases, for a given number of available channels, as shown in Figs. 3.1a and 3.1b. Therefore, defining our networks to be more interference-tolerant by increasing the threshold θ, increases the number of configurations that contribute to the minimum interference level.

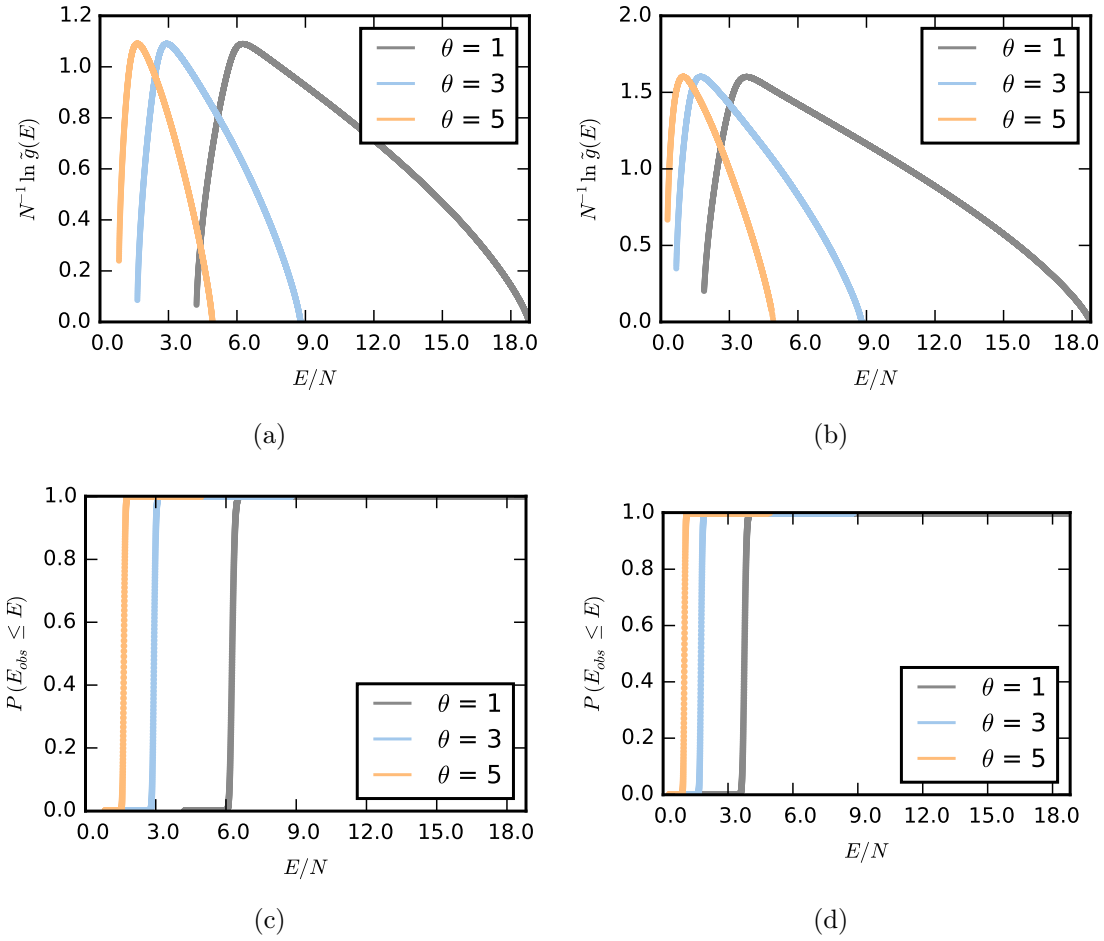


Figure 3.1: Degeneracy ($\ln \tilde{g}(E)$) vs. interference (E/N) using $q = 3$ (a) and $q = 5$ (b). Degeneracy CDF using $q = 3$ (c) and $q = 5$ (d).

		Potts Model q-value				
		2	**3**	**4**	**5**	**6**
	1%	7.594	4.222	2.692	1.893	1.393
$\boldsymbol{\theta}$	**3%**	3.242	1.641	1.013	0.677	0.469
	5%	1.694	0.832	0.482	0.291	0.164

Table 3.4: Lowest energy values found during the Wang Landau simulations (E/N). Note that these may not be the true ground states of the system, due to the spin-glass nature of the frustrated ground state.

Next, we discuss the effect on the DOS of varying q, for fixed θ. Figs. 3.2a and 3.2b show the DOS for a range of q, when $\theta = 1\%$ and $\theta = 5\%$ respectively. The degeneracy curves are shifted up and left as q increases. The maximum interference does not change

with the number of channels, since these microstates result from all nodes possessing the same spin which is not affected by q. The degeneracy does change, but only due to there being q degenerate maximum interference microstates. The total number of microstates also increases with q according to q^N and accounts for the greater volume under the DOS curves for greater q.

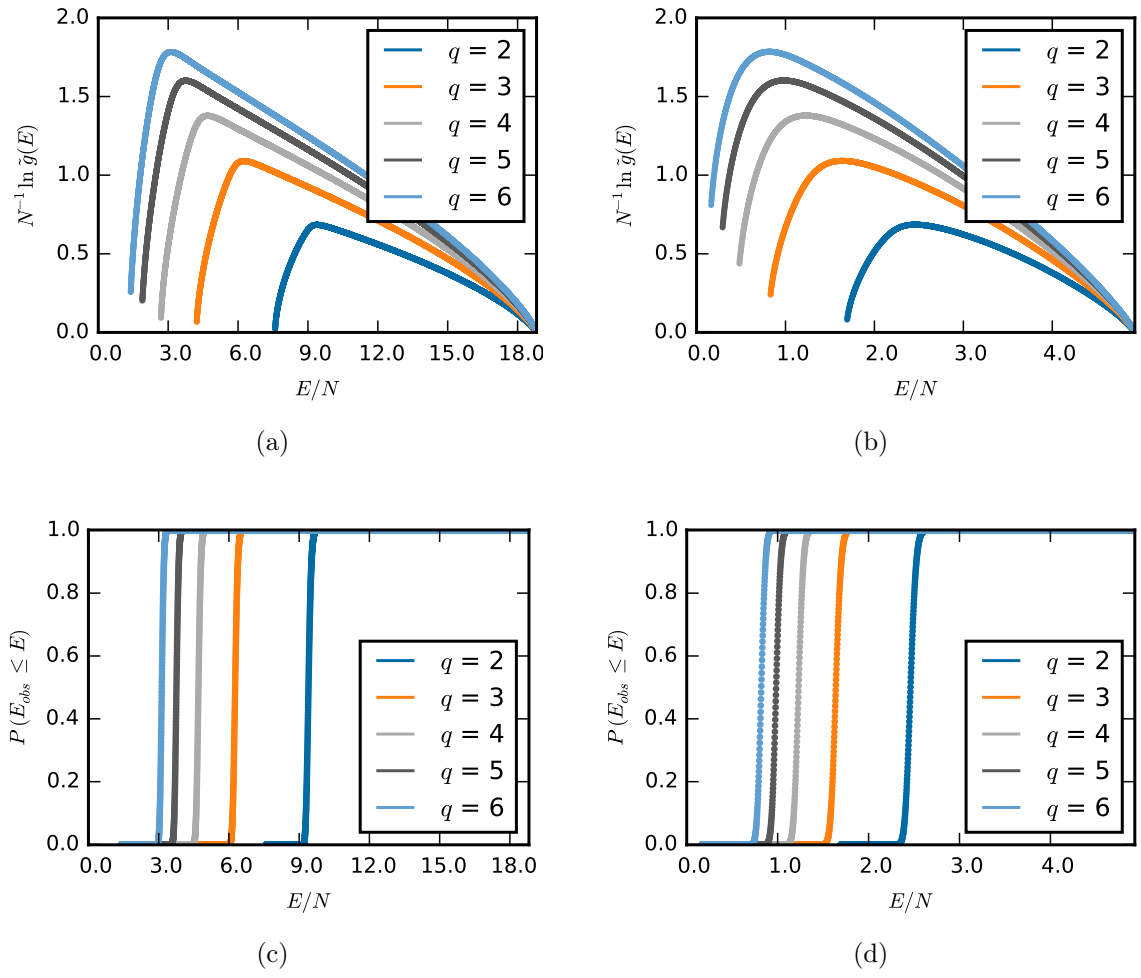


Figure 3.2: Degeneracy ($\ln \tilde{g}(E)$) vs. interference (E/N) on $\theta = 1\%$ (a) and $\theta = 5\%$ graphs (b). Degeneracy CDF on $\theta = 1\%$ (c) and $\theta = 5\%$ graphs (d).

As seen in Fig. 3.2, the minimum interference decreases when the number of available channels increases, for a given θ. The degeneracy of the minimum interference levels also increase. Therefore, the larger the number of available channels (q) the larger the number of configurations that exhibit minimum interference.

		Potts Model q-value				
		2	**3**	**4**	**5**	**6**
	1%	9.528	6.385	4.808	3.859	3.226
$\boldsymbol{\theta}$	**3**%	4.476	3.006	2.267	1.820	1.524
	5%	2.530	1.699	1.284	1.034	0.868

Table 3.5: Critical interference value (E_0) such that $P(E_{obs} \leq E_0) = 0.9$.

It is important to note that the peak of the degeneracy curves shifts towards lower interference both when the number of available channels increases and θ increases. The interference level corresponding to the largest number of configurations decreases both when we consider more interference-tolerant (larger θ) networks and when we include more channels. In Tab. 3.5, the interference level which bounds 90% of spectrum allocation configurations decreases as both θ and q increase. Moreover, all of the CDF plots are very steep; the majority of the states occur over a very short interference range. The interference which bounds 90% of the configurations is very close to that which bounds 10%, because certain interference levels have many orders of magnitude more configurations than others. Hence, by accepting marginally higher interference levels than the critical E_0 values in Tab. 3.5, we dramatically increase the probability of finding one of these configurations randomly by choosing spins with uniform distribution. By identifying the critical interference levels which bound the majority of configurations, we can identify an acceptable interference range. With this information, the task of distributed, dynamic spectrum allocation in wireless networks becomes significantly easier.

3.3.2 Critical Phenomena on the Interference Graphs

The antiferromagnetic Potts model has been shown to undergo a crossover from a spin glass type phase to a paramagnetic phase on complex networks driven by temperature. Having performed a classical Monte Carlo study of this system using the combined approach of the Metropolis and Wolff cluster algorithms from Chapter 2, our experience with these networks is that even this local/global spin update approach loses ergodicity due to the extremely long tunnelling times in this spin-glass type phase.

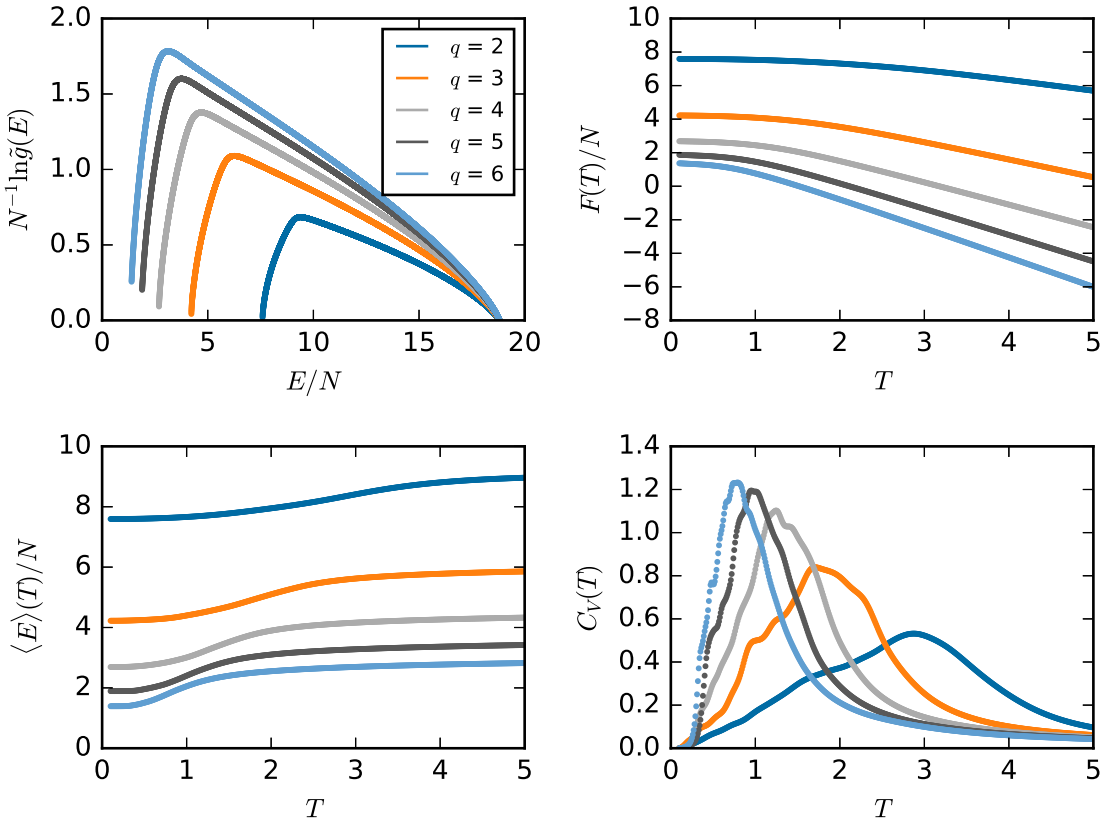


Figure 3.3: Plot of the $\theta = 1\%$ graph thermodynamic observables found using the Wang-Landau estimate for the density of states. The specific entropy ($N^{-1}\ln\bar{g}(E)$), free energy (F), internal energy ($\langle E\rangle$) and specific heat (C_v) are shown using 500 interpolated temperature values per curve.

