This is to certify that the
dissertation entitled

Statistical Mechanics of Vertex Cover

presented by

Charles W Fay IV

has been accepted towards fulfillment
of the requirements for the

Doctoral degree in Physics and Astronomy

Major Professor's Signature

12/13/07

Date

STATISTICAL MECHANICS OF VERTEX COVER

By

Charles W Fay IV

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Physics and Astronomy

2007

ABSTRACT

STATISTICAL MECHANICS OF VERTEX COVER

By

Charles W Fay IV

Vertex cover is a member of a class of problems that is known to be computationally hard, and includes the spin-glass and the hard-core lattice gas problems. In computer science this class of problems is known as NP-complete. NP-complete problems display a phase transition between easy and difficult instances. In the worst instances these problems can require a number of operations that scales exponentially in relation to the size of the input. The class of NP-complete problems provides one of the main motivations for quantum computing, and they impinge on nearly every area of physics.

This dissertation presents an investigation of computational techniques to simulate and analyze the behaviour of vertex cover in two and three dimensions. First, the method of reducing a graph by removing small cliques, known as core percolation, is presented. Core percolation by leaf removal is analyzed on the Bethe lattice. The percolative behaviour of the Erdös-Réyni random graph, the triangular lattice, the square lattice and the FCC and simple cubic lattices are examined after leaf removal and after triangle removal. The scaling is found to be in the same universality class as standard percolation. Then a novel approach to finding a near ground state at zero temperature is found for vertex cover, by the analysis of local structure. The behaviour of this algorithm is compared to the replica method and some exact methods. The data generated compares favorably to methods for finding exact covers

ACKNOWLEDGMENTS

I would like to thank my wife Angela, and my family for all of their encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

IMAGES IN THIS DISSERTATION ARE PRESENTED IN COLOR

Chapter 1

Introduction

In recent years we have seen an explosion in the crossover of scientific disciplines. One area that has seen a large amount of interest is the area of computational optimization. As computer simulation becomes increasingly important for a wide variety of disciplines, it becomes necessary to do simulations as quickly and efficiently as possible. A large number of common problems have been found to map to known computationally difficult problems. Among these problems are the ground state of the spin-glass, protein folding, neural networks, time table design, and games like Tetris and Sudoku.

In physics, much work has been done on the ground state of spin-glasses. It is known that finding the ground state of a spin-glass is computationally hard. In the second chapter of my thesis I will review the basics of the spin-glass problem, outlining the replica method, replica symmetry breaking and the rough free-energy landscape this implies. I will outline similar problems known to physicists, that can be mapped to a spin-glass, such as the hard-core lattice gas, protein folding and neural networks. These systems can be modeled with statistical mechanics and display glassy behaviour. These problems also map to the computationally hard class of problems known in Computer Science as nondeterministic polynomial complete (NP-complete).

In the third chapter, I will review the concept of computational complexity in NP-complete problems, outlining some basic NP-complete problems. I will map the hard-core lattice gas to the NP-complete problems of vertex cover, and maximum independent set. NP-complete problems display a phase transition, between regimes where is it commonly easy to find a solution and regimes where it is difficult to find a solution. As physicists we can apply what we know about phase transitions to study and classify the transition region. I will review the replica symmetric solution to the hard-core lattice gas to find a solution to vertex cover, and compare this to our analysis of the solution on the Bethe lattice [37].

Chapter 4 will examine algorithms I used in generating my numerical results. First, I will present some basic algorithms used to find exact solutions on vertex cover. I will also present the leaf-removal and triangle procedures which are a subset of the minimum vertex cover algorithm by Tarjan and Trojanowski [87]. Leaf removal was previously utilized by Karp [46] in a study of matching, and Bauer and Golinelli [4] for calculating the maximum independent set on a random graph. Leaf removal has also been used by Correale et. al. to study core percolation in Boolean networks [15]. I will analyze leaf-removal on the Bethe lattice [38]. Since exact solutions of NP-complete problems can handle no more than a few hundred sites, we often have to resort to heuristics to find near optimal solutions. A good heuristic will give some insight into the actual solutions and the nature of the ground state. Two common heuristics for vertex cover are greedy and random selection, I will outline these two before presenting and analyzing a new heuristic for vertex cover, that we have termed LoPR, which relies upon local cluster geometries [40] [37].

Chapter 5 is devoted to analyzing the percolative results of leaf removal on the random graph, and the triangular, square, simple cubic and FCC lattices. Here we show that leaf removal is in the same universality class as standard percolation. The results for the triangular lattice have been published [40], while a paper detailing the results on the other regular lattices is in preparation [37]. Chapter 5 also contains the results of triangle removal on the FCC and triangular lattices [38]. Triangle removal is also in the same universality class as standard percolation.

The sixth chapter presents the approximate solution of maximum independent set problems using our LoPR procedures, on the triangle, square, FCC, simple cubic lattices and the random graph. LoPR shows good results on finding the cardinality of the maximum independent set, but fails to fully represent the frozen fraction. Results for LoPR on the triangle lattice have been published [40], a paper detailing the results on the other lattices is in preparation [39].

Chapter 2

Physics of Hard Computational Problems

2.1 Spin glasses

A spin glass is a collection of spins that in its low-temperature state has a backbone of frozen and disordered sites [22], the sites not residing on the backbone are allowed to fluctuate. In order to construct such a phase two ingredients are necessary, disorder and frustration (see section 2.1.1).

In the nineteen-fifties and early nineteen-sixties, measurements were made on alloys such as AuFe and CuMn that exhibited unusual properties in the AC magnetic susceptibilities. The experimental properties as listed by Chowdhury [13] are: (i) the ac susceptibility $\chi_{ac}(T)$ exhibits a cusp at temperature T_g at low-frequency and in low magnetic fields see Fig. 2.1, (ii) no sharp anomaly appears in the specific heat (iii) below T_g the magnetic response is dependent upon previous magnetizations, (iv) below T_g the remnant magnetic field decays very slowly over time, (v) below T_g a hysteresis curve, laterally shifted from the origin appears in the magnetization, (vi) below T_g no magnetic Bragg scattering is observed, demonstrating an absence of periodic long range order, (vii) the susceptibility begins to deviate from the Curie law at temperature $T >> T_g$ [13].

Physical systems that have been associated with spin glasses include, dilute magnetic alloys (such as AuFe and CuMn mentioned earlier) and amorphous magnetic systems (such as FeMn, CoMn and $CrSnTe_4$). Chowdhury, [13] contains a description of many physical SG systems.

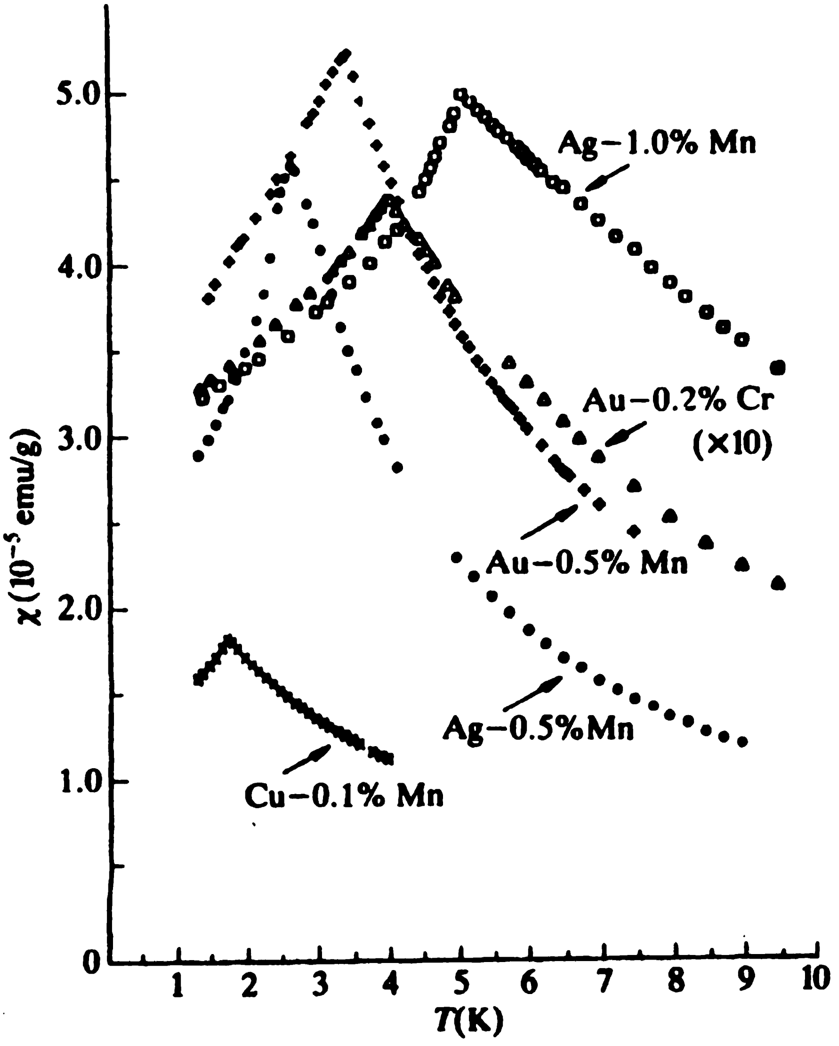


Figure 2.1: Cusp in the AC susceptibility of dilute magnetic alloys [22]

Progress toward a mean field solution for spin glasses began in 1975, with the

Edwards-Anderson model (EA), [19];

$$H = -\frac{1}{2} \sum_{\langle ij \rangle} J_{ij} \vec{s_i} \cdot \vec{s_j} \tag{2.1}$$

where J_{ij} is the magnetic coupling between sites i and j, if $J_{ij} > 0$ the coupling is ferromagnetic and if $J_{ij} < 0$ the coupling is anti-ferromagnetic. The spins are classical dipoles free to point in any direction. Edwards and Anderson introduced the spin-glass order parameter defined by,

$$q_{EA}(t) = [s_i(t) \cdot s_i(0)]_{ave}, \tag{2.2}$$

where the square brackets ($[]_{ave}$), denote the time average. The EA model's order-parameter is a measure of the long-time auto-correlation [13]. In the long time limit as $t \longrightarrow \infty$, we expect Eq. (2.2) to be equal to the static EA order parameter,

$$q_{EA} = [[s_i]^2]_{ave}, \tag{2.3}$$

$$\tag{2.4}$$

the brackets ([]) denote the thermal average. In the spin glass phase, because of a lack of long range ferromagnetic order, the magnetization is

$$m = [s_i] = 0, \tag{2.5}$$

while the spin-glass order-parameter is non-zero. The random spin interactions are given by a Gaussian distribution of,

$$P(J_{ij}) \propto \exp(-J_{ij}^2 / 2J^2 \rho_0), \tag{2.6}$$

where $J^2 = \sum_{ij} J_{ij}^2$ and ρ_0 is the density of bond occupation [19].

Sherrington and Kirkpatrick [83] solved an Edward-Anderson type system with Ising-spins, having N infinite range interactions. Moreover, in terms of the free energy, it has been shown that as the dimension, D, of the EA model increases [78],

$$\lim_{D \to \infty} EA(D) = SK. \tag{2.7}$$

The SK model is important because it was the first spin-glass system solved, giving mean-field values for the spin-glass. The disorder and frustration is supplied by the parameter J_{ij} which can vary randomly between positive and negative values.

The Hamiltonian of the SK model is similar to the EA model, containing Ising spins in place of the vector spins,

$$H = -\frac{1}{2} \sum_{\langle ij \rangle} J_{ij} s_i s_j \tag{2.8}$$

with a Gaussian distribution of J_{ij} given as,

$$P(J_{ij}) = \frac{1}{J\sqrt{2\pi}} \exp\left(-\frac{(J_{ij} - J_0)^2}{2J^2}\right) \tag{2.9}$$

with J_0 and J scaled, to meet the requirement that the free energy is extensive, by,

$$J_0 = \tilde{J}_0 / N \quad \text{and} \quad J = \tilde{J} / N^{1/2} \tag{2.10}$$

where $\tilde{J}_0$ and $\tilde{J}$ are intensive and N is the number of spins in the system. The relative magnitudes of $\tilde{J}_0$ and $\tilde{J}$ determine if spin-glass ordering or ferromagnetism occur at low temperatures [50]. The partition function of the SK model is,

$$Z = \sum_{s_i} \exp\left(\frac{\beta}{2} \sum_{\langle ij \rangle} J_{ij} s_i s_j\right), \tag{2.11}$$

from which the physical properties may be calculated for example the free energy,

$$F = -k_B T \ln Z. \tag{2.12}$$

In systems that are highly disordered like spin-glasses, it is often difficult to calculate quantities like the free energy that are dependent upon the averaged logarithm of the partition function. The replica method allows us to determine the logarithm of the partition function by the expansion of a replicated partition function,

$$\langle \ln Z \rangle = \Big\langle \lim_{n \to 0} \frac{Z^n - 1}{n} \Big\rangle, \tag{2.13}$$

the brackets ($\langle\rangle$) signify an average over the disorder. Eq. (2.13) is derived from the expansion [53],

$$Z^n = 1 + n \ln Z + O(n^2). \tag{2.14}$$

For a more detailed derivation of the replica method on the SK model see Appendix B. In the replica method, the partition function is copied n times. To average over the quenched couplings, one takes the thermodynamic limit $N \longrightarrow \infty$, and then uses the mathematically dubious process of doing analytic continuation on integers, to take the limit as $n \longrightarrow 0$. The order parameter is now a matrix with elements,

$$q_{\alpha\beta} = \langle s_i^{\alpha} \cdot s_i^{\beta} \rangle, \tag{2.15}$$

α and β are replica indices. If all overlaps between replicas are assumed to collapse into a steady state value,

$$q = q_{\alpha\beta}., \text{ if } \alpha \neq \beta. \tag{2.16}$$

This is known as the replica symmetric solution. After a bit of work we arrive at the order parameter and the magnetization from the SK model (see Appendix B),

$$
\begin{aligned}
q &= 1-\frac{1}{\sqrt{2\pi}}\int dze^{-z^2/2}\mathrm{sech}^2\left(\frac{\tilde{J}}{kT}q^{1/2}z+\frac{\tilde{J}_0}{kT}m\right) && (2.17)\\
m &= \frac{1}{\sqrt{2\pi}}\int dze^{-z^2/2}\tanh\left(\frac{\tilde{J}}{kT}q^{1/2}z+\frac{\tilde{J}_0}{kT}m\right). && (2.18)
\end{aligned}
$$

$m = 0$ and $q \neq 0$ indicates the spin-glass phase.