The Wang-Landau algorithm provides a work around to this problem without having to perform replica exchange Monte Carlo. Having estimated the density of states for the three interference networks, we can finally discuss the critical behaviour of the Potts model. In Sec. 3.2.1, we have seen that an estimate for the density of states gives access to the free energy ($F(T)$), the average energy ($\langle E\rangle(T)$) and the specific heat ($C_v(T)$).

In Fig. 3.3, an increase in the temperature coincides with an increase in the internal energy. In the absence of a definable magnetisation, this rapid increase of the internal energy over a short temperature range is indicative of some critical behaviour. The rapid increase in internal energy is visible in both the $\theta = 3\%$ and 5% networks as well. For each graph, the internal energy decreases for a given temperature as q increases. This is presumably due to less frustration in the network.

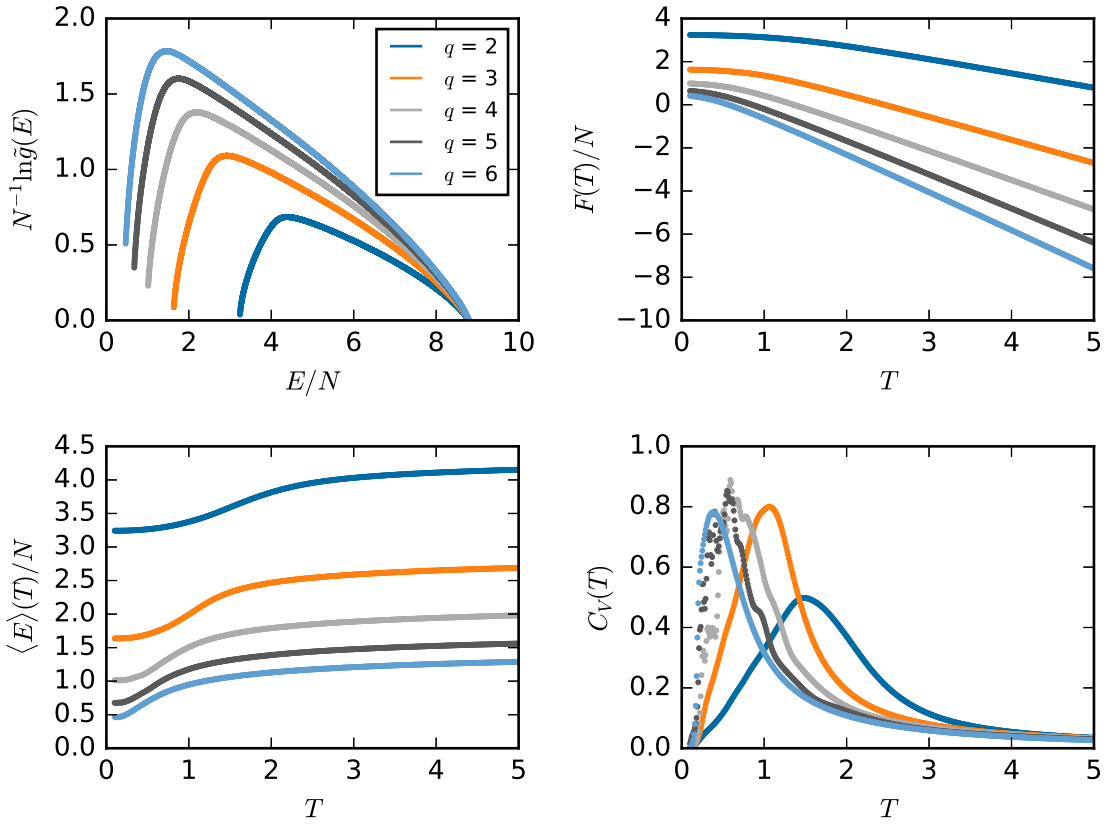


Figure 3.4: Plot of the $\theta = 3\%$ graph thermodynamic observables found using the Wang-Landau algorithm estimate for the density of states.

Since we expect to find some crossover between a spin-glass and paramagnetic phase, the position of the peak in the specific heat is quite important. In general, the peak in specific heat occurs at lower temperature as q increases and for less interconnected graphs with a larger θ value. Unfortunately, it becomes obvious that the Wang-Landau simulations which converge slowly and therefore have only been run up to a small f_{cut} exponent do not estimate the density of states precisely enough for a useful specific heat curve. In Fig. 3.3, all of these results use an $f_{\text{cut}} = \exp[10^{-5}]$. Even at this level, some artefacts (wiggles) can be seen in the specific heat curves.

In Fig. 3.4, the $f_{\text{cut}} = \exp[10^{-4}]$ specific heat curves stand out very obviously from the higher precision curves. These lower precision curves are not very useful for determining the critical temperatures. We can only comment on the general trend of the mass of the curve tending to lower temperatures as q increases. The higher precision curves are however very well behaved.

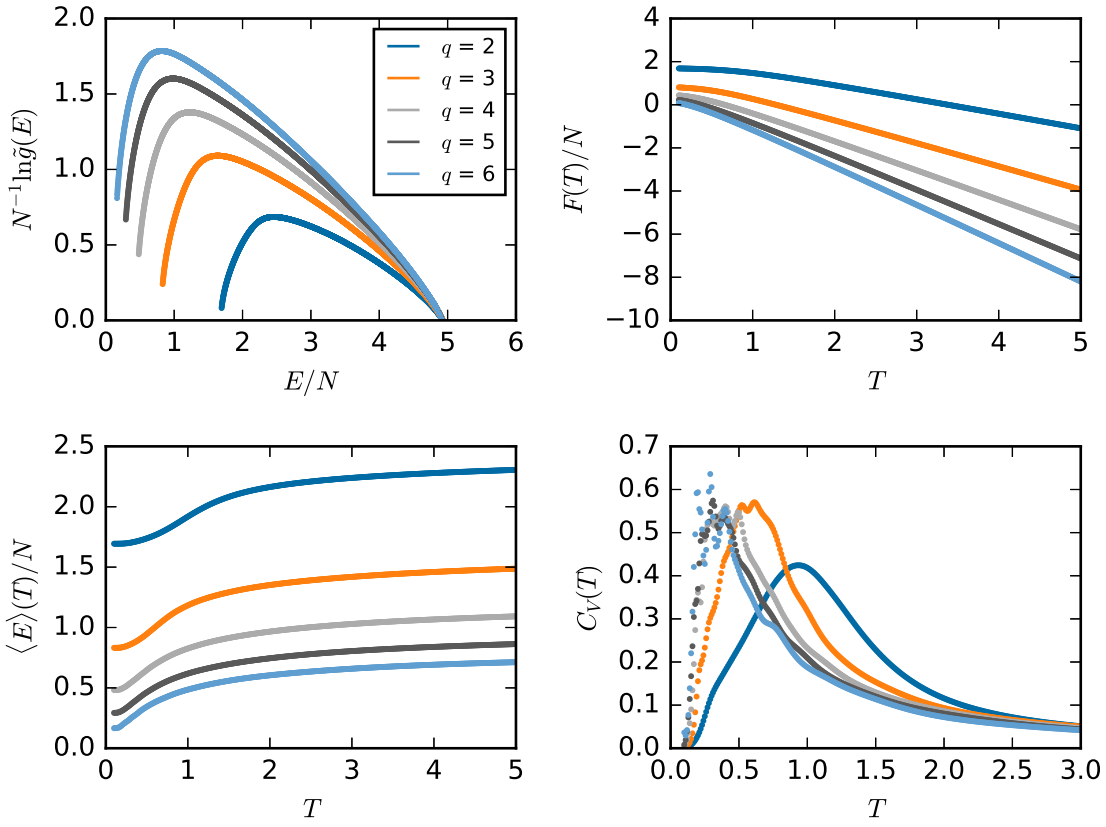


Figure 3.5: Plot of the $\theta = 5\%$ graph thermodynamic observables found using the Wang-Landau algorithm estimate for the density of states.

In Fig. 3.5, only the high precision $q = 2$ curve can be calculated without significant artefacts due to error in the density of states. The $q = 3, 4$ and 5 curves are not as clear as we would like and the lowest precision $q = 6$ curve gives very little information about the precise location of the critical behaviour.

While $f_{\text{cut}} = \exp[10^{-4}]$ is very useful to estimate the density of states to a relatively high precision, it seems to be insufficient for calculating thermodynamic observables.

3.4 Conclusion and Future Work

In this chapter, we demonstrate a new method for modelling interference in wireless networks. For this purpose, the antiferromagnetic Potts model is embedded on interference graphs. To the best of our knowledge, this is the first numerical estimate of the density of states on wireless networks. The Wang-Landau algorithm is used to perform these calculations. It is shown that different interference graphs and Potts models

clearly affect the degeneracy of the interference levels. In particular, there are changes to the value of the lowest interference levels, ground state entropy and the cumulative distribution function of the density of states.

The value of the lowest interference levels decrease in models with more available frequency channels and/or less dense interference graphs. The degeneracy of the lowest interference levels also increase under either/both of these conditions. Furthermore, the area under the density of states curves increase and are shifted towards lower interference values. We define a critical interference level marked by a rapid increase in the cumulative distribution function of the density of states. Having more frequency channels available and/or less dense interference graphs result in lower critical interference values. Therefore, under these conditions, the majority of spectrum allocation configurations result in lower interference values. Due to this behaviour, there is a higher probability that randomly sampled spectrum configurations result in low interference levels. This work provides some insight towards developing distributed spectrum allocation in wireless networks.

The antiferromagnetic Potts model on complex networks exhibits a crossover between a frustrated phase resembling a spin-glass and a paramagnetic phase. This spin-glass phase makes traditional Monte Carlo simulations exceptionally difficult to perform correctly. Using the Wang-Landau estimate for the density of states, we were able to calculate the free energy, internal energy and specific heat of the model. The critical temperature found using the peak in the specific heat decreases as q increases. In general, the lower the value of θ, the higher the critical temperature. This is similar to the behaviour from Chapter 2 that a more interconnected graph results in a higher critical temperature with a less sharp crossover. For networks which converge slowly, the resulting lower precision in the density of states had a notable effect on the specific heat curves. Unfortunately, these low precision simulations are not suitable for identifying the critical temperature.

In the future, it would be extremely beneficial to pursue a modification to the Wang-Landau to speed up convergence and reduce simulation times. A new technology such as this would allow for a more precise estimate of the density of states which is suitable to calculate the specific heat curves. An additional bonus to lower simulation times is that we could then perform a statistical error analysis of the Monte Carlo data, which at present is not feasible.

Chapter 4

Graph Rewiring

During the course of this work, the concept of dynamical networks has been discussed many times. Physical processes which can be described using graph theory often involve networks which evolve over time. For example, in a wireless interference graph, the edges can change due to load or failure. In biological systems, the network of interactions between genes, proteins and other biologically relevant molecules change due to the current needs of the cell [55].

Studying the properties of evolving networks usually involves taking a 'snapshot' of the physical network that is thought to be 'characteristic' of its general behaviour. Some properties of the network are then studied and in some cases this is repeated for a number of snapshots. However, this approach forbids us from asking questions about 'equilibrium' properties of a network ensemble, given some constraints such as the number of nodes ($|V|$) or edges ($|E|$) in a graph.

The next section briefly discusses the history of graph rewiring algorithms. Following this, we describe our approach to studying dynamic networks through rewiring using Markov Chain Monte Carlo. We propose a new algorithm to explore the configuration space of a network ensemble. Our rewiring algorithm performs 'local' rewiring updates and conserves the number of nodes and edges in the network. In Sec. 4.2, we provide the motivation behind our algorithm. In Sec. 4.3, we present our algorithm, prove that it is ergodic, describe the exact probability distribution of small graphs and an observable called the graph diameter. In Sec. 4.4, we describe our Monte Carlo simulations and present our estimated graph probability distributions and observable means for given graph ensembles. Finally, we discuss the scaling behaviour of the autocorrelation time of our algorithm and compare the estimated observables against literature values. We close this chapter in Sec. 4.5 with our plan for future work based

on rewiring.

4.1 Background

One of the earliest and most well-known network rewiring models is the Watts-Strogatz model introduced in Ref. [7] and discussed in previous chapters. This random graph construction model performs a rewiring of the edges incident to each node with a given probability p. The resulting graph has properties in between a regular lattice and a fully random graph. The target node of each rewired edge is chosen with uniform probability from the other nodes. This random graph model was introduced as the simplest possible method to build graphs which exhibit clustering and the small-world phenomenon. This is a significant result, since clustering and the small-world phenomenon are characteristic of very well-studied real-world networks. These two phenomena are present in the power-grid network, the neural network of *C. elegans* and the network of movie actors who have worked together [1]. However, the Watts-Strogatz model produces a single static graph and does not explore the entire graph configuration space.

To the best of our knowledge, the first Monte Carlo algorithm for the purpose of rewiring graph adjacency matrices appears in Ref. [56]. The algorithm randomly selects four matrix elements, $(i_1j_1, i_1j_2, i_2j_2, i_2j_1)$, which form a rectangular cycle in the adjacency matrix. The adjacent cycle entries must take alternating values, for example $(1, 0, 1, 0)$, with 1 corresponding to the presence of an edge $\{i, j\}$. The values of the adjacent cycle entries are then exchanged, proposing a new Monte Carlo state. Since the entries form a rectangle in the matrix, the sum over the rows and columns is invariant under this transformation and the degree distribution is preserved.

The number of rewires necessary for degree distribution preserving algorithms to produce independent graphs is not currently known, but a number of estimates have been given. A similar rewiring algorithm to Ref. [56] is proposed in Ref. [57] which accepts all proposed transitions. The procedure is known as the degree distribution (DD) preserving algorithm and is described below.

- Pick an endpoint u_1 with uniform probability and a uniform random neighbour u_2. This is the first edge.
- Pick a second endpoint, v_1 and uniformly choose an incident neighbour called v_2.
- Swap the edges (u_1, u_2) and (v_1, v_2) to create (u_1, v_2) and (v_1, u_2).

The authors in Ref. [57] show that the minimum relaxation time of the algorithm is at least $O(|E|^6)$. However, the bound is very wide and depends on an arbitrarily chosen precision ϵ: the distance between the true and observed stationary distributions.

In Ref. [58], a joint degree distribution (JDD) preserving algorithm is presented:

- Pick an endpoint u_1 with uniform probability and a uniform random neighbour u_2. This is the first edge.
- Pick a second endpoint, v_1, such that the degrees $k(u_1) = k(v_1)$ and uniformly choose an incident neighbour called v_2.
- Swap the edges (u_1, u_2) and (v_1, v_2) to create (u_1, v_2) and (v_1, u_2).

This JDD preserving algorithm is used to sample graphs with a joint probability matrix, $K(i, j)$, of the number of edges between vertices of degree i and j. The authors do not propose a theoretical bound on convergence.

Since a tight theoretical bound on the number of rewires necessary to produce independent graphs does not currently exist for either of these algorithms, a recent paper has proposed an effective rule of thumb which can be used instead [59]. The number of necessary rewires is estimated using the stationary property of the graph probability distribution and a chosen precision level ϵ, between the true and observed distributions. The authors prove that a graph with approximately independent edges and a specified degree distribution can be generated after $N_{\text{ind}} = \frac{1}{2}|E| \ln \epsilon^{-1}$ rewires. The same approximation for the JDD preserving algorithm requires $N_{\text{ind}} = |E| \ln \epsilon^{-1}$. In a graph containing roughly 10^3 nodes, $7.5|E|$ and $15|E|$ rewires are considered sufficient for the DD and JDD preserving algorithms respectively. For practical Monte Carlo simulations, this is an acceptable number of rewires for graphs of this size. The majority of edges become independent after this number of rewires, but a small number of edges are particularly persistent and need an order of magnitude more rewires to become independent.

The algorithms mentioned so far have used rewiring for graph formation or as a Monte Carlo method to sample the space of graphs with a given degree distribution or joint degree distribution. To the best of our knowledge, there have been no published examples of a Monte Carlo algorithm which samples from the space of graphs with a constant number of nodes and edges. Graphs of this kind can occur in sensor networks, where a number of sensors communicate to form a network which can change over time [60]. Furthermore, it may be necessary to estimate ensemble averages of observables which depend on both the graph structure and some other dynamical element

embedded on the graph, for example a Potts spin. With this in mind, we present a method to sample from the ensemble of graphs with a given $|E|$ and $|V|$. Our work to date has focused on the most basic graph ensemble; spanning trees of the complete graph K_n, which defines $|E| = n - 1$ and $|V| = n$. These graphs are connected and contain no loops, repeated edges or self-loops. Spanning trees are an important class of graphs, since they contain the minimum number of edges to keep the graph connected and there exists only one path between any pair of nodes.

4.2 Motivating the Algorithm

Our proposed Monte Carlo algorithm, which we outline in Sec. 4.3, samples from the configuration space of graphs with a fixed number of nodes and edges. This sampling depends on the graph probability distribution. The set of graphs of interest in this section are spanning trees of a complete graph. The complete graph with five nodes, K_5, is shown in Fig. 4.1. A spanning tree of K_5 is a subgraph that contains five nodes and four of the edges in K_5. Later, we will show that the graph probability distribution depends on the symmetries of the subgraphs. The graph probability distribution can

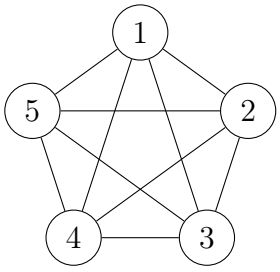


Figure 4.1: K_5 graph.

be used to estimate the expected value of observables which depend on graph structure;

$$\langle O \rangle = \sum_{G} O(g)\pi_g. \tag{4.1}$$

We use the notation throughout this chapter that G is the ensemble of graphs and g is a particular graph configuration. The graph observable of particular interest in this chapter is the diameter (d). In tree graphs, the diameter is the longest path. The diameter is very interesting because it depends on the entire structure of the graph, is bounded by the number of edges and is easily calculated. Since it depends on the overall graph structure, the diameter is a slowly changing observable compared to other graph observables. The integrated autocorrelation time (τ_{int}) of the diameter can therefore be

used to judge the efficiency of our Monte Carlo algorithm. This allows us to measure the number of rewires necessary to sample independent graphs. It should be noted however, that performing τ_{int} rewiring sweeps does not guarantee that all edges have been rewired.

4.2.1 Graph Ensemble

In order to describe a graph unambiguously, it is necessary to consider each node with an identifying label. Cayley's formula states that the number of spanning trees of a complete graph containing n nodes, with a label at each vertex, is n^{n-2}. Many of these graphs will have the same structure of adjacency and non-adjacency of nodes. Graphs with this property are isomorphic. Graph isomorphism defines an equivalence class of graphs, which may differ by their labelling structure.

Two simple graphs G and H are isomorphic if there is a bijection,

$$\Theta : V(G) \to V(H), \tag{4.2}$$

which preserves adjacency and non adjacency of vertices. Therefore each edge must obey,

$$\{u, v\} \in E(G) \iff \{\Theta(u), \Theta(v)\} \in E(H), \tag{4.3}$$

where $\{u, v\}$ is an edge in the graph G and $\{\Theta(u), \Theta(v)\}$ is the edge mapped to the graph H. Both graphs are isomorphic, $G \cong H$, if Eq. 4.3 is satisfied for all edges in the edge sets $E(G)$ and $E(H)$.

An automorphism is a permutation of the vertex set which preserves adjacency and non-adjacency between the graph and its image. It maps G to itself. Therefore, an automorphism, α, of a graph is an isomorphism between G and itself,

$$\alpha : V(G) \to V(G), \tag{4.4}$$

such that

$$\{\alpha(u), \alpha(v)\} \in E(G) \iff \{u, v\} \in E(G). \tag{4.5}$$

The set of automorphisms of a graph obey the following conditions:

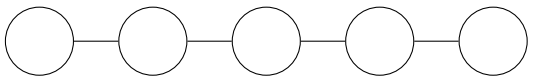
- If α and β are automorphisms of a graph, then their composition $\alpha \circ \beta$ is also an

automorphism.

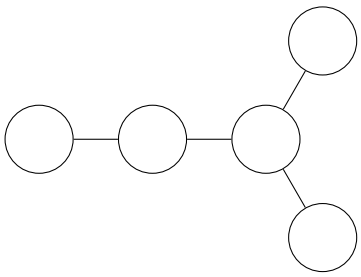
- Composition of automorphisms is always associative.
- The identity map is always an automorphism of a graph.
- Since automorphisms are bijections, if α is an automorphism of a graph, then the inverse, α^{-1} exists and is also an automorphism.

Since the set of automorphisms obey closure, associativity, identity and inverse, they form a group under composition of morphisms. The automorphism group of a graph is denoted $\mathrm{Aut}(g)$.

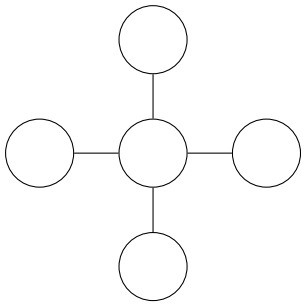
For small complete graphs, it is possible to identify and count all of the isomorphic spanning trees. Many of the isomorphic graphs share the same labelling, but differ by a symmetry operation. Such graphs are automorphic to each other.



(a) Linear graph.



(b) Fork graph.



(c) Star graph.

Figure 4.2: The three graph isomorphism classes of spanning trees of K_5. We have named the classes in order to refer to them with relative ease. In most cases, these are not commonly accepted graph names and are chosen to be descriptive.

The probability distribution of spanning trees depends on the number of graph labellings of each isomorphism class. The number of labellings of a class is

$$l(g_i) = \frac{n!}{|\mathrm{Aut}(g_i)|}, \tag{4.6}$$

where $|\mathrm{Aut}(g_i)|$ describes the size of the automorphism group of the class g_i. Summing over the number of labellings of each isomorphic graph class of spanning trees gives

the size of the ensemble:

$$\sum_{G} l(g_i) = n^{n-2}. \tag{4.7}$$

In practice, the equivalence classes g_i may be distinguished for small graphs by some convenient graph invariant eg. max degree. This can be seen in Figs. 4.2a, 4.2b and 4.2c. For larger graphs, this becomes a far more involved task.

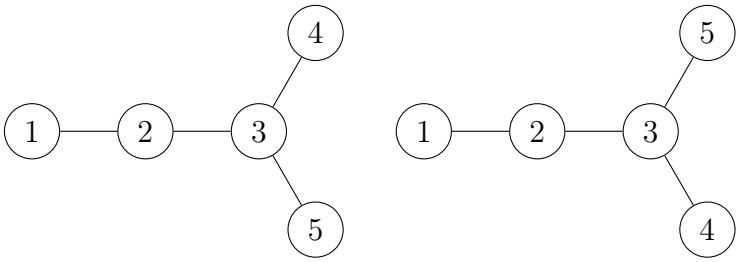


Figure 4.3: The two automorphisms of the Fork graphs in Fig. 4.2b. Fig. 4.3a is the result of the identity and Fig. 4.3b is the result of a rotation.

Conceptually, the automorphism group is the group of symmetries of a graph. A very symmetric graph has a large automorphism group. Complete graphs are the most symmetric by definition. The size of the automorphism group of a complete graph is $n!$, the proof of which is quite simple. Any permutation of the vertices will preserve the adjacency and non-adjacency of the graph, which means that the group has the same size as the set of permutations of n elements. Therefore, there are $n!$ unique permutations of K_n which are all automorphisms:

$$|\mathrm{Aut}(K_5)| = 5! = 120. \tag{4.8}$$

The stationary distribution of graphs π_g is governed by the rules of probability:

$$\pi_g \in [0, 1] \; \forall \; g \in G \tag{4.9}$$

$$\pi_g = \left(n^{n-2}\right)^{-1} \tag{4.10}$$

$$\sum_{G} \pi_g = 1 \tag{4.11}$$

The probability distribution of the equivalence class g_i is therefore given by

$$\pi_{g_i} = \sum_{g \in g_i} \pi_g. \tag{4.12}$$

As discussed above, the number of labelled graphs in each equivalence class is given by the labelling of the class, so the class probability is

$$\pi_{g_i} = \frac{l(g_i)}{n^{n-2}}, \tag{4.13}$$

where

$$l(g_i) = \frac{n!}{|\mathrm{Aut}(g_i)|}. \tag{4.14}$$

Before describing the graph rewiring algorithm, we will show how to calculate the exact probability distribution for spanning trees of K_5. The three equivalence classes of spanning trees of K_5, distinguished by their maximum degree are the Linear, Fork and Star graphs. The size of their respective automorphism group are

$$|\mathrm{Aut}(g_{\mathrm{Line}})| = 2, \tag{4.15}$$
$$|\mathrm{Aut}(g_{\mathrm{Fork}})| = 2, \tag{4.16}$$
$$|\mathrm{Aut}(g_{\mathrm{Star}})| = (n-1)! = 24. \tag{4.17}$$

The size of the automorphism groups of the Line and Fork graph can be easily determined by inspecting the morphisms which induce graph symmetries and including the identity morphism. The automorphism group of the star is not as trivial to determine. One way of determining this is to use the relation that the automorphism group of a graph is the same as its graph complement. The complement of a graph is obtained by removing all edges and placing edges where none existed. The complement of the Star graph is the union of K_4 and C_1; all radial nodes are adjacent to each other and the central node is a disconnected component of size one. The automorphism group of this graph is isomorphic to that of K_4, namely 4!. Therefore K_5 has an automorphism group of size 4!. The number of labellings of each graph equivalence class of K_5 can now be given as

$$l(g_{\mathrm{Line}}) = n!/2 = 60, \tag{4.18}$$
$$l(g_{\mathrm{Fork}}) = n!/2 = 60, \tag{4.19}$$
$$l(g_{\mathrm{Star}}) = n!/24 = 5. \tag{4.20}$$

This demonstrates that the sum of labellings of the three equivalence classes of spanning

trees of K_5 account for all of the labelled spanning trees;

$$l(g_{\text{Line}}) + l(g_{\text{Fork}}) + l(g_{\text{Star}}) = n^{n-2}. \tag{4.21}$$

4.3 Rewiring

In this section, we outline our Monte Carlo rewiring procedure and prove that the resulting Markov chain is ergodic. Therefore, we must establish that the Markov chain produced by the algorithm is irreducible, aperiodic, and positive recurrent. Following this, we discuss the large-graph limit of the graph diameter

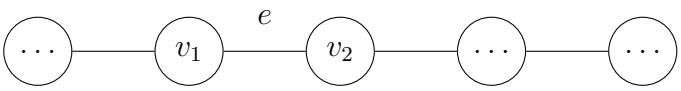
4.3.1 Rewiring Steps

The steps necessary to perform an update sweep of the graph involve;

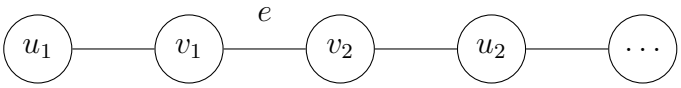
- Pick an edge, e_X randomly from the set of all edges with uniform probability. The nodes incident to this edge are the head (H) and tail (T) nodes.
- Form a set of the nodes in the neighbourhood of both H and T: $S = \{(N(H) \cup N(T)\} \setminus \{H, T\}$, where $N(H)$ is the set of vertices adjacent to H.
- Pick one node, M, with uniform probability from S as the node to be moved.
- Determine the head (H) and tail (T) from the endpoints of e_X so that $M \in N(H)$.
- Remove the edge between H and M, ($H \nsim M$).
- Create an edge between T and M, ($T \sim M$).