Almeida and Thouless showed that the SK model was correct above the critical temperature T_g. At low temperatures, the entropy calculated from the symmetric SK method becomes negative and the solution becomes unstable. The entropy calculated from the symmetric SK model is [16] [50],

$$
\begin{aligned}
\frac{S}{N} &= -\left[\frac{\tilde{J}_0 m^2}{2T}+(\frac{\tilde{J}^2}{4kT^2})(1-q)(1+3q)-\frac{Hm}{T}\right]\\
&\quad +k(2\pi)^{-1/2}\int dz\exp(-\frac{1}{2})\ln(2\cosh Z). && (2.19)
\end{aligned}
$$

This entropy is negative and nonphysical when $T \longrightarrow 0$, $H = 0$, $m = 0$. In finite fields, $H > 0$, the onset of replica symmetry breaking (RSB) defines the Almeida-Thouless line [16].

In 1977, Thouless, Anderson and Palmer (TAP) generated mean-field equations for the SK model [88]. The TAP equations describe a "many valley" picture of the rough free-energy surface defining the solution space. The lowest valleys correspond to stable thermodynamic states, the vast majority of valleys are meta-stable states [9]. The number of solutions (stable and meta-stable) is exponential in N ($\exp(\alpha N)$). We can imagine the surface as a series of valleys separated by barriers that become infinitely tall in the limit N becomes large. In these valleys, we may have local minima that are separated by finite barriers, any valleys not corresponding to global minima would be meta-stable states [22].

Replica symmetry breaking is evident in the order-parameter, in the RS solution

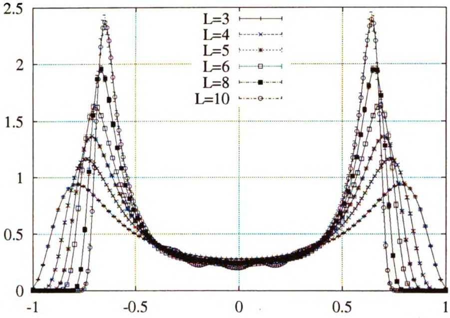


Figure 2.2: The order parameter distribution, $P(q) = \langle P_J(q) \rangle$ for a spin-glass [78].

there is one spin-glass order parameter, while in the RSB solution there exists an infinite number of order parameters [74] [75] [76] forming the matrix in Eq. 2.15. The Parisi method of symmetry breaking proceeds by starting with the replica symmetric matrix and dividing it into $(n \times m_1) \times (n \times m_1)$ blocks, see Fig. 2.3. The off-diagonal blocks are left unmodified with a value of q_0 and the diagonal blocks are assigned a value of q_1. The diagonal blocks are further subdivided iteratively [58]. Breaking the symmetry of the order-parameter into an infinite number of matrix elements makes the determination of a solution significantly more difficult [62]. This matrix is associated with a function $q(x)$ describing the order parameter [75]. This function exists on the interval $[0-1]$ and is defined by,

$$q(x) \quad = \quad q_i \text{ if } m_i < x < m_{i+1} \tag{2.20}$$

The sum is done over all $n(n-1)/2$ pairs of replica indices, the normalization insures $q^{(0)} = 1$ [74] [62]. Parisi lists some interesting qualities of the function

$q(x)$; (i) $q(x)$ is piecewise constant under K-step symmetry breaking taking on K+1 values, (ii) $x(q)$ the inverse function is discontinuous, and gives the order parameter distribution,

$$P(q) = \frac{dx}{dq}, \tag{2.21}$$

(iii) $q(x)$ is continuous in the $K \to \infty$ limit, leading to continuous replica symmetry breaking [78]. The function $P(q)$ has the form of two delta functions at $\pm q_{EA}$ with a flat region in between in zero magnetic field [78]. Figure 2.2 shows the order parameter distribution for a spin glass. The quantity $P_J(q)$ is the "probability distribution of the overlap among two equilibrium states" [78]. When we average $P_J(q)$ over the disorder, we arrive at the mean field approximation of the order parameter distribution,

$$P(q) = \langle P_J(q) \rangle. \tag{2.22}$$

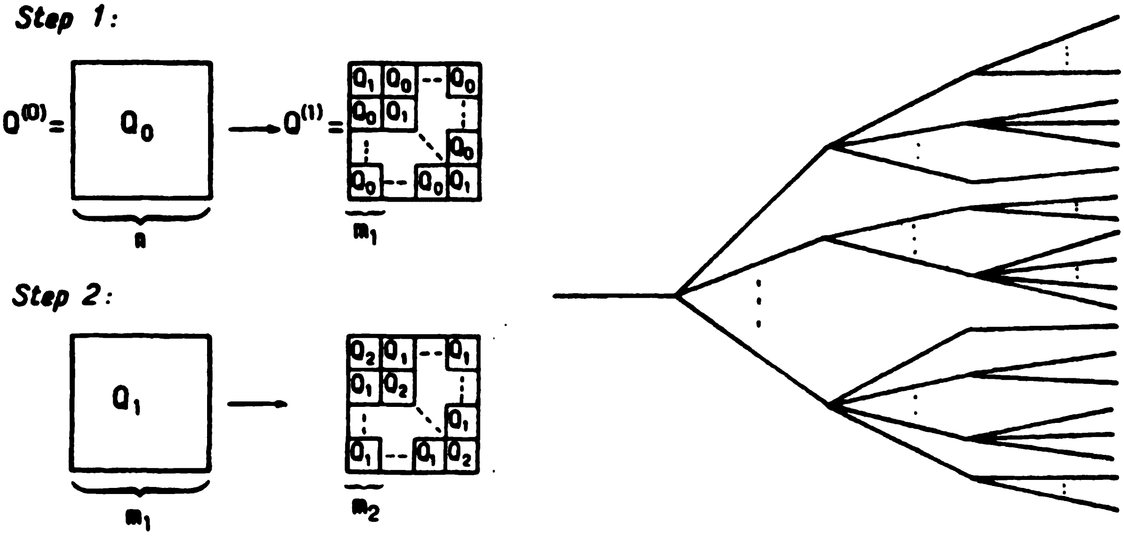


Figure 2.3: Diagram of Parisi's replica symmetry breaking procedure and the tree-like structure of its solutions. From Mezard et. al. [57] [58].

The Parisi method of symmetry breaking the solution implies a multiplicity of states, organized in a ultrametric structure [57]. The structure of the space, implies that the order parameter is not self-averaging, but depends on the realization of the couplings [60]. Fig. 2.3(b) shows the tree like structure of an ultrametric space, stable states are the lowest points in the trees. The distance between two states is the number of steps one needs to make on the ultrametric tree to get to another state. The representation becomes a genealogical tree such as in Fig. 2.3. The states of the systems are the extremities of the trees, while the overlap between states "depends only on their closest ancestor" [60]. These states are believed to correspond to the global minimums from the TAP equations [22]. For a stable state α there is an ultrametric distance $D_\alpha(i,j)$ between the N sites. There is a minimum distance d such that for two states with an overlap $q_{\alpha\beta}$, the distances D_α and D_β are equal and $D_\alpha(i,j) \geq d$ [60]. This means for any three states forming a triangle, the triangle will be either equilateral or isosceles. For a ultrametric space we have the inequality,

$$d_{\alpha\beta} \leq \max(d_{\alpha\gamma}, d_{\beta\gamma}). \tag{2.23}$$

"In an ultrametric space after any number of fixed length steps one discovers oneself at the same initial distance from the starting point" [62]. All points then lie in one cluster. The equilibrium states are divided by energy barriers that become infinitely high in the thermodynamic limit. These states are believed to correspond to the global minimums from the TAP equations [22].

Questions have been raised about the interpretation of the existence of an infinite number of ground states on real finite size spin-glasses by Newman and Stein, where they have shown that there are a small number of ground states with the metastable states sitting on the boundaries of the real solutions [68] [67].

Glass phases have been shown to exist in p-spin glasses that are not frustrated [25], and have also been demonstrated in spin systems without disorder [7].

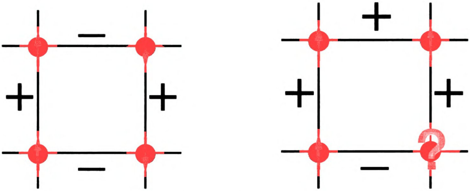


Figure 2.4: Frustration as shown on two plaquettes, the first plaquette is not frustrated. The second is unable to fully satisfy all bonds.

2.1.1 Frustration

Frustration is the term that is applied to the competing interactions that form an important ingredient in any spin-glass [89]. To examine frustration, lets consider for a moment a 2-d square lattice with an Edwards-Anderson Ising model, Fig. 2.4. For the purposes of illustration, the interactions here are $\pm J$, a "$+J$" interaction is ferromagnetic, causing the spins to want to be aligned, while a "$-J$" interaction is anti-ferromagnetic causing the spins to be in opposite directions. If we place two ferromagnetic bonds randomly and two anti-ferromagnetic bonds randomly on the square plaquette, Fig. 2.4(a), we can see that it is easy to satisfy the plaquette regardless of the placement of the bonds illustrating a case of disorder but no frustration. However, if we place an odd number of "$-J$" bonds on the plaquette, then there will be one bond that cannot be satisfied. If all the bonds have the same value for $\mid J \mid$ then there is a degeneracy in the ground state of 4. If different values of $\mid J_{ij} \mid$ are placed into the frustrated plaquette, the degeneracy is broken, and there exists the possibility of metastable configurations [22]. If we place the two smallest values of $\mid J_{ij} \mid$ on opposite sides of the plaquette, there is a metastable state if the second weakest bond is broken, separated by a barrier from the true ground state (weakest bond) [22]. Glassy behaviour is expected in diluted geometrically frustrated systems, such as the hard-core lattice gas [92], and the diluted anti-ferromagnet in a field [79].

2.2 Hard core lattice gas

The hard-core lattice gas, like the Ising model, is a fundamental model for physicists in condensed matter. Its Hamiltonian,

$$H = \sum_{\{ij\}} J_{ij} n_i n_j - \mu \sum n_i, \tag{2.24}$$

describes a lattice with nearest neighbor repulsion ($J > 0$), and chemical potential μ. n_i represents the occupancy variable, $n_i = 1$ indicates an atom at site i, and a value of $n_i = 0$ means the site is vacant. By adjusting the chemical potential we can control the number of atoms. At low temperatures for $\mu > 0$, $J \rightarrow \infty$ and $\mu/J \rightarrow 0$ the lowest energy state corresponds to the maximum density of atoms on the lattice, with the constraint that no two atoms occupy nearest neighbor sites. The maximum density is equivalent to the NP-complete problem of maximum independent set and is related to the vertex cover problem. I will define NP-completeness and discuss it in more detail in Chapter 3.

The Hamiltonian for the hard-core lattice gas can be transformed to the Ising model by the substitution of,

$$n_i = \frac{s_i - 1}{2}, \tag{2.25}$$

which yields the diluted anti-ferromagnetic in a field,

$$H = \sum_{ij} J_{ij} s_i s_j - \sum h_i n_i. \tag{2.26}$$

The lattice gas often differs from the Ising model by placing a constraint upon the number of atoms in the lattice. The equivalence breaks down for random systems when the chemical potential and the magnetic field involve the quenched coupling [82]. However by using the replica method on the hard-core lattice gas a glass state can be shown [82] [81] [69](see appendix B).

Disorder in this model may be introduced through the turning off or on of the interactions J_{ij}. Frustration is introduced by the geometry of the lattice, providing the elements of possible glassy behaviour. The hard-core lattice gas has been studied in relation to glasses [82] [92] [77] and vertex cover [94] [34] [2] [93].

Once we have a description of the system we need a set of observables; *physically, we analyze the following quantities;*

- *Density, ρ*
- *Degeneracy, entropy*
- *Frozen fraction*

We will be doing our work with the Helmholtz free energy, F,

$$F = E - TS = -kT \ln Z. \tag{2.27}$$

The thermal average of the density is defined as,

$$\rho = [\frac{1}{N}(\sum_i n_i)] = \frac{1}{Z}\sum_{n_i}(\frac{1}{N}\sum_i n_i \exp(-\beta H)). \tag{2.28}$$

Which we can see is,

$$\rho = \frac{1}{\beta N}\frac{\partial}{\partial \mu}\frac{\exp(-\beta H)}{Z} = \frac{1}{\beta N}\frac{\partial \ln Z}{\partial \mu}. \tag{2.29}$$

At $T = 0$ the density corresponds to the normalized cardinality of the maximum independent set, and $1 - \rho$ gives the density of the vertex cover. The disorder average of the free energy, F , $\langle F\rangle$, is equal to

$$\langle F\rangle = -k_B T\langle \ln Z\rangle, \tag{2.30}$$

and we may use the replica method to solve for $\langle \ln Z\rangle$, in systems where a standard formalism is very difficult.

2.2.1 A physical example

A physical example of a lattice gas arises in the modeling of a nano-structure made up of 2 types of atoms, A and B. Because of the relative sizes of A, and B, there are favorable combinations, so we define (observe) occupancy rules.

- A - B = Good
- B - B = Good
- A - A = Bad

This would be the case if A were too large to occupy adjacent sites without distorting the lattice. This can then be modeled by the hard-core lattice gas where atoms of type A are the particles and B are vacancies.