The initial configuration of the Markov chain is usually chosen to be the Linear graph such as in Fig. 4.2a for convenience. This initial configuration has no detrimental effect to our sampling because our Markov chain is universal and we carefully monitor the necessary thermalisation time before taking any measurements.

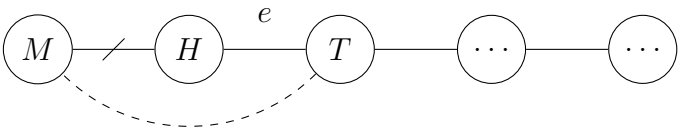
The number of edges in the graph remains constant as one edge is created and destroyed per rewire. Since we are sampling the ensemble of spanning trees with n nodes and $n-1$ edges, no cycles can be produced solely from a rewiring operation in this ensemble. The spanning trees of K_n are all connected, since there exists a path between every pair of nodes. Our algorithm preserves this connectivity by splitting the graph into two disconnected subgraphs when $H \nsim M$ and reconnecting them with $T \sim M$.



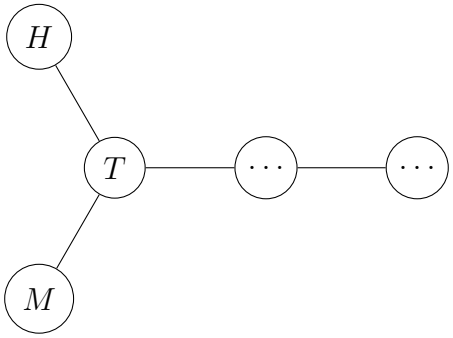
(a) Step 1: Choose e_X with uniform probability $p_e = |E|^{-1}$, from all edges in the graph.



(b) Step 2 & 3: Determine the neighbours of endpoints v_1 and v_2. In this case, they are labelled u_1, u_2, v_1, v_2. The set S is the union of these, excluding v_1 and v_2 themselves: $S = \{N(v_1) \cup N(v_2)\} \setminus \{v_1, v_2\}$. Pick the node to be moved, M, from S with uniform probability $p_M = |S|^{-1}$, where $|S|$ is the number of element in S.



(c) Step 4, 5 & 6: Determine the head node H to be the endpoint of the edge e_X which is incident to the node M, or $H \sim M$. Destroy the edge between H and M, $H \nsim M$. Create a new edge between T and M, $T \sim M$.



(d) This is the rewired graph after one iteration of our algorithm. Having performed one rewire, a graph-dependent observable is calculated and stored as the Markov chain history and for analysis at a later stage.

Figure 4.4: The graph rewiring procedure.

When rewiring the node M, that entire subgraph attached to it is rewired to T. Choosing a node with degree $k = 1$ for M results in only a single node being translated along the graph. All rewires are essentially a translation of the subgraph containing M but not H. If the chosen edge e_X has a degree $k = 1$ node as an endpoint, this node

will be incorporated into the graph. The degree of the head node always decreases and the degree of the tail node always increases upon successful completion of a rewire. In order to lower the degree of a node, it must be the head of a rewire step. The H vertex must have more than one neighbour (one neighbour other than T), otherwise the set S contains only neighbours of T. Fig 4.4 illustrates how the degree of the T node always increases while that of H always decreases.

Having defined the steps involved in the rewiring algorithm, it is necessary to prove that the resulting Markov chain converges to the desired probability distribution. The necessary properties to ensure convergence are discussed in the next section. It is also shown that the algorithm possesses these properties.

4.3.2 Ergodicity

Our rewiring algorithm produces a series of random graphs in which each successive graph is dependent on the previous graph. The probability of realising a graph y at time $t+1$, given graph x at time t is the transition probability:

$$P(X_{t+1} = y | X_t = x) = p_{xy}. \tag{4.22}$$

By construction, our algorithm satisfies this Markov property and the resulting graph sequence is a Markov chain.

A Markov chain is ergodic if it is irreducible, aperiodic and positive recurrent [15]. The fundamental limit theorem for irreducible Markov chains states that for any ergodic Markov chain, there exists a unique stationary distribution:

$$\pi_y = \lim_{n\to\infty} p_{xy}^{(n)}, \tag{4.23}$$

where $p_{xy}^{(n)}$ is the n-step transition probability

$$p_{xy}^{(n)} = P(X_{t+n} = y | X_t = x). \tag{4.24}$$

In order to estimate π-averaged observables by time-averaging over the successive graph configurations produced by our algorithm, we must prove that it is ergodic. We start by considering the accessibility of every pair of Markov chain states.

Irreducibility

A Markov chain is irreducible if for each pair of states, (x, y), there exists an $n \geq 0$ for which $p_{xy}^{(n)} > 0$. We must demonstrate that it is possible to transition from each graph state to every other graph state for our algorithm to produce an irreducible Markov chain. It is sufficient to show that every state can access the Linear graph and return, even if this transition occurs through many intermediate states.

Each vertex in a Linear graph has degree $k(v_i) = 2$, except the endpoints which have degree $k(v_{end}) = 1$. All other states in the space G of spanning trees of the complete graph K_n for $n > 3$ must have at least one vertex with a degree larger than two. Spanning trees with $n \leq 3$, results in a single graph equivalence class and do not need rewiring. Given an arbitrary spanning tree with $n > 3$, the transition to the Linear graph requires a degree lowering operation.

This transition is considered here on a Star graph. An edge incident to a node with $k(v_i) > 2$ is chosen with probability $p_e = 1/(n-1)$. The central node has degree, $k(v_i) = (n-1)$ and all other nodes have degree, $k(v_i) = 1$. Using our algorithm, the centre and one of the radial vertices are labelled as H and T respectively. A second vertex adjacent to H is chosen as the M vertex, with probability $p = 1/(k(H) - 1)$. By performing the rewiring in Fig. 4.5, the degree of H is lowered. The radial nodes can be repeatedly chosen as the T node. This lowers the degree of the central node until the max degree in the graph is two.

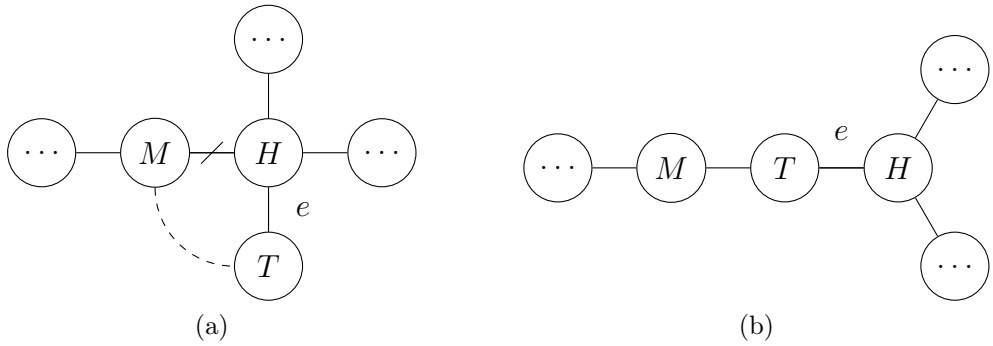
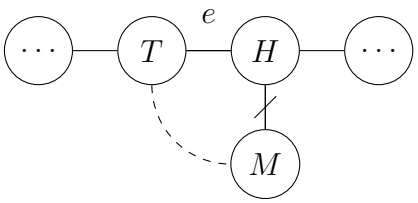


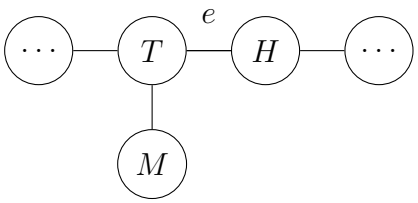
Figure 4.5: This transition lowers the degree of H. Nodes adjacent to the hub marked H are rewired and become part of the Linear subgraph containing T. Repeated application of this transition to all nodes with $k > 2$ transforms the graph into the Linear graph.

More generally, we can choose to rewire edges where $k(T) > 1$ also. The rewiring of the hub node will attach M to T, even though T is not a terminal vertex and this subgraph is not Linear. The rewiring is repeated and M can either be translated down

the T subgraph as in Fig. 4.6 or the T subgraph can be can be rewired onto M. This transforms this segment of the subgraph into the Linear graph. This is repeated for all H nodes with degree greater than two.



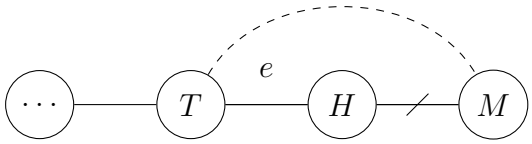
(a) A node M is rewired away from a hub H. This shifts M towards the end of the Linear subgraph attached to T.



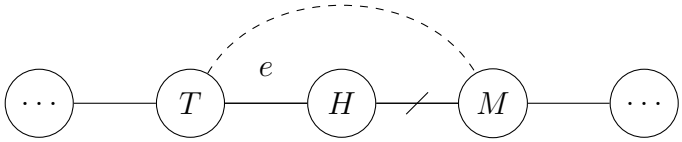
(b) M is shifted along the subgraph by a distance of one edge. Repeating this operation decreases the degree of any hubs and places the M-type nodes at the end of the Linear graph.

Figure 4.6: Translating a node M to the end of a Linear subtree.

Therefore, given any tree we can decrease the degree of any node with $k(v_i) > 2$ using two different methods. This can be repeated for all nodes with degree $k(v_i) > 2$, until we get the Linear graph.



(a) Rewiring at the end-node of a graph.



(b) Rewiring at a mid-point node of a graph.

Figure 4.7: Increasing the degree of a specific node to create a hub. The operation in Fig. 4.7b can be repeated to increase the degree of the node marked T here.

In order to transition from the Linear graph back to any other graph in the ensemble, we must be able to increase the degree of any node in any position. To demonstrate this, we create a Star graph from the Linear graph. It is now necessary to have a degree

increasing operation in addition to the translation operation. A degree increasing operation can be performed at any position in a subgraph. Fig. 4.7 illustrates a degree increasing rewire at the endpoint and at a position towards the middle of a subgraph. By performing a degree increasing operation at the endpoint of a graph and translating the resulting branch down the graph, we can create any spanning tree including the star graph.

So far we have demonstrated rewiring operations which can;

- lower the degree of a node,
- increase the degree of a node near the endpoint of graph,
- increase the degree of a node in middle of graph and
- translate a node of any degree to any position in tree.

Since these operations can be used to transition from any graph to the Linear graph and back, the Markov chain is irreducible. It should be noted that all of these transitions are performed using the same rewiring technique and the result of a rewire differs only depending on the neighbour structure of the edge chosen to rewire around.

Aperiodicity

The period of the state x, (o_x), is the greatest common divisor of the number of transitions n, such that $p_{xx}^{(n)} > 0 \ \forall \ n$. All states of an irreducible Markov chain have the same periodicity and any state x is aperiodic if $o_x = 1$. To prove that our Markov chain is aperiodic, we will demonstrate that a transition between the Linear and Star graphs exists in which $o_x = 1$.

Starting with the Linear graph, we will show that there exists two consecutive numbers, t and s, that satisfy the n-step transition back to state x such that $p_{xx}^{(n)} > 0$. Since two consecutive natural numbers have a greatest common divisors of one, the existence of t and s is sufficient to prove that our Markov chain is aperiodic.

A Star graph containing n nodes has a central hub node with degree of $n-1$. It is possible to transition from a Star graph to a Linear graph in $q = n-3$ operations by rewiring an edge incident to the central node $n-3$ times. This is shown in Fig. 4.8.

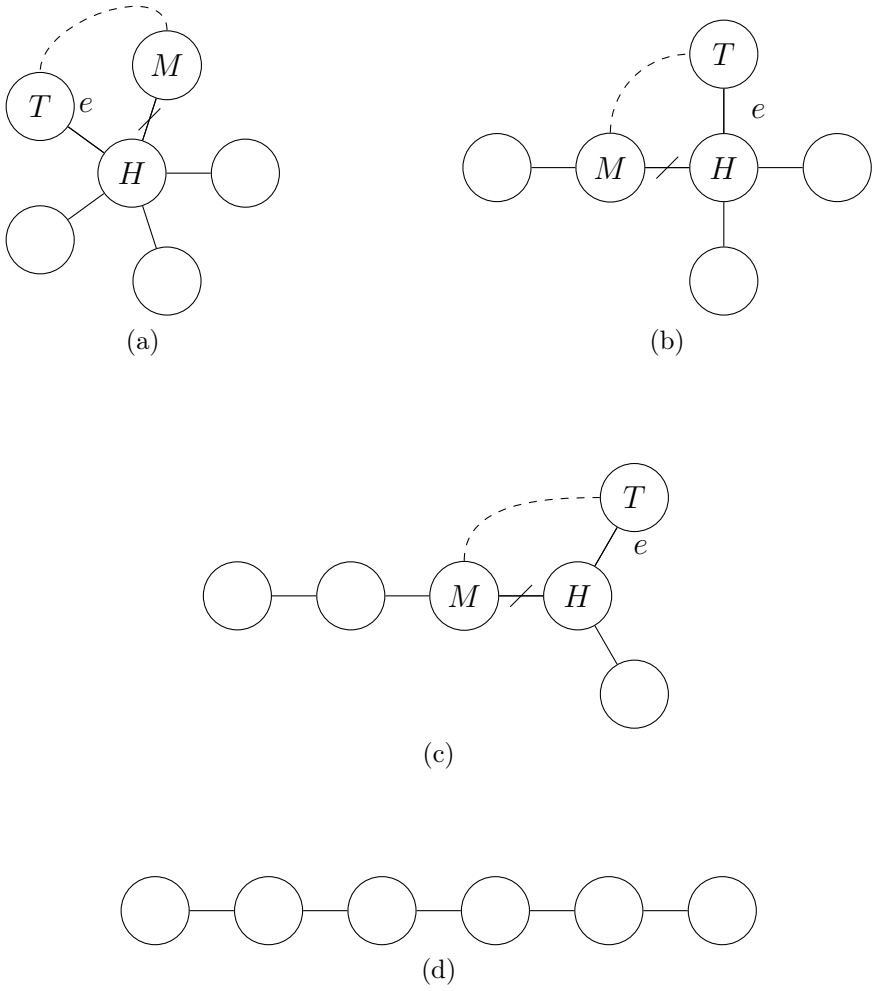


Figure 4.8: Transitioning from a Star to a Linear graph in $n-3$ operations.

The inverse transformation from the Linear graph back to the Star using $q = n-3$ operations also exists and is shown in Fig. 4.9.

This transformation can be considered using the n-step transition probabilities: $p^q_{\text{Star}\to\text{Linear}} > 0$ and $p^q_{\text{Linear}\to\text{Star}} > 0$. Define t so that

$$p^t_{StarStar} = (p^q_{\text{Star}\to\text{Linear}} \cdot p^q_{\text{Linear}\to\text{Star}}) > 0. \tag{4.25}$$

The number of rewiring operations necessary to perform this transformation is $t = 2q = 2(n-3)$ is even for all graph sizes.

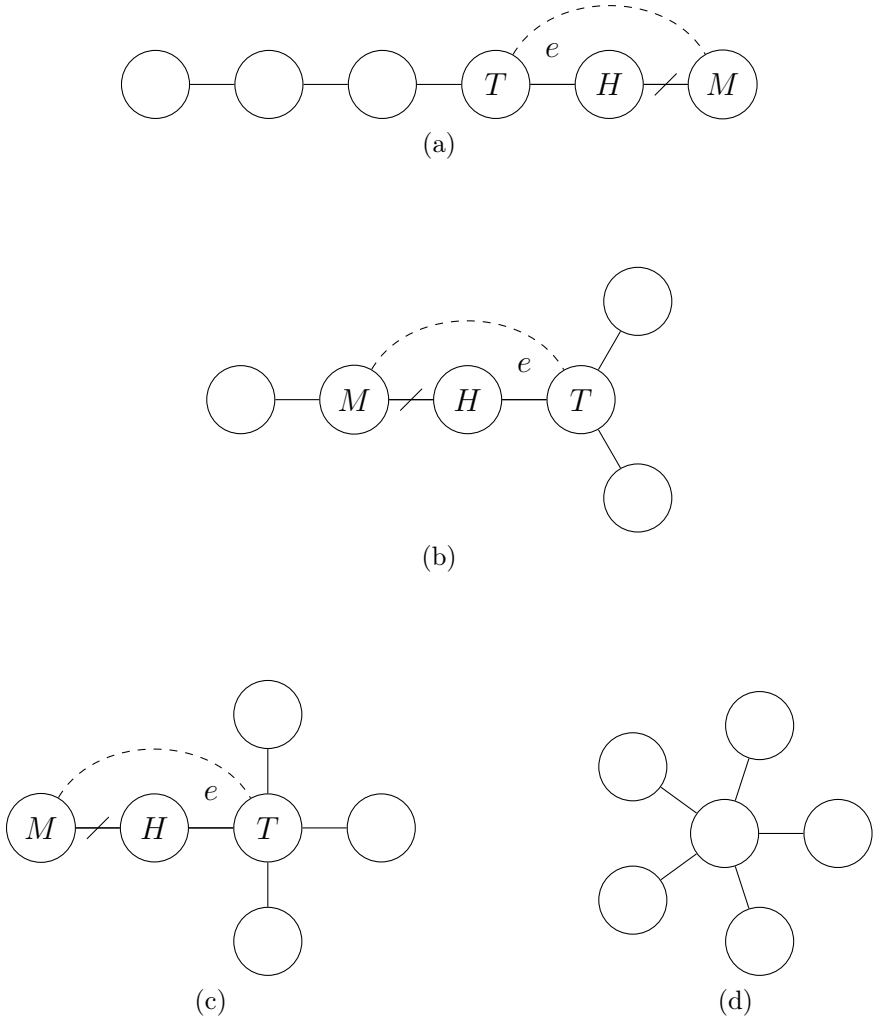
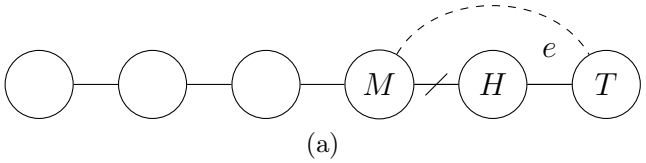


Figure 4.9: Transitioning from a Linear to a Star graph in $n-3$ operations.

It is possible to perform the round-trip from Star to Star in many other ways. We define the value of s by taking one of these other paths in state space. Perform the q operations from the Star to Linear graph just like above. However, we perform a different return sequence of transitions. Either perform the trivial rewire in Fig. 4.10, or a subgraph translation operation as in Fig. 4.6 in addition to the q necessary operations to transition from the Linear to the Star graphs. Overall, $r = q + 1$ operations are used to transition from the Linear graph back to the Star in this manner. Therefore, $p^s_{xx} > 0$ for $s = r+q = 2q+1$, which must be odd. The period of the Star graph is then given by the greatest common denominator of t and s. The period is one, since t and s are consecutive integers and must have greatest common divisor of one. Therefore our Markov chain is aperiodic.



(a)

Figure 4.10: Graph rewiring which maps the Linear graph back to itself.

Positive Recurrence

Finally, since any irreducible finite state Markov chain is positive recurrent, it follows that our algorithm satisfies the conditions necessary for ergodicity. Therefore, there exists a unique stationary distribution π and is given by

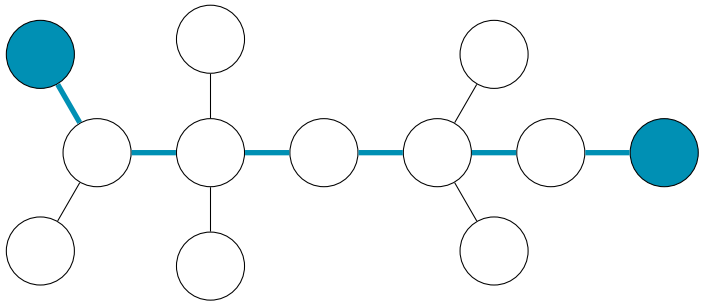
$$\pi_j = \lim_{n\to\infty} p_{ij}^{(n)}. \tag{4.26}$$

The stationary distribution is also universal, since it does not depend on the initial state of the system.

4.3.3 Graph Diameter

Having proven the ergodicity of our algorithm, we can use it to estimate the ensemble average of graph observables. The graph diameter d, of a spanning tree is defined as the longest path between any pair of vertices in. This observable depends on the overall graph structure and provides a measure of the maximum distance that information needs to travel in a wireless tree network.

Undirected spanning trees have two special properties: there exists only one path between any pair of nodes and any node can be chosen as a root of the graph. In order to calculate the graph diameter, we perform a breadth-first search to find the furthest node from a randomly chosen root node, r. The BFS algorithm searches over the immediate neighbours of r first. The neighbours are put in a queue to have their distance inspected in turn. Each of these nodes incident to r will have a distance of one. Each subsequent level of neighbours will have a distance of the current node plus one. Eventually, all nodes will have been visited and the furthest node v_1 from r is recorded.



(a)

Figure 4.11: Marked in blue are the endpoints and edges in the longest path. The diameter is the length of this longest path. The diameter depends on the structure of all nodes and edges in the graph. To calculate the diameter, the distance of all respective pairs of nodes must be tested. The diameter of a tree is calculated by performing a Breadth-First Search from any node, to identify one of the endpoints, v_1. By performing a second Breadth-First Search with v_1 as the root node, the second endpoint v_2 of the longest path is found to be the node furthest from v_1. The distance between v_1 and v_2 is the graph diameter. In this case, the longest path is not unique, but this does not change the diameter.

Since there exists only one path between any pair of nodes, the furthest node v_1 from the root r must be one of the endpoints of the longest path. There may be many nodes which are equidistant from r, in which case any one of these is a valid choice for v_1. To calculate the diameter, a second BFS is rooted at the node v_1, since the furthest node from an endpoint of the longest path is the other endpoint. The second endpoint v_2 is identified and the distance between them is returned by the BFS. In a spanning tree, the largest graph distance is the diameter.

Using our rewiring algorithm, we can estimate the expected value of the graph diameter over the graph ensemble,

$$\bar{d} \approx \langle d \rangle = \sum_{G} d(g_i)\pi_{g_i}. \tag{4.27}$$

Analytically, a very similar problem has already been addressed in Ref. [61]. In that paper, the authors focus on the height of a spanning tree, g, with ordered vertices, $P_1, P_2, \ldots, P_n$. The height of a tree, $h_{P_1}(g)$, is defined as the length of the longest path in g from an arbitrarily chosen root node P_1. An expression for the asymptotic distribution of the number of spanning trees with n vertices having exactly height, k, is found using a recursion relation. In the limit where n and k are large, the authors go on to calculate an expectation value of the height of trees starting from P_1 over the set of the n^{n-2} spanning trees with n nodes. It should be noted, the height of a tree

from a node P_1 is bounded by the diameter of that tree, since P_1 is not guaranteed to be an endpoint of the longest path,

$$\frac{1}{2}d(g) \leq \min_i h_{P_i}(g) \leq h_{P_i}(g) \leq \max_i h_{P_i}(g) \leq d(g). \tag{4.28}$$

The expectation value of the height of trees with root node P_1 as a function of the number of nodes $|V|$ given in Ref. [61] is

$$\langle h_{P_1} \rangle \approx \sqrt{2\pi|V|} = 2.50663\sqrt{|V|}. \tag{4.29}$$

In the next section, we will show our numerical estimate of the mean diameter using our algorithm over three orders of magnitude of graph size. The line of best fit of graph diameter versus graph size $|V|$ is calculated using the non-linear least squares method and compared with the analytical expectation value of tree height. Errors quoted for the parameters are estimated by the square root of the diagonals of the parameter covariance matrix.

4.4 Results and Discussion

Finally, in this section we present the results of our Monte Carlo simulations. We demonstrate that our algorithm samples from the graph distribution with a high level of precision in Sec. 4.4.1. For small spanning trees, we compare our Monte Carlo estimates of the graph distribution, $(\bar{\pi})$, with the exact distribution. In Sec. 4.4.2, we also compare the estimated mean of the graph diameter for small graphs with the expectation value. These estimates of the mean graph diameter and their statistical errors are compared for the K_4, K_5, K_6 and K_7 ensembles. In Sec. 4.4.2, we show that the integrated autocorrelation times of the graph diameter are under control as we increase the graph size and finally, the asymptotic behaviour of the graph diameter is discussed as the graph size increases in Sec. 4.4.2.

4.4.1 Graph Distribution

In this section, the Monte Carlo estimated probability distribution, $\bar{\pi}$, is presented and compared with the exactly known probability distributions of the K_7 ensemble. When working with spanning trees containing few nodes, the graph isomorphism classes can be identified by hand, as in Fig. 4.12. For graph ensembles larger than K_7, the high amount of possible branching makes it very difficult to identify all of the graph

isomorphism classes. The spanning tree isomorphism classes for this ensemble are shown in 4.12. The corresponding estimated probability distribution $\bar{\pi}_{g_i}$ is shown in Tab 4.1. Not shown in this section, but included in App. A are the graph isomorphism classes for the K_4, K_5 and K_6. Also in App. A are the graph probability distributions, errors and exact results in Tabs. A.1, A.2 and A.3.