This is the case with Xenon on Graphite. Xenon takes the role of the atom of type A, and the vacancies are analogous to the atoms of type B. Graphite forms a hexagonal lattice, the minimums of the potential energy fall in the centers of the hexagonal lattice (forming a triangular lattice). The noble gas prefers to sit in the energy wells, the difficulty arises because the size of the deposited atoms is larger than the cell spacing, making it impossible to occupy nearest neighbor sites. We know that the maximum independent set corresponds to the densest packing in the first layer.

In the Xenon and graphite system experimentally several things happen. At low temperature and pressure Xenon has a incommensurate configuration. As pressure is increased the system under goes a transition into a commensurate phase [41] [43] [65] [70] [42] [36] [72]. This results from the $\sqrt{3} \times \sqrt{3}$ spacing 4.26 Å, being slightly smaller than the spacing for the lowest Xe-Xe energy, 4.40 Å [41].

Joos et al. calculated the maximum allowable compression, by minimizing the enthalpy per atom of a double layer. The potential for a Xe atom above a graphite layer can be given as,

$$V(\vec{r}, z) = V_0(z) \sum_{\vec{g}} V_g(z) \exp(i\vec{g} \cdot \vec{r}). \tag{2.31}$$

The $\vec{g}$ are the reciprocal lattice vectors for the two dimensional graphite surface. V_g decreases rapidly with increasing height.

Periodic lattice structures were relaxed to strain-free configurations. The surface coverage of n for a compressed configuration of hexagonal symmetry on the $\sqrt{3} \times \sqrt{3}$ plane can be given by,

$$\frac{n_{HC}}{n_0} = \frac{9l^2 - 9l + 3}{9l^2 - 15l + 7}. \tag{2.32}$$

where l is the number of atoms per side of the hexagon. For expanded configuration of hexagonal symmetry,

$$\frac{n_{HE}}{n_0} = \frac{9l^2 - 9l + 3}{9l^2 - 3l + 1}. \tag{2.33}$$

For striped compressed and expanded.

$$\frac{n_{SC}}{n_0} = \frac{3l}{3l - 2}, \tag{2.34}$$

$$\frac{n_{SE}}{n_0} = \frac{3l}{3l + 2}. \tag{2.35}$$

The coverage is obviously just an expression of the density.

Joos et al. then calculated the energies per atom of striped and hexagonal configurations. These show curves that give an energy minimum for a particular fractional coverage, with the minimum energy corresponding to the maximum coverage. This idea can also be extended to the formation of a second layer providing a threshold coverage of the first layer before the second layer begins to form.

2.3 Diluted anti-ferromagnet and the hard-core lattice gas

A direct mapping exists between the hard core lattice gas and diluted antiferromagnets in a field (DAFF's). The Hamiltonian of the DAFF is,

$$H = J \sum_{\langle ij \rangle} \epsilon_i \epsilon_j s_j s_j - B \sum_i \epsilon_i s_i, \tag{2.36}$$

where, B is the applied magnetic field, s_i are Ising spins, and ϵ_i represents the dilution. Sites are present, $\epsilon_i = 1$, with probability p and absent, $\epsilon_i = 0$, with probability $1 - p$.

DAFF's were originally studied as the experimental mapping of the random field Ising model, a fundamental model of disordered materials. This mapping applies for DAFF's on bipartite lattices as occurs for the system $Fe_x Zn_{1-x} F_2$ [71] [48] [30] [3], and hence the hard core lattice gas on diluted square and cubic lattices map to the random field ising model.

In the DAFF a domain state (DS) that is similar to a spin glass has been shown, this state is characterized by "domains, metastability and slow dynamics" [71]. In the H, T plane, the DS region occurs between the long range ordered anti-ferromagenetic phase and the paramagnetic phase [71], above a critical field.

Glaser et. al. examined the phase diagram in the B, x plane [30], showing the onset of glassy behavior at a critical concentration, and field. In the B, x plane the low concentration, low field state is the paramagentic phase and followed by the onset of the DS state and the long range anti-ferromagnetic state [30].

More recently the issue of geometrical frustration has gained attention [79]. The hard core lattice gas on triangular, face-centered cubic and random graphs is geometrically frustrated due to odd loops in the graph. Frustration due to odd loops is common to the hard core lattice gas and Ising antiferromagnets.

2.4 Spin glass applications

2.4.1 Neural networks

Spin glass theory has also found applications in biological systems such as neural networks, and protein folding. The Hopfield method of modeling neural networks can be described by the Hamiltonian [62] [35],

$$H = \sum_{i<j} J_{ij} s_i s_j. \tag{2.37}$$

Under a sequential model the neurons are updated by the equation,

$$s_i(t + \delta t) = sign(\sum_i J_{ij} s_i + h_i), \tag{2.38}$$

where we define,

$$\begin{aligned} s_i &= \text{state of the } i\text{th neuron} \\ J_{ij} &= \text{synapse from } j\text{th neuron to the } i\text{th neuron, assumed to be symmetric} \\ h_i &= \text{threshold of } i\text{th neuron} \end{aligned} \tag{2.39}$$

t and δt define the intervals between the learning events [62]. $s_i = 1$ if the neuron is firing and $s_i = -1$ if it is not firing. Learning can be introduced by the Hebb's rule [1] [62]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \zeta_i^\mu \zeta_j^\mu, i \neq j. \tag{2.40}$$

The sets of $\{\zeta_i^\mu\}$ are determined by the learning process, assuming a value of ± 1 with equal probability [1]. Amit, Gutfreund and Sompolinsky solved the symmetric case, showing the onset of metastable states of the associative memories in the low temperature regime [1]. Another learning model has been presented by Toulouse et. al. [90]. Parga and Virasoro examined the ultrametric structure of memories, generated by a Markovian process of categorizing words as they are stored [73]. Mézard applied the replica method to several different types of learning patterns modeling short term memory [55]. Gardner calculated the maximum storage capacity of such networks [27] [28]. A more mathematical treatment of neural networks is given in the book "Mathematical Aspects of Spin Glasses and Neural Networks" [8].

2.4.2 Protein folding

Protein folding has been modeled as a spin glass, by Rao and Bhattacarjee, using the Hamiltonian [80],

$$H = \sum_{i,j} \frac{J_{ij}}{(j-i)^{\sigma}} s_i \cdot s_j \tag{2.41}$$

here σ is a measure of the correlation. While Bryngelson and Wolynes used the form [11],

$$H = -\sum_{i} \epsilon_i(\alpha_i) - \sum_{i} J_{i,i+1}(\alpha_i, \alpha_{i+1}) - \sum_{ij} K_{ij}(\alpha_i, \alpha_{i+1}, r_i, r_j), \tag{2.42}$$

ϵ_i is a measure of the energy associated with an amino acid state, α_i, $J_{i,i+1}$ is related to the interactions along the chain taken as nearest neighbor interactions, K_{ij} are long-range interactions, caused when acids far apart on the chain are brought together by bends in the chain, r_i is a position vector to the ith element [80]. Frankel showed that the protein folding problem, like the spin glass, is a member to the computer science classification of problems known as NP-complete [24]. Spin glass theory has also provided a method of optimizing code for protein folding [31]

2.4.3 Combinatorial optimization

The spin glass problem has been shown to be NP-complete, this allows one to apply statistical physics to other NP-complete problems. In this thesis we will concentrate on the problems of vertex cover and maximum independent set. Computational complexity will be studied further in chapter 3. A large amount of work has been done in the area of combinatorial optimization, in relation to the spin glass [60] [26] [62].

Chapter 3

Computational Complexity

The difficulty or complexity of a problem in computer science is measured in terms of the time it takes to solve the problem. Time, τ, is a measure of the number of operations needed to find a solution. A theoretical computer such as a Turing machine is used to define a standard number of operations an algorithm uses. A problem is considered hard if it takes an exponential number of steps to solve the problem, in this case τ scales as $\tau \sim \exp(\alpha N)$. If a problem has an input of size N and it takes $O(N^{\alpha})$ operations to solve it, in the worse case scenario, that problem is known as polynomial. In practice if α is very large, it can still be very difficult to find a solution. But, a polynomial time algorithm is, in general, better than an exponential time algorithm. If a problem can be shown to be solvable using a polynomial algorithm the problem is said to be tractable. For many problems it is difficult to determine tractability or intractability.

It is possible to make some distinctions about the time needed to solve some problems in the worst case. Already we have defined those problems which are soluble in polynomial time as tractable, these problems are said to be in the class P. The problems that are said to be soluble by a "nondeterministic" algorithm are in the class NP. Primarily the concept of a nondeterministic algorithm is a "definitional device for capturing the notion of polynomial time verifiability" [29]. Polynomial time verifiability means that if a possible solution is available, it can be determined in polynomial time if it is a viable solution. A nondeterministic algorithm may be defined, informally, as an algorithm which finds a "solution" to a given instance, by guessing and checking [29]. It is still an open question if NP is also a member of P ($NP = P$), however, the widely held belief is that they are not equivalent.

There are two ways to pose computational problems; as a decision problem or as an optimization problem. The decision problem is considered easier, and is usually used to prove NP-completeness. A decision problem poses the question such that there are two possible solutions, yes or no. An optimization problem is a problem where the object is find the best possible solution, for example, minimizing the energy of the ground state of a spin-glass. We will be concentrating on solutions of optimization problems.

3.1 NP-completeness

The first classification of NP-complete (NPC) problems was by Stephen Cook in 1971 [14]. Later, Karp added to that number [44] [45]. Currently there are hundreds of problems that have been classified as NP-Complete [29]. There are two major characteristics of NPC problems;

1. all NPC problems are soluble in polynomial time by a "nondeterministic" algorithm,
2. any problem in the class NPC may be transformed to another problem in NPC by the application of a polynomial time reduction.

The first characteristic simply means that a possible solution is verifiable in polynomial time, while the last characteristic means that if any NPC problem were solvable in polynomial time all NPC problems would be, or conversely if it were proven that any problem in NPC were proven to be intractable all problems in NPC would be intractable. Proof of NPC generally consists of showing a polynomial transformation into a known NPC problem.

There are six basic NP-Complete problems; the traveling salesman, 3-SAT, 3-dimensional matching, minimum vertex cover, Hamiltonian circuit and number partitioning [29].

In all NPC problems there are instances that are easy to solve, and instances that are difficult to solve. It can be shown that there is a phase transition that exists between easy and hard instances [94] [33]. It is particularly hard to find a solution near the critical value in NP-complete problems, since near the critical point complexity problems display critical slowing down. One way to determine the critical point for an NPC problem is to measure the time required to find a solution in respect to a variational parameter [34], such as the connectivity in a graph theory problem such as vertex cover.

3.1.1 P to NP transition

When finding a vertex cover, or a ground state, in the hard-core lattice gas, there are 2 possible states for every vertex, so that if there are N vertexes that means there are 2^N configurations, that may be a possible solution. If each configuration has to be tested, it is not hard to see that even with a small number of nodes, (ie N=100 then $2^N = 1.27 \times 10^{30}$) it becomes prohibitively difficult to find a solution in any reasonable time. Fortunately, the typical case is often easier than the worst case scenario, however, no exact algorithm exists to solve NP-complete problems in less than exponential time in the worse case scenario. We will discuss specific algorithms more in chapter 4.

When solving NPC problems on graphs, such as vertex cover, It has been observed that there is a phase transition in computational complexity, as the connectivity of the graph is increased. This can be understood in relation to percolation (see Appendix A) [64]. We know the probability, $P(s)$, of a node being on a cluster of size s in a random graph below the critical point is,

$$P(s) \sim \exp(-\alpha s), \tag{3.1}$$

where α depends on the concentration of present sites. With N clusters,

$$N \exp(-\alpha s_{max}) \sim 1, \tag{3.2}$$

so that,

$$s_{max} \sim \frac{\ln N}{\alpha}. \tag{3.3}$$

The largest cluster, s_{max} is proportional to $\ln N$. The time,τ, to find a solution for the largest cluster is,

$$\tau \sim s_{max} \sim \exp\left(\frac{\ln N}{\alpha}\right) \sim N^{1/\alpha}. \tag{3.4}$$

With N clusters the time to find a solution for the graph is,

$$\tau \sim N^{1/\alpha+1}. \tag{3.5}$$

Thus the time to find a solution below the critical point is polynomial. Above the critical point, the largest cluster is on the order of $O(N)$ and not $O(\ln N)$ as it is below the threshold. This means the time to find a solution of the largest cluster at worst is,

$$\tau \sim \exp\left(\frac{N}{\alpha}\right), \tag{3.6}$$

which is clearly exponential in time. This case illustrates a phase transition that is observed in the computational complexity as average coordination number of the vertexes, c, is increased.

Some relevant questions, we then ask from a computer science perspective are;

- *How do we speed up our calculations (algorithmically)?*
- *Are there easy instances or sub-graphs that can be dealt with in a way that speeds up the calculations?*
- *In NPC problems not all instances are hard, what are the characteristics of easy instances?*

While as a physicist, one might ask, do the physical examples of these problems give us insight in how to solve them? What insight can the study of phase transitions in physics provide to computational problems.