The sampled probability distributions are found by histogramming the graph configurations over s samples, generated after performing a rewiring sweep. We perform m independent Monte Carlo replica to estimate the distributions $\pi^r_{g_i}$. The mean over these replica is taken and we use the bootstrap method to estimate the statistical error on this mean using B bootstrap resamples. The z-score shown in Tab. 4.1 clearly presents how far $\bar{\pi}_{g_i}$ is from the exact distribution π_{g_i} in units of the standard error.

Tree ($n^{n-2} = 16807$)	$\lvert\mathrm{Aut}(g_i)\rvert$	l_{g_i}	π_{g_i}	$\tilde{\pi}_{g_i}$	$\lvert z\rvert$
Line	2	2520	0.1499375	0.1499594(157)	1.39
Fork	2	2520	0.1499375	0.1499275(131)	0.77
Trident	6	840	0.0499792	0.0499930(083)	1.69
Pitchfork	24	210	0.0124948	0.0124985(044)	0.84
Star	720	7	0.0004165	0.0004170(007)	0.84
Handle	1	5040	0.2998751	0.2998690(170)	0.36
HandleFork	2	2520	0.1499375	0.1499251(148)	0.84
Pentane	8	630	0.0374843	0.0374749(070)	1.35
TriFork	12	420	0.0249896	0.0249928(056)	0.57
Tri	6	840	0.0499792	0.0499719(115)	0.63
Assym. Cross	4	1260	0.0749688	0.0749710(123)	0.18

Table 4.1: Sampled ($\bar{\pi}_i$) and exact (π) isomorphism class probability distribution of the K_7 ensemble. Over 70% of the z statistics are within one standard error and all are within two. $\chi^2 = 9.94$ for ten degrees of freedom. The probability of finding a χ^2 as large as this is $P = 0.55$. This value of χ^2, P and the distribution of the z statistics by the empirical normal distribution rule strongly indicate that $\bar{\pi}$ is sampled from π and that the errors are well under control.

To estimate $\bar{\pi}_{g_i}$, the rewiring algorithm was run for $s = 10^7$ sweeps. This was repeated for $m = 100$ simulations. The statistical standard error of the mean, which are shown in parentheses in Tab. 4.1, are estimated using $B = 10^6$ bootstrap resamples of the m independent probability distribution estimates.

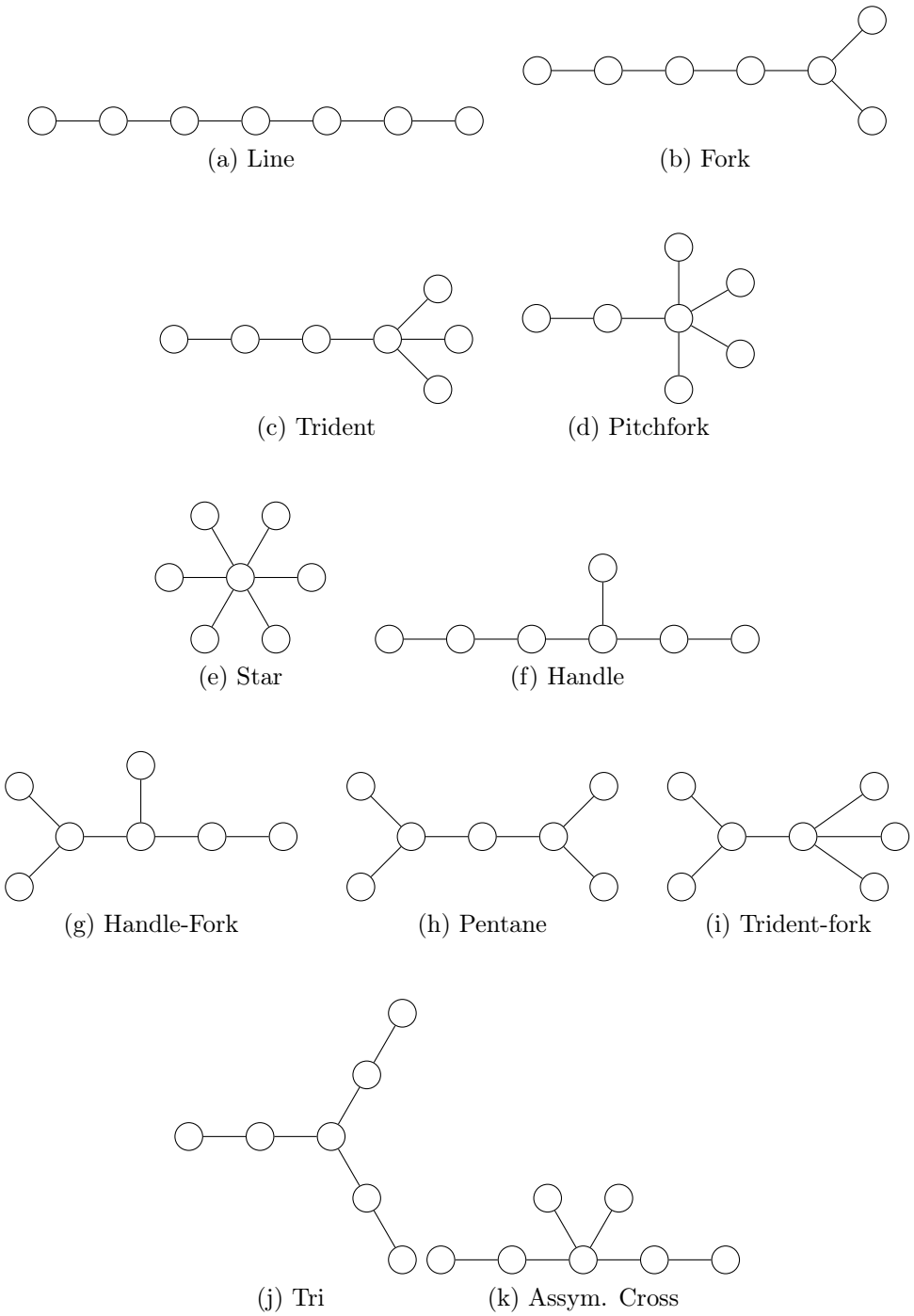


Figure 4.12: Non-isomorphic spanning trees of K_7. All spanning tree of K_7 are automorphic to one of these graphs.

The goodness of fit of $\bar{\pi}_{g_i}$ to the exact distribution π_{g_i} can be found using the χ^2

test. The χ^2 statistic in our case is,

$$\chi^2 = \frac{1}{N} \sum_{i=g_1}^{g_N} \left(\frac{\pi_i - \bar{\pi}_i}{\sigma_{\bar{\pi}_i}} \right)^2 . \tag{4.30}$$

Our null hypothesis is that the observed $\bar{\pi}_{g_i}$ is consistent with π_{g_i} and that any deviation between the two is purely by chance. This assumption is tested using P-values. The P-value is the probability of observing a χ^2 statistic as large as what we have calculated. For significance, we assume that if $P < 0.05$, then $\bar{\pi}_g$ and its standard error are very unlikely to be sampling from π_g.

As we can see in the caption of Tab. 4.1, the $\chi^2 = 9.94$ for ten degrees of freedom and 70% of z-scores are within one standard error. The estimated probability distribution is in very good agreement with the exact values and we can be confident that our rewiring algorithm is sampling from the distribution of K_7 correctly.

In this section, we have provided the first evidence that our algorithm can accurately estimate the graph probability distribution once the ensemble is small enough to identify every graph isomorphism class. The errors in the graph probability distribution can also be estimated with good precision. These results give us confidence in our algorithm and naturally lead to estimating the mean of graph observables.

4.4.2 Graph Diameter

Having demonstrated that we can accurately and precisely reproduce the graph distribution for small graph ensembles, the next step is to use our algorithm to estimate ensemble averages. The graph diameter was chosen as an interesting graph observable for a number of reasons; the diameter can be easily calculated on large graphs, it depends on the entire graph structure and therefore changes slowly due to local updates and an analytical value related to the graph diameter expectation value in the limit as graph size tends towards infinity exists [61].

In order to estimate the mean graph diameter with a small standard error, we first show that the integrated autocorrelation time (τ_{int}) is under control. Then we demonstrate that we can accurately estimate the graph diameter which agrees with the expected value of the graph diameter for small graph ensembles. Finally, we show our estimate of the fit of the mean graph diameter from our Monte Carlo experiments agrees with the bound on the height of trees given in Eq. 4.28.

Integrated Autocorrelation Time

The maximum graph diameter grows linearly with the graph size and changes slowly during graph rewires due to the local updating. It also gives a good description of the graph configuration that it is calculated for. Estimating τ_{int} of the diameter allows us to describe the efficiency of our algorithm in selecting structurally different graph configurations. To estimate τ_{int}, we used the method laid out in Sec. 2.2.2. The rewiring Monte Carlo algorithm was run for each graph ensemble and a total of 10^6 graph diameter values were calculated. To control for thermalisation errors, 10^5 rewiring sweeps were performed before starting the measurement phase.

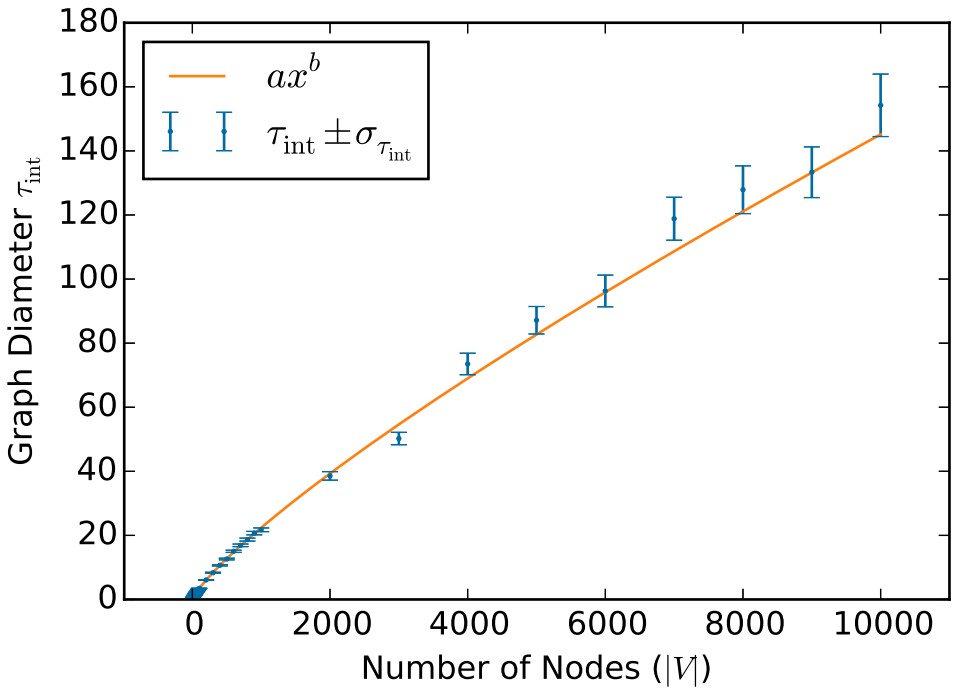


Figure 4.13: Log-log plot of the integrated autocorrelation time (τ_{int}) of the graph diameter versus number of nodes $|V|$, fitted to a monomial of the form $\tau_{\text{int}}(|V|) = 0.08233(76)|V|^{0.8116(21)}$ over three orders of magnitude. The fit uses all data points except the outlier at $|V| = 10$. The reduced χ^2 for 24 degrees of freedom is 0.93.

Fig. 4.13 shows the τ_{int} values and their error bars in blue, over three orders of magnitude of graph size. The line of best fit,

$$f(x, a, b) = ax^b, \tag{4.31}$$

was calculated using the Maximum Likelihood Estimation method and minimising,

$$\chi^2 = \sum_i \left(\frac{f(x_i, a, b) - y_i}{\sigma_i} \right)^2 , \tag{4.32}$$

using the least squares method.

The model function was not chosen based on some theoretical result, but rather from a *by-eye* initial fit and should only be used as a guideline. The reduced χ^2 indicates that our data gives an excellent fit to our model. The resulting monomial depends on the size of the graph and the exponent is below one. Therefore, as the system size increases through three orders of magnitude, τ_{int} is still very manageable. While it is not reasonable to directly make a comparison of our $\tau_{\text{int}} \approx |V|^{0.81}$ with the relaxation time calculated in Ref. [57] of $O(|E|^6)$, it does compare well with the $7.5|E|$ and $15|E|$ rewires necessary to produce independent graph configurations from Ref. [59] for graphs of 10^3 nodes.

Small graph diameter means vs exact.

Having shown that τ_{int} of the graph diameter is under control, we can now estimate the error for the graph diameter precisely. Each graph diameter mean in this section and the next was estimated using 10^6 samples measured after a rewiring sweep. The Markov chain was thermalised using 10^5 sweeps. When the Markov chain graph diameter history is inspected, the initial transient from low probability graph diameter configurations towards the mean takes many times less sweeps than this thermalisation time. The statistical analysis of the diameter mean and variance were performed using the binning method described in Ref. [17], using a bin size of 1000, which is much larger than τ_{int}. This results in uncorrelated means of each of the bins. The standard error of the sampling distribution of the sample mean was estimated using these 1000 binned means. These means were resampled using the 10^6 Bootstrap resamples and the standard deviation of these resamples was calculated.