Mézard et al. in their book "Spin Glass Theory and Beyond", provide a basic mapping of a computation optimization problem to statistical physics [62]. The partition function is,

$$Z = \sum_{\text{configurations}} \exp(-\beta \times \text{cost(configuration)}) \tag{3.7}$$

The mapping is then [62],

OPTIMIZATION	STATISTICAL PHYSICS
instance	sample
cost function	energy
optimal configuration	ground state
minimal cost	ground state energy

3.1.2 The traveling salesman

One of the first problems to be identified as computationally complex is commonly known as the Traveling Salesman Problem (TSP). It may be posed as: *given a fixed number of cities and various roads connecting them, with differing speed limits, what is the fastest round trip where the traveling salesman visits every city once?* The decision problem could be posed as: *given a fixed number of cities and various roads connecting them, with differing speed limits, does there exist a round trip with a time less than some maximum,* $t_{trip} \leq t_{max}$*?* A partition function for the traveling salesman problem can be written as,

$$Z_{TSP} = \sum_{\text{tours}} \exp\Big(- \beta N^{1/r+1} L_{\text{tour}} \Big). \tag{3.8}$$

Mezard and Parisi did a statistical mechanical analysis based upon this partition function, and derived its replica symmetric solution [56]. To make the entropy extensive, at low temperatures, we reduce the relevant number of tours by connecting a node to two other nodes with probability,

$$p(d) \overset{d \to 0}{\sim} \frac{d^r}{r!}, \tag{3.9}$$

where d is the typical distance between neighbors,

$$d \sim N^{-1/(r+1)}. \tag{3.10}$$

This makes the free energy,

$$F = -\frac{1}{\beta} \frac{1}{N^{1/(r+1)}} \log Z. \tag{3.11}$$

We can write a generalized partition function, if at each site we introduce an m-component spin S_i of fixed length $S_i^2 = m$,

$$Z = \int \Pi_i d\mu(s_i) \exp\left(\gamma \sum_{i<j} \exp\left(-\frac{1}{\beta N^{1/(r+1)}}\right) d_{ij} S_i \cdot S_j\right), \tag{3.12}$$

where,

$$d\mu \equiv C_m d^m s \delta(S^2 - m). \tag{3.13}$$

The partition function in Eq. (3.12) is the same as Z_{TSP} in the limit,

$$Z_{TSP} = \lim_{m \to 0} \frac{1}{m} \lim_{\gamma \to \infty} \frac{Z}{\gamma^N}. \tag{3.14}$$

3.1.3 3-SAT and K-SAT

Satisfiability (SAT) is a logical problem that asks if a set of N Boolean variables organized into M clauses, can be satisfied simultaneously [59]. A clause is a set of K literals that may be a variable ($l_p^i = x_j$) or a negated variable ($l_p^i = \bar{x}_j$). Variables form a clause by being in disjunction (logical OR); $C_p = l_p^1 \vee l_p^2 \vee l_p^3$ [34]. The Boolean formula can be expressed in conjunctive normal form as,

$$F = (x_1 \vee \bar{x_3} \vee \bar{x_4}) \wedge (\bar{x_1} \vee x_2 \vee \bar{x_4}), \tag{3.15}$$

$\wedge$ is a logical AND. The example given in Eq. (3.15) is 3-SAT with $M = 2$ clauses and $N = 4$ Boolean variables. 3-SAT was shown to be NP-complete by Cook in 1971 [14], with a transition near $\alpha \simeq 4.2$ [49], where $\alpha = M/N$. When $K = 1$, or 2 the problem can be solved with a polynomial time algorithm. The general case, K-SAT, where the number of variables in each clause is allowed to vary, is NP-complete if $K \geq 3$ [29]. Monasson et al. [64] studied the onset of the transition from P to NP using a mixture of 2 variable and 3 variable clauses. They showed for this $(2+p)$-SAT model the transition between polynomial/exponential occurs near the critical value of $p > p_0 \sim 0.4$, where the formula contains M clauses, $(1-p)M$ two variable clauses and pM three variable clauses [64] [51].

K-SAT is important because it is very general and many other problems have been transformed into K-SAT to prove NP-completeness. A Hamiltonian for K-SAT may be written such that the energy is a measure of the unsatisfied clauses, and solved with the replica method. Then $E = 0$ signifies a satisfied instance [63]. Recently, K-SAT has been the subject of interest in statistical physics [61] [54]. Kirkpatrick and Selman showed that the transition in K-SAT is similar to the spin glass transition, they applied finite size scaling to find scaling exponents for $2 \leq K \geq 6$ [49].

3.1.4 Vertex cover

The vertex cover (VC) problem is a graph theory problem that was identified by Karp as NP-complete [44]. Before we formulate vertex cover, I should define some basic concepts in graph theory. A graph is a set of vertexes V and a set of edges E. For the purpose of this thesis each edge is defined by a pair of vertices (i,j) which are connected by an edge, we might also think of this as a bond. We define the cardinality of a set as the size of a set, for example, the cardinality of the set of vertexes V is the number of vertexes, N. The average number of edges incident on a vertex can be found by,

$$c = \frac{2 \mid E \mid}{N}, \tag{3.16}$$

where $\mid E \mid$ is the cardinality of the edge set. There is are critcial coordinations, c^*, changes in the lattice occur. For example, we might talk about the coordination where percolation of the lattice occurs, see appendix A. The probability a bond exists is,

$$p = \frac{c}{z}, \tag{3.17}$$

where z is the maximum coordination of the lattice. Thus the corresponding critical probability occurs at,

$$p_C = \frac{c^*}{z}. \tag{3.18}$$

In general, covering can be formulated as: *given any graph* $G = (V, E)$*, what is the minimum number of vertexes that need to be covered such that a particular fraction of edges are covered?* An edge (i, j) is said to be covered if at least one if its end points is in V_{vc}, the set of covered nodes. $V_{vc} \subset V$ and is a full vertex cover if and only if all edges in G have at least one endpoint in V_{vc}.

The minimal vertex cover problem consists of finding a full vertex cover V_{vc} with the smallest cardinality, $\mid V_{vc} \mid$. In the remainder of this dissertation, we will assume that the fraction of covered edges is set to one. The fraction of covered nodes is,

$$x_C = \frac{\mid V_{vc} \mid}{N}. \tag{3.19}$$

The decision variant of the problems asks; *can a full vertex cover be constructed with a fixed fraction,* x*, of covered nodes?*

For those unfamiliar with graph theory, vertex cover can be visualized by the analogy of a museum, with a series of hallway and the junctures where the hallways meet. The museum has a number of guards that stand in the junctures and protect any hallway they can see down. Then we can pose vertex cover, by asking how many guards does the museum need, such that there is at least one guard looking down all hallways.

Vertex cover can be shown to be equivalent to the ground state of the hard core lattice gas [94], and shows a second order phase transition in a random graph [34]. Using the equivalence to a hard-core lattice gas and the replica method Weigt and

Hartmann showed that the replica symmetric solution was accurate to $c^* = e$ on a random graph [94] [33], here c^* is the replica symmetry breaking critical point. Hartmann and Weigt also calculated a typical time to find a solution for vertex cover using a backtracking algorithm [91].

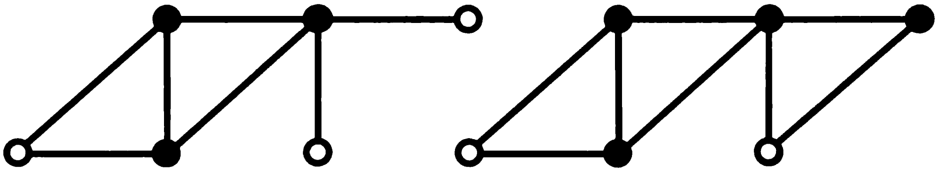


Figure 3.1: A solution for VC and MIS on two clusters. The solid vertexes are in the minimum vertex cover, while the open circles are in the maximum independent set.

Figure 3.3 shows two clusters with a calculated minimum vertex cover. In the left cluster, all the edges can be covered by selecting the three solid vertexes. This cover is denegerate however as several of the covered vertexes may be swapped for the uncovered (open) vertexes. The addition of a new edge in the right cluster, requires that one more edge must be covered, increasing the cardinality of the VC by one.

3.1.5 Maximum independent set

The Maximum Independent Set(MIS) problem is closely related to VC. We define and independent set (IS) by; *given a graph $G = (V, E)$, a set of vertexes V_{IS}, where $V_{IS} \subset V$, is independent if no vertex in V_{IS} is connected to another vertex in V_{IS} by an edge.* The maximum independent set involves finding the independent set with the largest cardinality (known as the independence number), the fractional MIS is given by

$$\rho_{MIS} = \frac{\mid V_{MIS} \mid}{N}. \tag{3.20}$$

The MIS and the minimum VC are related by,

$$1 = x_c + \rho_{MIS}, \tag{3.21}$$

so that,

$$V_{MIS} + V_{vc} = V. \tag{3.22}$$

Physically, we find that the maximum independent set is the same as the maximum density of the hard core lattice gas, at $T = 0$.

Again we may pose a decision problem: *Given graph G is there an independent set of size* ρ *greater than or equal to some maximum,* $\rho \geq \rho_{max}$; or an optimization problem, *what is the maximum size for an independent set in Graph G*? Throughout the course of the work done for this thesis we concentrated on the optimization form of the VC and MIS problems.

Figure 3.3 shows two clusters with a calculated maximum independent set, the open vertexes. In the left cluster, three vertexes is the maximum number of vertexes that may be found without a connecting egde. This MIS is denegerate however as the location of some open vertexes may be moved. The addition of a new edge in the right cluster, requires that one more edge must be covered, decreasing the cardinality of the MIS by one.

3.1.6 NP-complete games

To illustrate the commonness of NP-Complete problems, here is a small list of common NPC problems, for a more extensive list please see the book by Garey and Johnson [29].

- Tetris [17] [10]
- Minesweeper [47]

- Sudoku (Latin Squares) [96]
- Mastermind [86]
- Timetable design [21]
- Multiprocessor scheduling [29]

3.2 Mappings

3.2.1 Mapping K-SAT to vertex cover

One can map a K-sat instance to vertex cover by placing a pair of nodes in the graph for each variable. An edge is placed in the graph connecting the node representing variable, x_i, and the node representing the negated variable, $\bar{x}_i$. For each literal in a clause we place a node into the graph (K nodes for each clause, a_i^1 to a_i^K), with edges connecting each of these nodes. Then the clause nodes are connected to the nodes representing the variables that occur in the clause. Thus for a 3-SAT instance involving N variables and M clauses, there will be a total of $2N + 3M$ nodes in the graph, and $6M + N$ edges. The resulting graph for eq. 3.15 is displayed in Figure 3.2.1

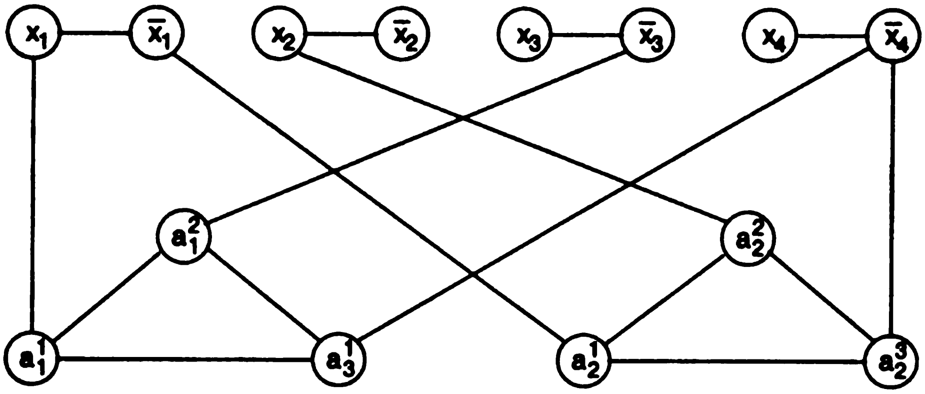


Figure 3.2: This is the mapping of Eq. 3.15 to a graph for vertex cover. Figure and example taken from Hartmann and Weigt [34].

3.2.2 Mapping vertex cover to the hard core lattice gas

Having defined both the hard-core lattice gas and the vertex cover problem, we would like to show how vertex cover may be encoded as a hard-core lattice gas. We will be using Weigt and Hartmann's notation [94]. First we need to define the site occupancy variables,

$$y_i = \begin{cases} 0 \text{ if } i \text{ is covered} \\ 1 \text{ if } i \text{ is uncovered} \end{cases} . \tag{3.23}$$

The constraint that no two adjacent sites may be occupied is enforced by the indicator function,

$$\chi(\vec{x}) = \prod_{(ij)} (1 - y_i y_j). \tag{3.24}$$

When the indicator function is one, the set $\vec{x} = (x_1, \ldots, x_N)$ corresponds to a vertex cover. This allows us the write the grand partition function,

$$\Xi = \sum_{y_i=0,1} \exp(\mu \sum_i y_i) \chi(\vec{x}). \tag{3.25}$$

Weigt and Hartmann found the replica symmetric solution for this expression [94]. Here I will state the solution, see Appendix B for the full treatment. The density of the vertex cover is,

$$x_c = 1 - \frac{2W + W^2}{2c}. \tag{3.26}$$

Note: $W(c)$ is the Lambert function and is given by the solution to $We^W = c$. The fraction which is always covered (frozen covered),

$$x_{fc} = W/c. \tag{3.27}$$

The fraction which is sometimes covered (degenerate),

$$x_d = W^2/c. \tag{3.28}$$

The fraction which is never covered (frozen uncovered),

$$x_{fu} = 1 - (W + W^2)/c. \tag{3.29}$$

3.3 Analytic results for VC: vertex cover on trees

As an approximation to this problem, we consider Bethe lattices. A Bethe lattice or a Cayley tree is a lattice, where a node at each level, $l+1$, is connected to at most α nodes at the next lower level, l. Each node then has a maximum connectivity of $z = \alpha + 1$. The structure looks like a tree branching out toward the lower levels. The center node connects z tree structures.

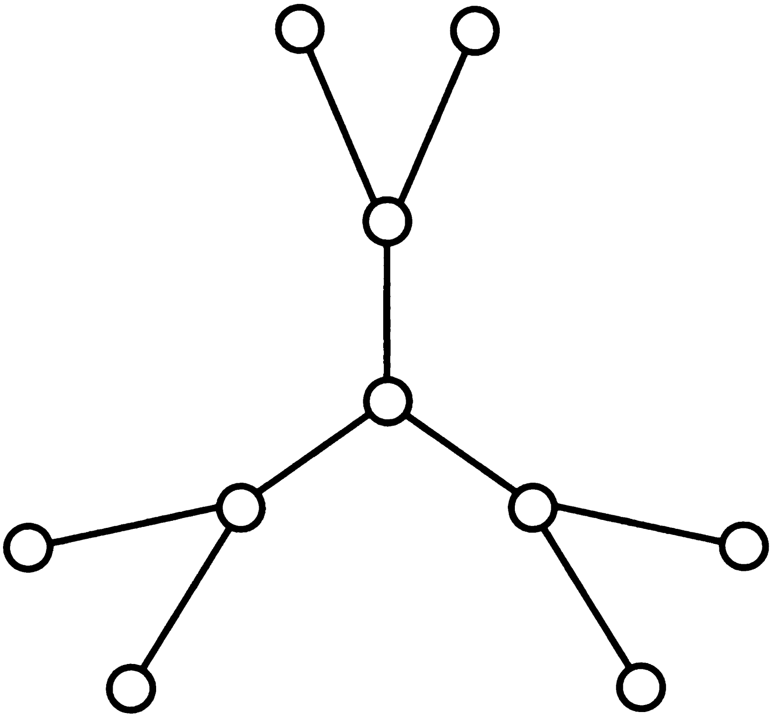


Figure 3.3: The center node and first 2 levels of a Bethe lattice where $z = 3$.