Tab. 4.2 shows that our estimate of the graph diameter mean and standard error agree well with the exact diameter expectation value. The z-score for the K_4 and K_5 ensemble suggest that the statistical errors may be slightly underestimated, which is reflected in the reduced chi-square which is just larger than we would like. However, the results of the K_6 and K_7 mean diameters are in excellent agreement, suggesting that the larger z-scores should not be of too much concern.

Graph Ensemble	$\bar{d}$	$\langle d \rangle$	$\lvert z \rvert$
K_4	2.74944(44)	2.75	1.29
K_5	3.43920(53)	3.44	1.51
K_6	4.10671(68)	$4.106\dot{4}8\dot{1}$	0.33
K_7	4.71172(79)	4.711370	0.45

Table 4.2: Monte Carlo ensemble average estimates of graph diameter with standard error on the mean vs. exact graph diameter mean values. The reduced chi-square, $\chi^2_{\text{red}} = 1.4$ for three degrees of freedom.

Scaling

As discussed in Sec 4.3.3, the analytical expression for the mean height of trees is given as

$$\langle h_{P_1} \rangle(|V|) \approx \sqrt{2\pi |V|}, \tag{4.33}$$

and this value relates to the graph diameter, $d(g)$ since,

$$\frac{1}{2} d(g) \leq \min_i h_{P_i}(g) \leq h_{P_i}(g) \leq \max_i h_{P_i}(g) \leq d(g). \tag{4.34}$$

We therefore expect the form of the scaling behaviour of the graph diameter in trees to be very similar.

This behaviour is derived in the limit as the number of vertices approaches infinity. Fig. 4.14 shows our estimate of the graph diameter depending on $|V|$ in blue and a model function based on the expected graph diameter fitted in orange.

The close agreement between our numerical results and the model break down for small graph ensembles. With this in mind, we performed our fit using the data points for $|V| \geq 700$, however the fit line was calculated down to $|V| = 10$ using the fit parameters. Even at small graph sizes, where the fit breaks down, there is good agreement between the data and model function. The model function that we used is

$$f(x, a, b, c) = ax^b + c, \tag{4.35}$$

and

$$\chi^2 = \sum_i \left(\frac{f(x_i, a, b, c) - y_i}{\sigma_{SE}} \right)^2, \tag{4.36}$$

where the errors for each data point were estimated using the binning method and bootstrap resampling. This model was chosen since it reflects the form of the expectation value of the height of trees which differs from the mean diameter by a linear term between one and a half as shown in Eq. 4.28. The additive constant in our fit function collects sub-leading terms of order $O(|V|^{-1})$ or smaller in the height of trees expectation value which result from the use of Stirling's formula. The sub-leading terms tend to zero as the size of the graphs increase.

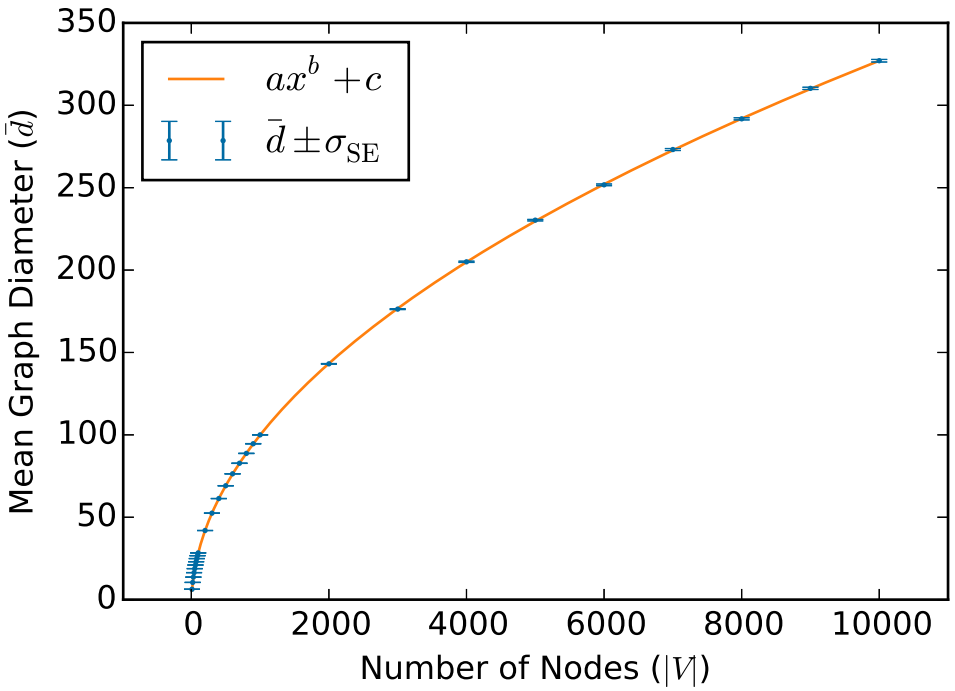


Figure 4.14: Log-log plot of the mean graph diameter $\bar{d}$ versus number of nodes $|V|$, fitted to a function of the form $\bar{d}(|V|) = 3.2(1)|V|^{0.504(3)} - 4.0(9)$. The fit is performed on the data points where $|V| \geq 700$. The reduced χ^2 for 10 degrees of freedom is 1.02.

The value of the fit parameter $b = 0.504(3)$ is found to be very close to the exponent of the expected value of 0.5. Our model results in a reduced chi-square of 1.02, with the residuals between our data and the model randomly scattered inside two standard deviations from the mean. To the best of our knowledge, our numerical results provide the most precise estimate of the mean of the graph diameter in spanning trees with $|V|$ nodes.

4.5 Conclusion and Future Work

In this chapter, we have outlined our rewiring algorithm which can be used to sample over the configuration space of spanning trees with $|V|$ nodes. The resulting Markov chain is shown to be ergodic, which makes it useful to calculate ensemble statistics. The graph probability distribution is estimated for small graphs and shown to agree very well with the exact distribution.

The graph diameter, which describes the structure of graph configurations is then investigated. The integrated autocorrelation time of the graph diameter is shown to scale like $\tau_{\text{int}} \approx |V|^{0.81}$, which means that the estimates of the diameter mean and variance can be calculated even for large graphs. It is then shown that the form of the analytical expectation value of the height of trees and our numerical results agree and the graph diameter scales like $\bar{d} \approx |V|^{0.5}$. The goodness of fit, given by the reduced chi-square suggests that our choice of model is appropriate and validates the form of the height of trees expectation value. There is some discrepancy between our data and model for small graph size, but this is most likely due to finite volume effects which are avoided by using data from trees with 700 nodes or more to perform the fit.

In terms of future work stemming from this chapter, we see this as early work and endeavour to study graph ensembles for arbitrary graph types, not only spanning trees. An important contribution to this area would be the ability to rewire graphs with a flexible number of nodes and edges. Furthermore, we are quietly confident that we can use the methods outlined in this chapter to study dynamic models embedded on dynamic networks. In many situations, it may be of interest to calculate ensemble averages over the joint probability distribution of graph and spin configurations.

Chapter 5

Concluding Remarks

In Chapters 2 and 3, the behaviour of the ferromagnetic Ising model and antiferromagnetic Potts model on complex interference networks has been shown to deviate from that expected on a lattice. This deviation is strongest on the most interconnected $\theta = 1\%$ graph, which differs the most from a lattice among the networks that we have studied. In Chapter 4, we presented a new Monte Carlo algorithm which samples from the probability distribution of trees with a fixed number of nodes and edges. The algorithm was shown to choose independent graph configurations efficiently even as the size of these graphs increase.

By performing Monte Carlo simulations on these empirical complex networks, which have no analytical description, a very rich collection of phenomena has been observed. The ferromagnetic Ising model is seen to exhibit a crossover from an ordered to a disordered phase driven by temperature. The antiferromagnetic Potts model appears to display spin glass behaviour at low temperature, which up to recently has only been observed in models containing mixed ferromagnetic and antiferromagnetic bonds between nodes. Again, driven by temperature, a crossover to a disordered paramagnetic phase is indicated by estimating the thermodynamic observables of the system. The density of states is also shown to be useful to describe the interference configurations which can occur in a dynamic, distributed frequency allocation model on a wireless communication network. In Chapter 4, by using simple, local updates to the graph, our rewiring algorithm effectively samples a complex state space of graphs.

As these examples show, the behaviour of complex networks provides an extremely rich research area. The interdisciplinary nature of network science exposes a very wide audience, with vastly different skills and research questions to the same types of problems. In my opinion, this will cause the rise in popularity of network science to

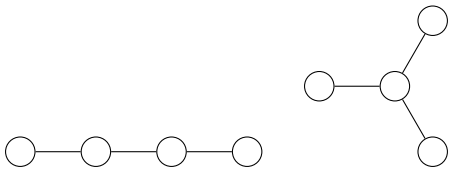
continue for the foreseeable future as different fields apply the insights of others to their own work. Overall, complex networks are a very fertile ground for new Monte Carlo methods, with the opportunity to study complex physics in systems with simple rules of interaction that could have a profound effect in unforeseen applications.

Appendix A

Graph probability distributions

In this appendix, the Monte Carlo results of our rewiring algorithm are presented which did not fit in Sec. 4.4.1.

A.1 K_4 Ensemble



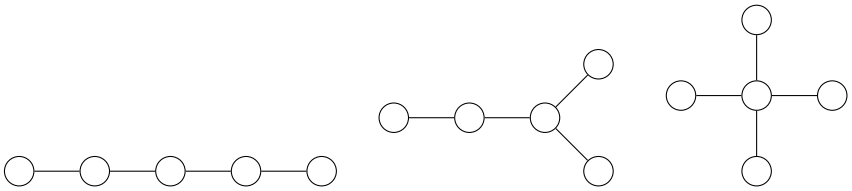
(a) Line (b) Star

Figure A.1: These graphs are the non-isomorphic spanning trees of K_4. All spanning trees of K_4 are automorphic to one of these graphs.

Tree ($n^{n-2} = 16$)	$\lvert\mathrm{Aut}(g_i)\rvert$	l_{g_i}	π_{g_i}	$\tilde{\pi}_{g_i}$	$\lvert z\rvert$
Line (Fig. A.1a)	2	12	0.75	.7500009(102)	0.09
Star (Fig. A.1b)	6	4	0.25	.2499991(102)	0.09

Table A.1: The sampled ($\bar{\pi}_{g_i}$) and exact (π_{g_i}) isomorphism class probability distribution of the K_4 ensemble. The z-statistic shows that the sampled probability distributions is very close to the expected value with far less than one σ in the difference. The $\chi^2 = 0.02$ for one degree of freedom, as described in Eq. 4.30. The probability of finding a χ^2 as large as this is $P = 0.11$.

A.2 K_5 Ensemble



(a) Line (b) Fork (c) Star

Figure A.2: Non-isomorphic spanning trees of K_5.

Tree ($n^{n-2} = 125$)	$\lvert\mathrm{Aut}(g_i)\rvert$	l_{g_i}	π_{g_i}	$\tilde{\pi}_{g_i}$	$\lvert z\rvert$
Line (Fig. A.2a)	2	60	0.48	.4800174(177)	0.98
Fork (Fig. A.2b)	2	60	0.48	.4799775(151)	1.62
Star (Fig. A.2c)	24	5	0.04	.0400051(061)	0.82

Table A.2: Sampled ($\bar{\pi}_i$) and exact (π) isomorphism class probability distribution of the K_5 ensemble. Two thirds of the z statistics are within one standard deviation, indicating that the $\bar{\pi}$ means are normally distributed and that the errors are reasonable. $\chi^2 = 3.87$ for two degrees of freedom. The probability of finding a χ^2 as large as this is $P = 0.86$. The χ^2 is a bit large, but there is still a good likelihood that $\bar{\pi}$ is sampled from π.

A.3 K_6 Ensemble

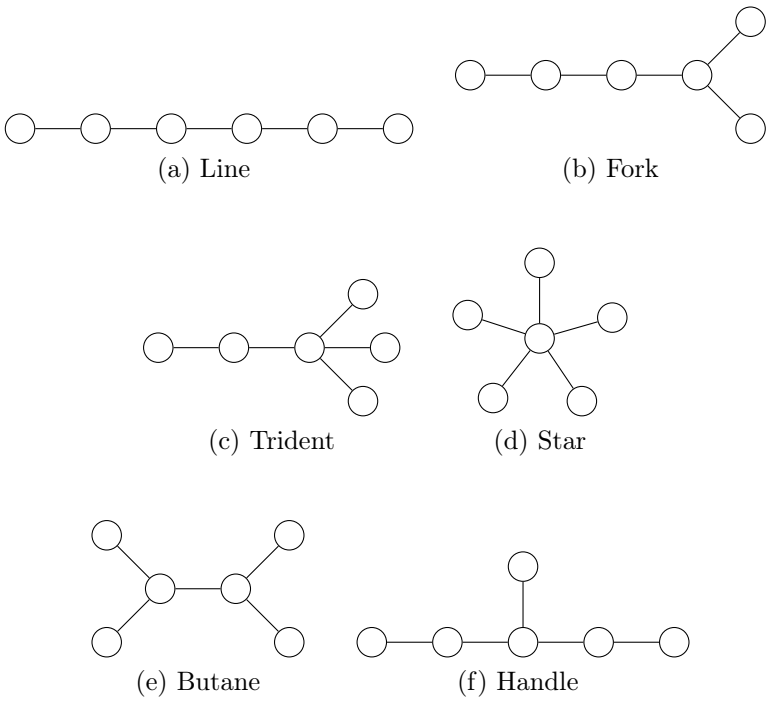


Figure A.3: Non-isomorphic spanning trees of K_6.