The limiting probability that a site is part of the maximal independent set, P_I^x, is given by the equation,

$$P_I^x = (1 - pP_I^x)^\alpha \tag{3.30}$$

where p is the probability a bond exists. A site is independent if none of its connected neighbors are part of the MIS. In the random graph limit, this yields,

$$P_I^x = \exp(-cP_I^x) = \frac{W(c)}{c}. \tag{3.31}$$

The probability that a site is degenerate, P_d, meaning that it is sometimes part of the cover and sometimes not is given by,

$$P_d = \alpha p P_I^x (1 - pP_I^x)^{\alpha - 1} = cP_I^x \exp(-cP_I^x) = c(P_I^x)^2 = \frac{W(c)^2}{c} \tag{3.32}$$

A site is degenerate if it is singly connected. The degenerate sites have probability 1/2 of being on the minimal vertex cover, so on average, the probability that a site is on the minimal vertex cover is,

$$P_v = 1 - \frac{W(c)}{c} - \frac{W(c)^2}{2c} \tag{3.33}$$

and the probability a site is in the MIS is,

$$P_{MIS} = \frac{W(c)}{c} + \frac{W(c)^2}{2c} \tag{3.34}$$

Note: $P_I^x = W(c)/c$. Since the maximum independent set, or the repulsive lattice gas, on random graphs is highly degenerate, we may define three types of sites: the fraction that is always covered x_{fc}, the fraction that is always uncovered, x_{fu}, and the fraction that is sometimes covered x_d as,

$$\begin{aligned} x_{fc} &= W/c & (3.35) \\ x_{fu} &= 1-\rho_b-\rho_e = 1-(W+W^2)/c & (3.36) \\ x_d &= W^2/c & (3.37) \end{aligned}$$

For a given random graph, all of the degeneracy occurs due to rearrangements of the atoms on the sites with volume fraction (x_d). One can also define the probability that a site is frozen P_f

$$P_f = x_{fc} + x_{fu} = 1 - W^2/c \quad (3.38)$$

The results for x_d, x_{fc} and x_{fu} on trees reproduce those found by Hartmann and Weigt using the replica symmetric theory Eqs. (3.27)-(3.29) (see Appendix B) [33] [94]. However, the Bethe lattice approach generalizes their analysis to random graphs with maximum coordination, $z = \alpha + 1$.

Chapter 4

Algorithms

One principle difficulty in examining *NP-complete* problems is the inability and perhaps impossibility of constructing algorithms that are efficient for large samples. In physics, typically we prefer to work at or near the thermodynamic limit, which is nowhere near the sizes that can be efficiently examined with exact algorithms. The best exact algorithms typically top out around a few hundred sites. This has channeled research for efficient solvers in several directions. One avenue, the development of exact algorithms is slowly becoming more efficient. The conjecture in computer science is that no algorithm capable of solving the worst case in less than exponential time will be found. Another avenue, centers around heuristic algorithms of polynomial complexity that provide near optimal solutions, for large lattices. Often, it is sufficient to get near the ground state, though in physics the structure of the ground state and subsequent excitations are the meat of our meal. Heuristics can illuminate behaviour that might be exploited in exact algorithms, and give a rough idea to the energy landscape. The push for new solvers has also generated interest in quantum computers where the multiplicity of states might be exploited.

Another important question is the difference between the worst case scenario and the average behaviour of the problem. Often the typical time to find a solution is less than the exponential time of the worst case scenario.

In this chapter, we will examine some exact algorithms provided by Alexander Hartmann, and an exact algorithm of complexity $O(2^{N/3})$ by Tarjan and Trojanowski [87]. The algorithm of Tarjan and Trojanowski leads to the graph reductions by leaf and triangle removal (the two simplest cases in their algorithm). Linear reduction of graphs has long been understood as a way to reduce the size of a problem by removing easy elements from a more complex graph. Karp applied leaf removal to matching [46]. Bauer and Golinelli applied the first case of Tarjan's algorithm, as core percolation by leaf removal, to the problem of VC on random graphs and found a phase transition at $c^* = e$ where a core of nodes that could not be resolved by leaf removal emerged [4]. This is of interest as $c^* = e$ is the replica symmetry breaking point on a random graph [4]. This phase transition will be examined in more detail in chapter 5. Recently, Correale et. al. applied leaf removal to simple Boolean networks [15]. We applied leaf removal to some regular lattices and have begun to examine triangle removal on the FCC lattices, triangular lattices and random graphs [38] [39]. After examining the exact algorithms, we will examine several heuristic algorithms. The two simplest heuristic algorithms, greedy and random selection will be examined briefly, followed our by local probability methods centered on node probabilities and edge probabilities.

4.1 Exact algorithms

Exact algorithms for NP-complete problems are an area of intense study. The most naive algorithms resort to exhaustive search, which is very inefficient. It is widely conjectured that no exact polynomial time algorithm exists, so there has been interest in super-polynomial time algorithms [95]. One important aspect of the search for faster exact algorithms is proving performance bounds of particular algorithms. We are not going to go into a lot of detail here on the fastest algorithms currently being used. There is a good review of exact algorithms for a variety of NPC problems by Gerhard Woeginger from 2003 [95]. We will discuss two basic method, *branch_and_bound* and *divide_and_conquer*, then the exact algorithm by Tarjan and Trojanowski, which has a complexity of $O(1.2599^N)$, and leads to some polynomial time graph reductions.

Alexander Hartmann kindly provided us with a program that finds exact ground states using a combination of *branch_and_bound* and *divide_and_conquer* methods [94]. Both algorithms yield a solution tree where every vertex is either covered or uncovered, this tree has 2^N possible leaves. This type of tree is called a binary backtracking tree, at each step there are two branches; one where the relevant node is covered and one branch where it is uncovered, vertexes that have not been reached in the solution tree are left free.

4.1.1 Divide and conquer

A divide and conquer algorithm is based on the concept that a vertex cover can be constructed from the covers of sub-graphs of graph G [94]. The goal then is to split G into sub-graphs and solve them individually. If the graph has a low connectivity, this works well, allowing it to be easily divided. At each step as edges are covered, they are removed allowing for more subdivision of the graph. As the graph moves toward higher connectivities, the graph becomes difficult to subdivide and the algorithm is less effective [94]. A pseudo-code for divide and conquer is given below.

```
DIVIDE_AND_CONQUER($G$)
get a free vertex $i$ of largest current degree $d_i$
mark $i$ as covered
```

```
$numcover_1 \leftarrow 1$
remove all edges $\{i, j\}$ from $E$
calculate all connected components$\{C_i\}$ of graph built by free vertexes
for all components $C_i$
    do
        $numcover_1 \leftarrow numcover_1 +$ divide_and_conquer$(C_i)$
insert all edges$\{i, j\}$ which have been removed

mark $i$ uncovered, $\vec{x}_2 \leftarrow x_i \leftarrow 0$
$numcover_2 \leftarrow 0$
cover all nodes $j$ adjacent to $i$ and remove incident edges $\{j, k\}$
calculate all connected components $\{C_i\}$;
for all components $C_i$
    do
        $numcover_2 = numcover_2 +$ divide_and_conquer$(C_i)$
for neighbors $j$ of $i$
    do
        mark $j$ as free
insert all edges $\{j, k\}$ which have been removed

mark $i$ as free
if $numcover_2 > numcover_1$
  then
        return $(numcover_1)$
  else
        return $(numcover_2)$
```

The divide and conquer algorithm initially will find a greedy cover (see section 4.3.1) based upon covering the highest connected nodes (at each call of the function). This function searches every possible branch, returning the best solution.

4.1.2 Branch and bound

For graphs with higher connectivities, branch and bound methods are better. In the branch and bound algorithm sub-trees of the binary backtracking tree are trimmed when they cannot lead to a better solution. The bounding is controlled by keeping the size of the smallest found vertex cover, we will call this *"current_min"*. The *current_min* is initialized to the number of nodes in the lattice. Nodes are covered (as we descend in tree) and the number of currently covered nodes is kept in X. At each step we choose the highest current degree node to cover, this means that the first descent through the solution tree is a greedy cover (see section 4.3.1).

At any step, if we want to improve on or match the *current_min*, we need to cover $F \leq current\_min - X$ nodes further down in the tree. The boundary is enforced if the number of free (uncovered) edges is greater than the sum of the degrees of the F highest degree free nodes, then a cover smaller than *current_min* cannot be constructed. This sub-tree is trimmed and the algorithm backtracks to find the next possible sub-tree [94]. The advantage to such an algorithm is that it can clearly provide the cardinality of minimum vertex cover and all associated ground states. A pseudo code for branch and bound is provided below.

```
BRANCH_AND_BOUND($G, current\_min, X$)
if all edges are covered
  then
       if $X < best$
          then
```

```
            $current\_min \leftarrow X$
        **return**

calculate $F \leftarrow current\_min - X, D \leftarrow \sum_1^F d_i$
**if** $D <$ number of uncovered edges
  **then return** ▷ Bound

take one free node $i$ with largest degree $d(i)$
cover $i$
$X \leftarrow X + 1$
remove all $(i, j)$ from $E$
branch_and_bound$(G, current\_min, X)$
insert all $(i, j)$
$X \leftarrow X - 1$

**if** $X >$ number of current neighbors
  **then**
        uncover $i$
        **for** for all neighbors $j$ of $i$
            **do** mark $j$ as covered
                $X \leftarrow X + 1$
                remove all incident edges $\{j, k\}$ from $E$
        branch_and_bound$(G, current\_min, X)$
        **for** for all neighbors $j$ of $i$
            **do** mark $j$ as free
                $X \leftarrow X - 1$

```

```
        insert all$(j,k)$ that have been removed

mark $i$ as free
return
```

Hartmann implemented this algorithm to analyze vertex cover for connectivities in the regime $4 < c < 10$, and $N \leq 140$ nodes. The *DIVIDE_AND_CONQUER* and *BRANCH_AND_BOUND* pseudo codes are taken from Hartmann and Weight [94].

4.1.3 An algorithm by Tarjan and Trojanowski

Tarjan's and Trojanowski's [87] algorithm for finding a MIS is a branch and bound style algorithm for finding a maximum independent set on a $N-$vertex graph with complexity $O(2^{N/3})$ [87].

Tarjan's algorithm identifies the degree of a vertex, up to $d \leq 6$, makes a determination of the structure around the vertex, from this one can determine the possible independent sets. The algorithm is recursive, often there exists a choice of independent sets, which need be evaluated as the algorithm progresses. From these independent sets the MIS can be determined.

One subset of cases is simple to analyze; this subset is comprised of complete sub-graphs that subtend on the boundary. This is far simpler to state than to implement as finding complete sub-graphs of a graph is in itself an *NP-complete* problem known as clique [29]. A complete graph is a graph where every node is connected to all other nodes, sometimes in graphs, there may be a sub-graph that is completely connected, this is known as a clique. Finding cliques in a graph is a difficult problem. However, small cliques remain accessible, such as leaves and triangles. On lattices though this may be made simpler as the possible cliques are limited by the geometry, such as the absence of triangles in cubic or square lattices. Once a clique that subtends the boundary is identified, the boundary node is placed in the independent set and all other nodes in the clique are covered. This procedure is exact for the cardinality of the VC but may not take into account all degenerate states, *ie.* a vertex cover is found but not all vertex covers are found.

4.2 Reduction of diluted graphs

Often when working with slow algorithms, a certain amount of preconditioning can speed up the calculations. Preconditioning sometimes known as reduction, has long been a way of attacking difficult problems. Sometimes preconditioning may simply involve sorting the list and gathering information that will be necessary for the exact algorithm. An example would be finding the connectivities and sorting them. The sorting of data may lead to an exponentially large gain in the running time [95]. Other times preconditioning may involve a reduction in the complexity of the instance, as in the case of the leaf and triangle removal algorithms featured below. The reduction removes the easy parts of the problem so that the exact solver may concentrate on the difficult problem, but one that is smaller than the original instance.

4.2.1 Core percolation by leaf removal

A leaf is defined as a node with connectivity one, leaf removal is the removal of the node with connectivity one, its adjacent node and all edges connected to the two nodes. We will refer to the node of connectivity one as the"leaf node", its adjacent node as the "root node" and nodes adjacent to the root, excepting the leaf, we will simply call the "adjacent nodes". When a leaf is covered the root node is placed in V_{vc}, or equivalently we place the leaf node in the V_{MIS}. When a

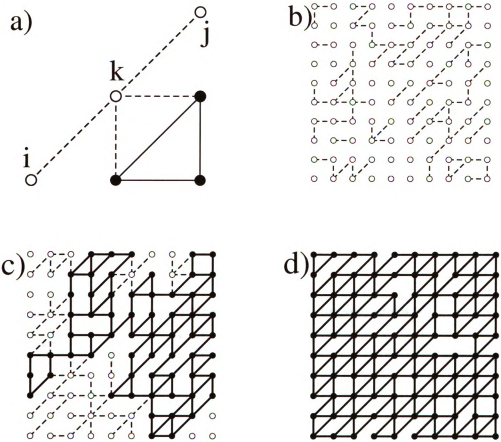
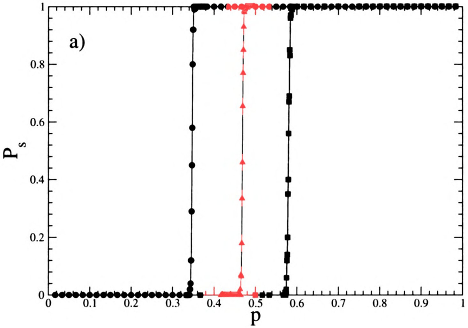


Figure 4.1: Illustration of leaf removal. The dashed edges were originally present, but were removed during the leaf removal process. (a) shows a small graph with 2 leaves when one is detected the root is covered and the dashed edges are removed. (b)-(d) are 100 node planar triangular graphs before and after core percolation. (b) Initial bond concentration of c=1.0, leaf removal removes all bonds from the graph. (c) Initial concentration of c=3.0, after leaf removal a percolating core remains; thus $p > p_c$ (d) Initial concentration of c=4.5, there are no singly connected bonds; thus leaf removal is ineffective in reducing the graph. Note that $c = pz$, where p is the bond probability and c is the average connectivity of nodes in the graph.



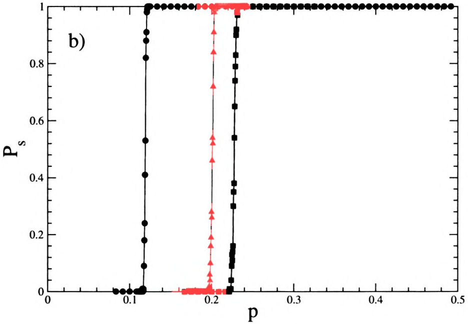


Figure 4.2: The spanning cluster probability of the triangular lattice (a) and FCC lattice (b) for N=1,000,000 node graphs. The black line corresponds to bond percolation, the red to core percolation, and the blue to triangle percolation.

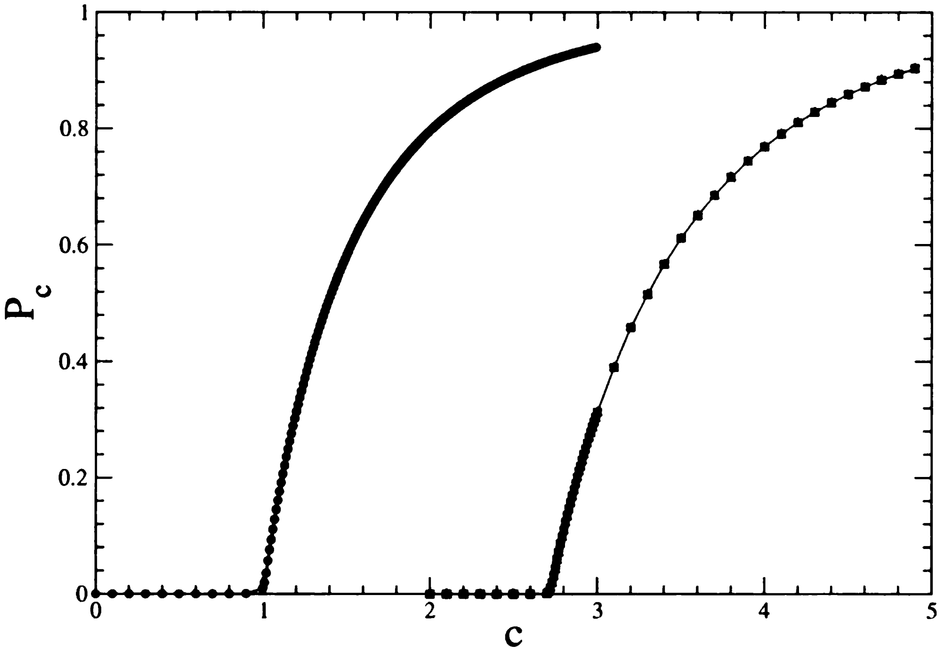


Figure 4.3: The largest cluster probability of the random graph. The black line corresponds to bond percolation, the red to core percolation, and the blue to triangle percolation, for N=1,000,000. Triangle removal and leaf removal coincide on random graphs as $N \longrightarrow \infty$.

leaf is removed, the connectivity of the adjacent nodes are reduced. The algorithm in its most basic terms is then identify a leaf, cover the leaf, remove the leaf, and examine the adjacent nodes for newly generated leaves. Below is a pseudo-code for leaf removal.

```
LEAF REMOVAL($G, \vec{x}$)
Make list $L$ of leaves
while $L > 0$
    do
        Choose leaf $i$ from list $L$
        Find $j$ the root of $i$ ($j = A(i)$)
        $x_i \leftarrow 1, x_j \leftarrow 0$
        for all neighbors $k$ of $j$
```

```
do
    Remove edge $\{j,k\}$
    Adjust $c_k$ the connectivity of $k$
    remove $i$ from $L$
    if $c_k = 1$
      then
          $L \leftarrow k$
          return
```

When the algorithm is finished the number of nodes covered is $| x |$ and the remaining part of the graph is called the core in G. In Figure 4.1(a) we see the results of leaf removal on a small graph with two leaves. When one leaf is chosen (say i or j) then k is covered and the dotted edges are removed, leaving a core.

As we show in chapter 5, below the percolation threshold for the core, the core clusters are logarithmic in size and VC on the core can be solved by an exponential algorithm in polynomial time. The core percolation threshold corresponds to the onset of replica symmetry breaking which sheds some light on the issue of where graphs become difficult to solve. The core has a minimum connectivity of two showing it is the onset of loops (leading to frustration) that creates the computational difficulty of MIS. Figure 4.2 shows the position of the transitions of standard bond percolation, p_b, core percolation by leaf removal, p_l, and core percolation by triangle removal, p_t, on the FCC and triangular lattices.

The evolution of the core occurs in three regimes. In the first regime, the connectivity is very small and so there is no core. In Figure4.1(b) we see the complete removal of the edges of a triangular lattice with initial bond concentration of $p = 1/6$. As the connectivity is increased, small clusters begin to form in the core. We see the emergence of a giant percolating cluster, as in Figure4.1(c). Finally, the graph has no leaves on a triangular lattice with connectivity $p = 3/4$, Figure4.1(d).

The algorithm runs in polynomial in time, which means that it is efficient on low connectivity graphs and we gain some advantage on intermediate connectivity graphs. This implies that one should construct exact solvers that begin with leaf removal before fully exploring the more highly connected nodes.

4.2.2 Leaf removal on the Bethe lattice

We would like to determine the concentration where the core percolates after leaf removal. To do this we will define four probabilities; I_n = A site is part of the MIS and is *not* the core, X_n = A site is *not* part of the MIS and is *not* on the core, I_c =A site is part of the MIS and is on the core, X_c =A site is *not* part of the MIS and is on the core. These probabilities need to have the following properties;

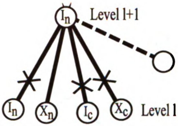


Figure 4.4: I_n

$$I_n + X_n + I_c + X_c = 1, \tag{4.1}$$

and

$$I_c + I_N = P_I \tag{4.2}$$

Where P_I is the probability a site is part of the MIS as found Eq. (3.30) from section 3.3. The probability a site is on the core is,

$$P_c = X_c + I_c. \tag{4.3}$$

For each of the four probabilities we can construct what may occur in lower levels of the tree. To make this easier to visualize, we provide 2 levels of the tree: the top level node in question and the types of connections to lower level nodes, including the possibility of an absent bond (dotted). Bonds will be crossed out when they are explicitly forbidden, and red when they are explicitly needed.

I_n: For a node to be independent and not on the core (see Figure 4.4);

1. it must not be connected to a node on the core,

2. it cannot be connected to another independent site.

This means the node can only be connected to another site that is not independent and not on the core. We write this probability as,

$$I_n = (1 - p + pX_n)^{\alpha}. \tag{4.4}$$

X_n: In order for a node to be covered and not on the core it must be connected to a minimum of one site that is independent and not on the core (see Figure 4.5). Thus we write the probability, X_n as,

$$\begin{aligned} X_n &= \sum_{l=1}^{\alpha} \binom{\alpha}{l} (pI_n)^l (1 - pI_n)^{\alpha - l}, \quad (4.5)\\ &= 1 - (1 - pI_n)^{\alpha - l}. \quad (4.6) \end{aligned}$$

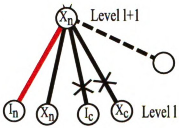


Figure 4.5: X_n

Here we notice that I_n and X_n form a closed set of equations. I_n may now be solved for analytically. Let us continue, however to the probabilities for X_c and I_c.

I_c: All adjacent sites must be unconnected or connected and covered (see Figure 4.6).

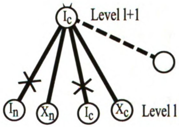


Figure 4.6: I_c

$$\begin{aligned} I_c &= \sum_{l=1}^{\alpha} \binom{\alpha}{l} (pX_c)^l (1 - pX_c - pI_n - pI_c)^{\alpha - l}, \quad (4.7)\\ &= \sum_{l=1}^{\alpha} \binom{\alpha}{l} (pX_c)^l (1 - p - pX_n)^{\alpha - l}, \quad (4.8) \end{aligned}$$

$$I_c = (1 - p + pX_n + pX_c)^{\alpha} - (1 - p + pX_n)^{\alpha} \quad (4.9)$$

X_c: at least one site must sit on the core and be in the MIS, all other sites must be disconnected or connected and covered (see Figure 4.7) .

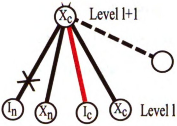


Figure 4.7: X_c

$$\begin{aligned} X_c &= \sum_{l=1}^{\alpha} \binom{\alpha}{l} (pI_c)^l (1 - pI_n - pI_c)^{\alpha - l}, & (4.10) \\ &= \sum_{l=1}^{\alpha} \binom{\alpha}{l} (pI_c)^l (1 - p + p(X_n + X_c))^{\alpha - l}, & (4.11) \\ &= (1 - pI_n)^{\alpha} - (1 - pI_c - pI_n)^{\alpha}. & (4.12) \end{aligned}$$

Eqs. 4.4, 4.6, 4.9, 4.12 satisfy Eq. 4.1 as required. These equations may now be solved numerically.

4.2.3 Core percolation by triangle (and leaf) removal

After leaves, triangles are the next clique. Triangles are identified by finding boundary nodes of degree two, the adjacent nodes are then tested for a connecting edge. If a triangle exists, the boundary node is placed in the IS, all other nodes in the triangle are covered and all edges contained in and incident to the triangle are removed. Figure 4.2 shows the position of the transitions of triangle removal, leaf removal and bond percolation on the FCC and triangular lattices.

Triangle removal is slower than leaf removal due to the step taken in testing for a connection between the adjacent nodes. As stated previously any clique sub-tending on the boundary may be removed with a similar procedure, however as the cliques become larger the number of nodes that need to be tested grows.

On regular lattices however, simple geometry determines the cliques that might be present. For example, on a bipartite lattice such as a square lattice a triangle cannot exist. While, on a lattice such as the triangular or FCC lattice all cliques are size 3 or smaller, so we need not search for anything larger than a triangle

```
TRIANGLE(AND LEAF) REMOVAL($G, N_{cover}, \vec{x}$)
 Make list $L$ of singly connected and doubly connected nodes
 while $L > 0$
    do
        Choose node $i$ from list $L$
       if $c_i = 1$
         then
                Find $j$ the root of i (j=A(i))
                $x_i \leftarrow 0, x_j \leftarrow 1$
                for all neighbors $k$ of $j$
                    do
                        Remove edge $\{j, k\}$
                        Adjust connectivity of $k$, $c_k$
                       if $c_k = 1 || c_k = 2$
                         then
                                add $k$ to $L$
         else $c_i = 2$
                if $i$ part of a triangle
                  then
                        Find $j$ the roots of $i$ ($j = A(i)$)
                        $x_i \leftarrow 0, x_j \leftarrow 1$
                        for all neighbors $k$ of $j$
```

```
                do
                    Remove edge $(j,k)$
                    Adjust connectivity of $k$
                    if $c_k = 1 \| c_k = 2$
                      then
                            add $k$ to $L$
            else remove $i$ from $L$
return
```

4.3 Heuristic algorithms

For larger samples where it becomes impractical to use exact solvers, it is often sufficient to find a near optimal solution. In this case, we can employ a heuristic algorithm [93]. Ideally a heuristic will run in polynomially bounded time. From a physics stand point it should demonstrate some flavor of the actual ground state. I will outline the simple heuristics of greedy cover and random selection, then I will discuss the local probability methods we have developed.

4.3.1 Greedy algorithms

The simplest heuristic is a greedy algorithm [33]. In this case, the algorithm is predicated on the fact that highly connected nodes are more likely to be covered. At each step the highest connected node is covered, when a node is covered the connectivity of adjacent nodes is adjusted. Lovasz [52] showed the greedy algorithm gives an upper bound on hypergraphs of

$$xN \leq (1 + \ln d) \mid V_{vc} \mid, \tag{4.13}$$

x is the fractional cover, N is the number of nodes in the lattice, d is the maximum degree and $\mid V_{vc} \mid$ is the cardinality of the minimum vertex cover.

```
GREEDY COVER($G, \vec{x}$)
$\vec{x} \leftarrow 0$
while Edges are uncovered
    do
        choose node $i$ with largest connectivity
        cover $i$; $x_i \leftarrow 1$
        update connectivity of $i$ and neighbors of $i$

return
```

4.3.2 Random selection algorithm

Another simple heuristic is the random choice algorithm, here the algorithm chooses one edge randomly at a time and covers each end, because of this property the algorithm has an upper bound of,

$$xN \leq 2 \mid V_{vc} \mid, \tag{4.14}$$

```
RANDOM SELECTION ALGORITHM($G, \vec{x}$)
$\vec{x} \leftarrow 0$
while Edges are uncovered
    do
        choose edge $E = \{i, j\}$
        cover $i, j$; $x_i \leftarrow 1, x_j \leftarrow 1$
```

```
        deletes edges incident to $i$ and $j$

return
```

4.3.3 Vertex-Local Probability Recursion (vLoPR)

By examining a local area on a graph, we can see that the probability a vertex is covered is dependent on the probabilities of the surrounding vertexes. If there is a vertex i with v_i neighbors, and every neighbor j is covered then every edge (i, j) is covered and i need not be covered. If, however, any one (or more) of the neighboring nodes is uncovered, then i must be covered.