Tree ($n^{n-2} = 1296$)	$\lvert\mathrm{Aut}(g_i)\rvert$	l_{g_i}	π_{g_i}	$\tilde{\pi}_{g_i}$	$\lvert z\rvert$
Line (Fig. A.3a)	2	360	$0.2\dot{7}$	.2777699(204)	0.39
Fork (Fig. A.3b)	2	360	$0.2\dot{7}$	.2777847(137)	0.5
Trident (Fig. A.3c)	6	120	0.0925926	.0925895(111)	0.27
Star (Fig. A.3d)	120	6	0.0046296	.0046303(023)	0.29
Butane (Fig. A.3e)	8	90	$0.069\dot{4}$	.0694618(080)	2.17
Handle (Fig. A.3f)	2	360	$0.2\dot{7}$	.2777638(200)	0.7

Table A.3: Sampled ($\bar{\pi}_i$) and exact (π) isomorphism class probability distribution of the K_6 ensemble. The majority of the z statistics are within one standard deviation, however $\bar{\pi}$ of the Butane class is larger than we would expect to see randomly among six categories. $\chi^2 = 5.77$ for five degrees of freedom. The probability of finding a χ^2 as large as this is $P = 0.66$. The χ^2 is slightly big, but still reasonable. Considering that the χ^2 statistic is extremely sensitive to outliers, this shows a good likelihood that $\bar{\pi}$ is sampled from π.

Bibliography

[1] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47–97, Jan 2002.

[2] F. Wang and D. P. Landau, "Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States," *Physical Review Letters*, vol. 86, p. 2050, Mar 2001.

[3] N. McBride, J. Bulava, C. Galiotto, N. Marchetti, I. Macaluso, and L. Doyle, "Degeneracy Estimation in Interference Models on Wireless Networks," *Physica A: Statistical Mechanics and its Applications*, vol. 469, pp. 540 – 550, 2017.

[4] F. Y. Wu, "The potts model," *Rev. Mod. Phys.*, vol. 54, pp. 235–268, Jan 1982.

[5] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Critical phenomena in complex networks," *Rev. Mod. Phys.*, vol. 80, pp. 1275–1335, Oct 2008.

[6] R. J. Trudeau, *Introduction to Graph Theory.* New York: Dover Publications, 1993.

[7] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[8] J. Viana Lopes, Y. G. Pogorelov, J. M. B. Lopes dos Santos, and R. Toral, "Exact solution of ising model on a small-world network," *Phys. Rev. E*, vol. 70, p. 026112, Aug 2004.

[9] A. Barrat and M. Weigt, "On the properties of small-world network models," *European Physical Journal B*, vol. 13, pp. 547–560, Jan 2000.

[10] R. H. Swendsen and J.-S. Wang, "Nonuniversal critical dynamics in monte carlo simulations," *Phys. Rev. Lett.*, vol. 58, pp. 86–88, Jan 1987.

[11] Leone, M., Vázquez, A., Vespignani, A., and Zecchina, R., "Ferromagnetic ordering in graphs with arbitrary degree distribution," *Eur. Phys. J. B*, vol. 28, no. 2, pp. 191–197, 2002.

[12] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Ising model on networks with an arbitrary distribution of connections," *Phys. Rev. E*, vol. 66, p. 016104, Jul 2002.

[13] A. Aleksiejuk, J. A. Hołyst, and D. Stauffer, "Ferromagnetic phase transition in Barabási-Albert networks," *Physica A: Statistical Mechanics and its Applications*, vol. 310, no. 1-2, pp. 260–266, 2002.

[14] G. Bianconi, "Mean field solution of the Ising model on a Barabási-Albert network," *Physics Letters A*, vol. 303, no. 2-3, pp. 166 – 168, 2002.

[15] A. Sokal, "Monte carlo methods in statistical mechanics: Foundations and new alogorithms," *Lectures at the Cargèse Summer School*, 1996.

[16] U. Wolff and Alpha Collaboration, "Monte Carlo errors with less errors," *Computer Physics Communications*, vol. 156, pp. 143–153, Jan 2004.

[17] B. A. Berg, "Introduction to Markov Chain Monte Carlo Simulations and their Statistical Analysis," *eprint arXiv:cond-mat/0410490*, Oct 2004.

[18] U. Wolff, "Collective monte carlo updating for spin systems," *Phys. Rev. Lett.*, vol. 62, pp. 361–364, Jan 1989.

[19] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, pp. 1–26, Jan 1979.

[20] B. Efron and R. Tibshirani, "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy," *Statistical Science*, vol. 1, no. 1, pp. 54–75, 1986.

[21] P. Hall, "Resampling a Coverage Pattern," *Stochastic Processes and their Applications*, vol. 20, no. 2, pp. 231–246, 1985.

[22] E. Carlstein, "The Use of Subseries Values for Estimating the Variance of a General Statistic from a Stationary Sequence," *The Annals of Statisitcs*, vol. 14, no. 3, pp. 1171–1179, 1986.

[23] "Simulation Methods in Physics Lectures: Reweighting, Jackknife, Bootstrap." `http://www.helsinki.fi/~rummukai/simu/reweight.pdf`. Accessed: 2016-10-27.

[24] A. M. Ferrenberg and R. H. Swendsen, "New monte carlo technique for studying phase transitions," *Phys. Rev. Lett.*, vol. 61, pp. 2635–2638, Dec 1988.

[25] R. B. Potts and C. Domb, "Some generalized order-disorder transformations," *Proceedings of the Cambridge Philosophical Society*, vol. 48, p. 106, 1952.

[26] J.-P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A.-L. Barabási, "Structure and tie strengths in mobile communication networks," *Proceedings of the National Academy of Sciences*, vol. 104, pp. 7332 – 7336, May 2007.

[27] C. A. Hidalgo and C. Rodriguez-Sickert, "The dynamics of a mobile phone network," *Physica A: Statistical Mechanics and its Applications*, vol. 387, pp. 3017–3024, May 2008.

[28] J. Candia, M. C. Gonzalez, P. Wang, T. Schoenharl, G. Madey, and A.-L. Barabási, "Uncovering individual and collective human dynamics from mobile phone records," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, pp. 1–11, May 2008.

[29] P. Wang, M. Gonzalez, C. A. Hidalgo, and A.-L. Barabási, "Understanding the spreading patterns of mobile phone viruses," *Science*, vol. 324, pp. 1071–1076, Mar 2009.

[30] P. Deville, C. Inard, S. Martin, M. Gilbert, F. R. Stevens, A. E. Gaughan, V. D. Blondel, and A. J. Tatem, "Dynamic population mapping using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 111, no. 45, pp. 15888–15893, 2014.

[31] S. Sarkar, K. Mukherjee, A. Srivastav, and A. Ray, "Understanding phase transition in communication networks to enable robust and resilient control," *American Control Conference (ACC)*, pp. 1549 – 1154, Jun 2009.

[32] S. Sarkar, K. Mukherjee, A. Srivastav, and A. Ray, "Critical Phenomena and Finite-size Scaling in Communication Networks," *American Control Conference (ACC)*, pp. 271 – 276, Jun-Jul 2010.

[33] S. Sarkar, K. Mukherjee, A. Ray, A. Srivastav, and T. A. Wettergren, "Statistical Mechanics-Inspired Modeling of Heterogeneous Packet Transmission in Communication Networks," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 42, pp. 1083–1094, Aug 2012.

[34] T. Tanaka and K. Kitagawa, "Interference Reduction in CDMA Channels: A Statistical-Mechanics Approach," *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops (WiOPT)*, pp. 604 – 609, 2008.

[35] M. Vehkapera, M. A. Girnyk, T. Riihonen, R. Wichman, and L. K. Rasmussen, "On achievable rate regions at large-system limit in full-duplex wireless local access," *International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 7 – 11, 2013.

[36] K. H. Hui, D. Guo, R. A. Berry, and M. Haenggi, "Performance Analysis of MAC Protocols in Wireless Line Networks Using Statistical Mechanics," *Annual Allerton Conference*, pp. 1315 – 1322, Sep-Oct 2009.

[37] K. H. Hui, D. Guo, and R. A. Berry, "Medium Access Control via Nearest-Neighbor Interactions for Regular Wireless Networks," *ISIT*, pp. 1813 – 1817, Jun 2010.

[38] M. Rajala and R. Ritala, "Mutual Information and Multidimensional Scaling as Means to Reconstruct Network Topology," *SICE-ICASE International Joint Conference*, pp. 1398 – 1403, Oct 2006.

[39] S. Jeon and C. Ji, "Nearly Optimal Distributed Configuration Management Using Probabilistic Graphical Models," *MASS Workshop*, pp. 1 – 8, 2005.

[40] A. Assa and M. Jahan, "Adaptive scheduling in Wireless Sensor Networks Based on Potts model," *International eConference on Computer and Knowledge Engineering (ICCKE)*, pp. 248 – 253, Oct 2012.

[41] P. Cardieri, "Modeling Interference in Wireless Ad Hoc Networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 551 – 572, 2010.

[42] H. ElSawy, E. Hossain, and M. Haenggi, "Stochastic Geometry for Modeling, Analysis, and Design of Multi-Tier and Cognitive Cellular Wireless Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 996 – 1019, 2013.

[43] M. Taranetz and M. Rupp, "A Circular Interference Model for Wireless Cellular Networks," *IEEE*, pp. 827 – 832, 2014.

[44] G. L. Stuber, *Principles of Mobile Communication.* Norwell, MA, USA: Kluwer Academic Publishers, 1st ed., 1996.

[45] G. H. Wannier, "Antiferromagnetism. The Triangular Ising Net," *Phys. Rev.*, vol. 79, pp. 357–364, Jul 1950.

[46] D. Sherrington and S. Kirkpatrick, "Solvable model of a spin-glass," *Phys. Rev. Lett.*, vol. 35, pp. 1792–1796, Dec 1975.

[47] M. Bartolozzi, T. Surungan, D. B. Leinweber, and A. G. Williams, "Spin-glass behavior of the antiferromagnetic ising model on a scale-free network," *Phys. Rev. B*, vol. 73, p. 224419, Jun 2006.

[48] C. P. Herrero, "Antiferromagnetic Ising model in scale-free networks," *Eur. Phys. J. B*, vol. 70, no. 3, pp. 435–441, 2009.

[49] P. Contucci, S. Dommers, C. Giardinà, and S. Starr, "Antiferromagnetic Potts Model on the Erdős-Rényi Random Graph," *Communications in Mathematical Physics*, vol. 323, no. 2, pp. 517–554, 2013.

[50] F. Krząkała and L. Zdeborová, "Potts glass on random graphs," *EPL (Europhysics Letters)*, vol. 81, no. 5, p. 57005, 2008.

[51] C. Zhou and R. N. Bhatt, "Understanding and improving the Wang-Landau algorithm," *Phys. Rev. E*, vol. 72, p. 025701, Aug 2005.

[52] P. D. Beale, "Exact Distribution of Energies in the Two-Dimensional Ising Model," *Phys. Rev. Lett.*, vol. 76, pp. 78–81, Jan 1996.

[53] G. H. Wannier, "Antiferromagnetism. The Triangular Ising Net," *Phys. Rev. B*, vol. 7, pp. 5017–5017, Jun 1973.

[54] B. J. Schulz, K. Binder, and M. Müller, "Flat Histogram Method of Wang-Landau and N-Fold Way," *International Journal of Modern Physics C*, vol. 13, pp. 477–494, 2002.

[55] N. M. Luscombe, M. M. Babu, H. Yu, M. Snyder, S. Teichmann, and M. Gerstein, "Genomic analysis of regulatory network dynamics reveals large topological changes," *Nature*, vol. 431, pp. 308–12, 2004.

[56] A. R. Rao, R. Jana, and S. Bandyopadhyay, "A markov chain monte carlo method for generating random (0, 1)-matrices with given marginals," *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 58, no. 2, pp. 225–242, 1996.

[57] R. Kannan, P. Tetali, and S. Vempala, "Simple markov-chain algorithms for generating bipartite graphs and tournaments," *Random Structures & Algorithms*, vol. 14, no. 4, pp. 293–308, 1999.

[58] I. Stanton and A. Pinar, "Constructing and sampling graphs with a prescribed joint degree distribution," *J. Exp. Algorithmics*, vol. 17, pp. 3.5:3.1–3.5:3.25, Sep 2012.

[59] J. Ray, A. Pinar, and C. Seshadhri, "A stopping criterion for Markov chains when generating independent random graphs," *Journal of Complex Networks*, vol. 3, no. 2, pp. 204–220, 2015.

[60] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[61] A. Rényi and G. Szekeres, "On the height of trees," *Journal of the Australian Mathematical Society*, vol. 7, pp. 497–507, Nov 1967.