The probability a vertex i is covered, x_i, exists on the interval $[0, 1]$. When $x_i = 1$, the vertex is covered and when $x_i = 0$ the vertex is uncovered (in the MIS). We can assume that a vertex i may not be covered (or uncovered) in all possible solutions, this node would then be degenerate and its probability would lie between 1 and 0.

Imagine a barbell graph (two nodes with one connecting edge), by inspection we can see that each vertex would be covered exactly one half of the time, so each $x_i = 1/2$, with a degeneracy of 2. Another simple case is chain of three nodes, with two bonds. Here the degeneracy is 1, both outside nodes have $x_i = 0$ and the central node is $x_i = 1$.

This algorithm can be expressed mathematically in a very simple formula;

$$x_i = 1 - \prod_{j=1}^{v_i} x_j. \tag{4.15}$$

This equation can be used to solve for x_i recursively. The average probability $\langle x \rangle = \sum x_i / N$ appears to be at worst an upper bound for the minimum fraction of vertexes covered $\langle x_C \rangle$. We call this type of procedure **Local Probability Recursion**(LoPR).

In the cases, of the barbell, or the 3 node chain, it can be shown that if one begins with the correct probabilities for all the vertexes except one, the correct probability can be calculated for that node. The second case can be generalized to any chain consisting of an odd number of nodes. The algorithm also finds the correct value for the cover probability for a 4-node chain and a square, $\langle x_{4-chain} \rangle = 0.5$, $\langle x_{sq} \rangle = 0.5$. If we were to use a triangle, three vertexes and three edges, then by inspection the cover probability of each x should be $2/3$. However, if we place values of $x = 2/3$ on two of the vertexes and calculate x_i on the third vertex we find that $x_i = 5/9$ after the first iteration and it does not converge to the correct value. In fact, the limiting probability for a triangle is $\langle x_{tri} \rangle = 0.61803$. This probability is a function of the algorithm and not the precision of the machine, as the same result was obtained on a 32-bit Pentium chip running Linux and a 64-bit Turion chip running Linux.

4.3.4 Analytic solution of vLoPR

Now if we want to find an analytic solution for the vLoPR algorithm, we make the probability x_i a constraint on the probability P of a vertex having a particular cover probability, x. For a vertex with v neighbors this will be written as;

$$P_v(x) = \int dx_1 \ldots \int dx_v P(x_1) \ldots P(x_v) \delta(x - 1 + \prod_{j=1}^{v} x_j). \qquad (4.16)$$

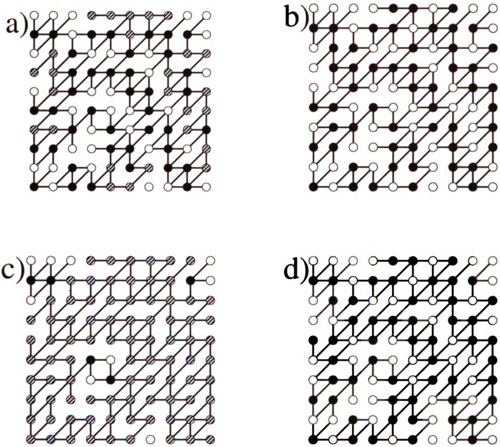


Figure 4.8: The minimum vertex cover on a 100 node diluted triangular lattice. a) The probabilistic solution found using the vertex-LoPR algorithm. The solid circles are nodes where a guard is necessary, the hatched nodes are ones, where the node is degenerate, and the open nodes are where a guard is not necessary. b) A specific minimum vertex cover generated from the vertex-LoPR probabilities using DIG. (c) The probabilistic solution found using the bond-LoPR algorithm. (d) A specific instance generated from (c) using DIG. The minimum cover for this graph is 54 as was confirmed by finding the exact cover using an exact solver, and found by both bond and site algorithms.

Using the identities,

$$
\begin{aligned}
P(x) &= \sum_{v=0}^{\infty} \frac{e^{-c}c^{v}}{v!} P_v(x) && (4.17) \\
\langle x \rangle &= \int_0^1 xP(x)dx, && (4.18)
\end{aligned}
$$

We can now write,

$$
\begin{aligned}
\int_0^{\infty} x \sum_{v=0}^{1} \frac{e^{-c}c^{v}}{v!} P_v(x) &= \int_0^{\infty} x \sum_{v=0}^{1} \frac{e^{-c}c^{v}}{v!} \int dx_1 \dots \\
&\quad \int dx_v P(x_1) \dots P(x_v) \delta(x - 1 + \prod_{j=1}^{v} x_j), && (4.19)
\end{aligned}
$$

and

$$
\begin{aligned}
\langle x \rangle &= \int_0^{\infty} xdx \sum_{v=0}^{\infty} \frac{e^{-c}c^{v}}{v!} \int dx_1 \dots \\
&\quad \int dx_v P(x_1) \dots P(x_v) \delta(x - 1 + \prod_{j=1}^{v} x_j), && (4.20)
\end{aligned}
$$

Now, performing the integration over x on the right hand side to get,

$$
\begin{aligned}
\langle x \rangle &= \sum_{v=0}^{\infty} \frac{e^{-c}c^{v}}{v!} \int dx_1 \dots \\
&\quad \int dx_v P(x_1) \dots P(x_v) (1 - \prod_{j=1}^{v} x_j). && (4.21)
\end{aligned}
$$

The integrals can now be separated, and since each integral is identical they can be written as a product,

$$
\begin{aligned}
\langle x \rangle &= 1 - \sum_{v=0}^{\infty} \frac{e^{-c}c^{v}}{v!} \prod_{j=1}^{v} [\int dx_j x_j P(x_j)] && (4.22) \\
&= 1 - \sum_{v=0}^{\infty} \frac{e^{-c}c^{v}}{v!} \langle x \rangle^{v}. && (4.23)
\end{aligned}
$$

So what we finally get is,

$$\langle x \rangle = 1 - e^{-c + c\langle x \rangle}. \tag{4.24}$$

And the probability $I = 1 - x$ of a vertex being in the independent set is,

$$\langle I \rangle = e^{-c\langle I \rangle}, \tag{4.25}$$

which is the same result that was derived in Eq. (3.31), for the maximum independent set on a Bethe lattice.

4.3.5 The vertex-LoPR algorithm

vLoPR is an iterative method, where each vertex is given a starting probability drawn from a uniform random sampling between $[0, 1]$; there is no bias in selecting vertexes. It would be possible to choose some initial probabilities based on local structure such as leaves. Once the probabilities have been assigned, the node list is randomized, a vertex i is selected from the list and a new cover probability x_i is calculated, using $vLoPR(G, x)$.

All vertexes are updated, and after each full sweep the averaged difference of the site probabilities,

$$\langle \delta x \rangle = \langle x_i - x_i^{previous} \rangle, \tag{4.26}$$

is calculated and iterations are stopped if $\langle \delta x \rangle < BOUND$. Typically a $BOUND = 5 \times 10^8$ was used.

```
FUNCTION VLOPR($G, \vec{x}$)
while $\langle \delta x_i \rangle \geq$ BOUND
    do
        Randomize node list
```

```
4        $SUM \leftarrow 0$
5        for $i \leftarrow 1$ to $N$ from Randomized list
6            do
7                $x_i^{previous} \leftarrow x_i$
8                $PRODUCT \leftarrow 1$
9                for all neighbors $j$ of $i$
10                   do
11                       $PRODUCT \leftarrow PRODUCT \times x_j$
12
13               $x_i \leftarrow 1 - PRODUCT$
14               $\delta x_i \leftarrow x_i - x_i^{previous}$
15               $SUM \leftarrow SUM + \delta x_i$
16
17       $\langle \delta x_i \rangle \leftarrow SUM/N$
18
19   return
```

Step 3 of randomizing the node list may be placed inside the while loop or before it, with good results (See section 4.3.10). While the algorithm was being tested, the node list was randomized before the while loop. Surprisingly, the results of these tests compare favorably to the results for the algorithm as stated above. In section 4.3.10, I analyze the effect of the location in the algorithm of the randomization procedure.

4.3.6 DIG

Once a graph has been "solved" using the vLoPR method, the graph has still not been resolved into a particular instance. The vertexes have been classified as covered, uncovered or degenerate. The next step is to generate a specific instance. We call the procedure we use the **Discrete Instance Generator**(DIG).

The DIG procedure is simple. We choose a vertex i from a degenerate cluster and uncover that vertex. The vertex list is now re-randomized and vLoPR is run again. In general, this procedure reduces the number of degenerate sites by at least 2. The choosing of degenerate nodes continues until there are no degenerate vertexes left in the graph.

The DIG procedure always ends after a vLoPR run, such that the final generated instance is always a true cover of the graph. We believe that if, the original vLoPR run resulted in a fractional cover, x_c, less than the true minimum fractional cover, then a true minimum fractional cover with be generated from the DIG procedure. Fractional covers with different cardinality fall into discrete instances with a small range of cardinality.

```
FUNCTION DIG($G, \vec{x}$)
while there are degenerate nodes
    do
        choose a node $i$ from list of degenerate nodes
        $x_i \leftarrow 0$
        FUNCTION VERTEX-LOPR($G, \vec{x}$)
        count degenerated nodes
        return
```

4.3.7 Bond-LoPR (bLoPR)

In the previous section we dealt with vertex probabilities irrespective of bonds. Now, we want to create a similar procedure where the probability that a bond is covered is forced to one. If we examine a bond (i,j), with vertex i at one end and j on the other, the vertexes have v_i and v_j neighbors respectively. Now in general there are 4 possible states for the bond to be in, one vertex covered X^{ij}_{10} or X^{ij}_{01}, two vertexes covered X^{ij}_{11}or degenerate X^{ij}_{D}. Just a comment here on the notation. The superscript identifies the edge and the subscript indicated the state of each vertex, 1 if covered, 0 if not covered, D if naturally degenerate, so X^{ij}_{10} means node i is covered and node j is not. These bond probabilities are determined by the $v_i - 1$ and $v_j - 1$ neighbors outside of the bond.

To calculate X^{ij}_{10} we ignore the bond (i,j), if one, or more, of the v_i neighbors are not covered then i must be covered, we also need to ensure that no neighbors of j are uncovered. These two terms can be combined so that the probability that vertex i is covered and j is not becomes,

$$X^{ij}_{10} = \Big(1 - \prod_{k \in n(i)\backslash j}^{v_i} x_k\Big)\Big(\prod_{l \in n(j)\backslash i}^{v_j} x_l\Big). \tag{4.27}$$

This can be then extended to the other 3 probabilities to get,

$$\begin{aligned} X^{ij}_{01} &= \Big(\prod_{k \in n(i)\backslash j}^{v_i} x_k\Big)\Big(1 - \prod_{l \in n(j)\backslash i}^{v_j} x_l\Big), & (4.28)\\ X^{ij}_{11} &= \Big(1 - \prod_{k \in n(i)\backslash j}^{v_i} x_k\Big)\Big(1 - \prod_{l \in n(j)\backslash i}^{v_j} x_l\Big), & (4.29) \end{aligned}$$

$$X^{ij}_{D} = \Big(\prod_{k \in n(i)\backslash j}^{v_i} x_k\Big)\Big(\prod_{l \in n(j)\backslash i}^{v_j} x_l\Big). \tag{4.30}$$

If a simple definition of,

$$Y_{ij} = \prod_{k \in n(i)\backslash j}^{v_i} x_k, \tag{4.31}$$

is made these can rewritten these as,

$$
\begin{aligned}
X_{10}^{ij} &= (1-Y_{ij})Y_{ji}, && (4.32)\\
X_{01}^{ij} &= Y_{ij}(1-Y_{ji}), && (4.33)\\
X_{11}^{ij} &= (1-Y_{ij})(1-Y_{ji}), && (4.34)\\
X_{D}^{ij} &= Y_{ij}Y_{ji}. && (4.35)
\end{aligned}
$$

One can find the probability a vertex i is covered from summing the edge probabilities,

$$x_i = \frac{1}{v_i}\sum_{j=1}^{v_i}\left(X_{10}^{ij} + X_{11}^{ij} + \frac{X_D^{ij}}{2}\right). \tag{4.36}$$

We can also write this as the complementary probability,

$$
\begin{aligned}
x_i &= \frac{1}{v_i}\sum_{j=1}^{v_i}\left(1 - X_{01}^{ij} - \frac{X_D^{ij}}{2}\right), && (4.37)\\
x_i &= 1 - \frac{1}{v_i}\sum_{j=1}^{v_i}\left(Y_{ij}(1-Y_{ji}) + \frac{Y_{ij}Y_{ji}}{2}\right), && (4.38)\\
x_i &= 1 - \frac{1}{v_i}\sum_{j=1}^{v_i}\left(Y_{ij} - \frac{Y_{ij}Y_{ji}}{2}\right), && (4.39)\\
x_i &= 1 - \frac{1}{v_i}\sum_{j=1}^{v_i}\Big(\prod_{k\in n(i)\backslash j}^{v_i} x_k - \frac{1}{2}(\prod_{k\in n(i)\backslash j}^{v_i} x_k)(\prod_{l\in n(j)\backslash i}^{v_j} x_l)\Big). && (4.40)
\end{aligned}
$$

Similarly to the vertex-LoPR algorithm, the bond-LoPR algorithm performs well on the barbell, and the 3-node chain. However, on the 4-node chain and the square it over estimates the cover probability, $\langle x_{4-chain}\rangle = 0.514719$, $\langle x_{sq}\rangle = 0.583433$ While the limiting probability for a triangle is underestimated, $\langle x_{tri}\rangle = 0.585786$.

4.3.8 Analytic solution of bLoPR

Similarly to the analysis of the vLoPR algorithm, we can then write our analytic representation of bLoPR as,

$$\begin{aligned} P_v(x) &= \int dx_1 \dots \int dx_v P(x_1) \dots P(x_v) \delta\Big(x - 1 + \frac{1}{v_i} \sum_{j=1}^{v_i} \Big[\prod_{k\in n(i)\setminus j}^{v_i} x_k \\ &\quad - \frac{1}{2} (\prod_{k\in n(i)\setminus j}^{v_i} x_k) (\prod_{l\in n(j)\setminus i}^{v_j} x_l) \Big] \Big). \end{aligned} \tag{4.41}$$

For a moment we will just look at the delta function and work on simplifying that,

$$\delta\Big(x - 1 + \frac{1}{v_i} \sum_{j=1}^{v_i} \Big[\prod_{k\in n(i)\setminus j}^{v_i} x_k - \frac{1}{2} (\prod_{k\in n(i) j}^{v_i} x_k) (\prod_{l\in n(j)\setminus i}^{v_j} x_l) \Big] \Big).$$

To begin we make the approximation that all vertexes converge to a steady state probability related to the average connectivity v,

$$\prod_{k\in n(i)\setminus j}^{v_i} x_k = \prod_{l\in n(j)\setminus i}^{v_j} x_l. \tag{4.42}$$

We rewrite the delta function as,

$$\delta\Big(x - 1 + \frac{1}{v} \sum_{j=1}^{v} \Big[\prod_{k\neq j}^{v} x_k - \frac{1}{2} \prod_{k\neq j}^{v} x_k^2 \Big] \Big) = \delta\Big(x - 1 + \prod_{k=1}^{v-1} x_k - \frac{1}{2} \prod_{k=1}^{v-1} x_k^2 \Big) \tag{4.43}$$

The analytic equation can be written as,

$$\begin{aligned} P_v(x) &= \int dx_1 \ldots \int dx_v P(x_1) \ldots \\ &\quad P(x_v)\delta\Big(x - 1 + \prod_{k=1}^{v-1} x_k - \frac{1}{2}\prod_{k=1}^{v-1} x_k^2\Big), \qquad (4.44) \\ P(x) &= \sum_{v=0}^{\infty} \frac{e^{-c}c^v}{v!} \int dx_1 \ldots \int dx_v P(x_1) \ldots \\ &\quad P(x_v)\delta\Big(x - 1 + \prod_{k=1}^{v-1} x_k - \frac{1}{2}\prod_{k=1}^{v-1} x_k^2\Big). \qquad (4.45) \end{aligned}$$

Multiply both sides by $\int x dx$ and integrate over dx to get,

$$\begin{aligned} \langle x \rangle &= \sum_{v=0}^{\infty} \frac{e^{-c}c^v}{v!} \int dx_1 \ldots \int dx_v P(x_1) \ldots \\ &\quad P(x_v)\Big(1 - \prod_{k=1}^{v-1} x_k + \frac{1}{2}\prod_{k=1}^{v-1} x_k^2\Big), \qquad (4.46) \end{aligned}$$

which can be reduced to,

$$\langle x \rangle = 1 - \frac{1}{\langle x \rangle} \exp(-c + c\langle x \rangle) + \frac{1}{2\langle x^2 \rangle} \exp(-c + c\langle x^2 \rangle). \qquad (4.47)$$

Equation (4.47) involves the higher moment $\langle x^2 \rangle$, so it is useful to examine other moments. The higher moments can be calculated by multiplying Eq. (4.45) by

$\int x^\alpha dx$ and integrating to get the α^{th} moment,

$$\begin{aligned}
\langle x^2 \rangle &= 1 - \frac{2\exp(-c + c\langle x \rangle)}{\langle x \rangle} + \frac{2\exp(-c + c\langle x^2 \rangle)}{\langle x^2 \rangle} \\
&\quad - \frac{\exp(-c + c\langle x^3 \rangle)}{\langle x^3 \rangle} + \frac{\exp(-c + c\langle x^4 \rangle)}{4\langle x^4 \rangle}, \qquad (4.48) \\
\langle x^3 \rangle &= 1 - 3\frac{\exp(-c(1 + \langle x \rangle))}{\langle x \rangle} + \frac{9\exp(-c(1 + \langle x^2 \rangle))}{2\langle x^2 \rangle} \\
&\quad - \frac{4\exp(-c(1 + \langle x^3 \rangle))}{\langle x^3 \rangle} + \frac{9\exp(-c(1 + \langle x^4 \rangle))}{4\langle x^4 \rangle} \\
&\quad - \frac{3\exp(-c(1 + \langle x^5 \rangle))}{4\langle x^5 \rangle} + \frac{\exp(-c(1 + \langle x^6 \rangle))}{8\langle x^6 \rangle}, \qquad (4.49) \\
\langle x^4 \rangle &= 1 - \frac{4\exp(-c(1 + \langle x \rangle))}{\langle x \rangle} + \frac{8\exp(-c(1 + \langle x^2 \rangle))}{\langle x^2 \rangle} - \frac{10\exp(-c(1 + \langle x^3 \rangle))}{\langle x^3 \rangle} \\
&\quad + \frac{17\exp(-c(1 + \langle x^4 \rangle))}{2\langle x^4 \rangle} - \frac{5\exp(-c(1 + \langle x^5 \rangle))}{\langle x^5 \rangle} + \frac{2\exp(-c(1 + \langle x^6 \rangle))}{8\langle x^6 \rangle} \\
&\quad - \frac{\exp(-c(1 + \langle x^7 \rangle))}{2\langle x^7 \rangle} + \frac{\exp(-c(1 + \langle x^8 \rangle))}{16\langle x^8 \rangle}. \qquad (4.50)
\end{aligned}$$

These as yet have resisted our attempts to write them in a general form.

4.3.9 The bond-LoPR algorithm

```
BOND LOPR($G, \vec{x}$)
  Randomize node list
 while $\langle \delta x_i \rangle \geq$ BOUND
    do
       for $i \leftarrow 1$ to $N$ from Randomized list
          do
             $x_i^{previous} \leftarrow x_i$
             $S \leftarrow 1$
             $S_l \leftarrow S_k \leftarrow 1$
```

```
    for all neighbors $j$ of $i$
      do
        for all neighbors $k$ of $i$
          do
            if $k \neq j$
              then $S_l \leftarrow S_l \times x_k$

        for all neighbors $k$ of $j$
          do
            if $k \neq i$
              then $S_k \leftarrow S_k \times x_k$

        $S \leftarrow S + S_l - 0.5 \times S_l \times S_k$

    $x_i \leftarrow 1 - S$
    $\delta x_i \leftarrow x_i - x_i^{previous}$
    $SUM \leftarrow SUM + \delta x_i$

$\langle \delta x_i \rangle = SUM/N$
return
```

4.3.10 Convergence of vertex and bond LoPR

In order to examine the convergence and performance of the LoPR algorithms we tracked three quantities; The first quantity is the fraction of covered nodes $\langle x \rangle$, the second is the difference between the average fraction of covered nodes and its value for the previous run $\delta\langle x \rangle = \langle x \rangle - \langle x^{previous} \rangle$, the third quantity is the aver-

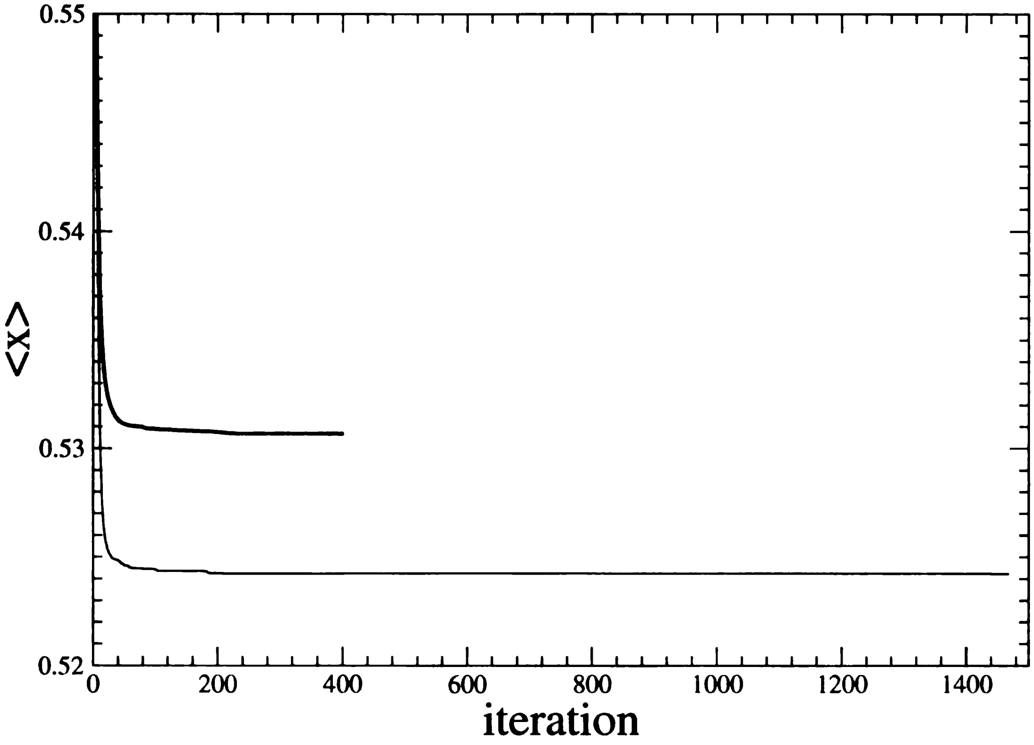


Figure 4.9: The minimum vertex cover as a function of iteration for a specific 10000 node triangular lattice, using the site (thin line) and bond (thick line) LoPR algorithms, while randomizing the ordering of nodes every iteration.

age of the differences between the site probability x_i and its value in the previous run $\langle \delta x \rangle = \langle x - x^{previous} \rangle$.

As a measure of the stability of the algorithm, I also tested running the algorithms with the node list randomized before the while loop, and samples where the node list was re-randomized before each pass through the lattice.

Figs. 4.9-4.14 show the convergence of the site-LoPR and bond-LoPR algorithms on an arbitrary $N = 10,000$ node triangular lattice with connectivity, $c = 3.0$. All simulations were started with the same initial conditions, and run on the same lattice. In Figs. 4.9- 4.11, we see the convergence profiles for the algorithm when the node list is randomized on each pass through the lattice. Figs. 4.12- 4.14 shows the convergence profile when the node list is randomized once.

Figs. 4.9 and 4.12, show that the algorithm converges very quickly to the neigh-

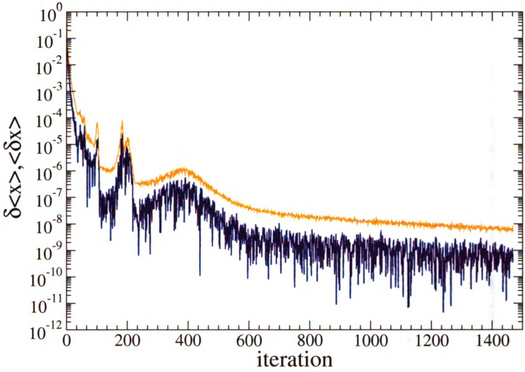


Figure 4.10: $\delta\langle x\rangle$ (orange) and $\langle\delta x\rangle$ (blue) as a function of iteration for the same 10000 node triangular lattice and initial conditions, as the previous figure, using the vertex-LoPR algorithms, while randomizing the ordering of nodes every iteration.

borhood of the correct value. We can see that as $\langle x\rangle$ converges it often does in a stepwise fashion, the value of $\langle x\rangle$ changes quickly. This indicates an avalanche where the probabilities over a significant portion of the graph change dramatically. The algorithm mathematically need not be stepwise since x is continuous. Convergences for these tests were run to a $BOUND = 5 \times 10^{-9}$. For all samples we studied, in chapter 6, convergences were done to a $BOUND = 5 \times 10^{-8}$, where the frozen fraction is largely set. The value of $\langle x\rangle$ over the whole graph converges much faster than the values at individual vertexes. This can be understood by remembering the barbell graph. If one end is covered the other is uncovered $\langle x\rangle = 1$, but if we flip the values of the nodes then $\langle x\rangle$, $(\delta\langle x\rangle = 0)$ remains unchanged but $\langle\delta x\rangle = 1$.

The convergence profiles for vLoPR, as shown in Figure 4.10 and Figure 4.13,

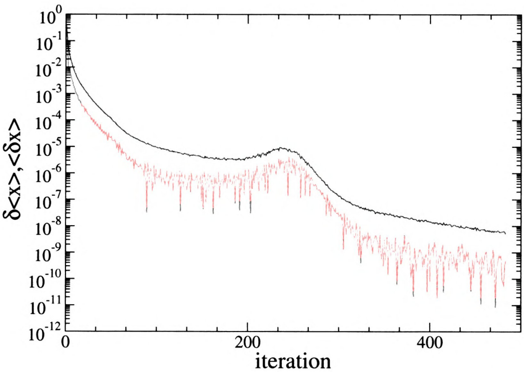


Figure 4.11: $\delta\langle x\rangle$ (orange) and $\langle\delta x\rangle$ (blue) as a function of iteration for the same 10000 node triangular lattice and initial conditions, as the previous figure, using the bond-LoPR algorithms, while randomizing the ordering of nodes every iteration.

illustrate that in this instance the algorithm proceeds through cycles of relaxation followed by reorganization. This continues until the system is unable to drop into a new lower state. This does not mean, however that the final state found is necessarily the global minimum energy state, it is possible that we have found a local minimum. The single list case, is similar to the re-randomized case, but there is a complete lack of noise on the quantities $\delta\langle x\rangle$ and $\langle\delta x\rangle$. Both variants seem to be robust enough to pass over several energy barriers. The noise in the standard version of the algorithm does not have a large effect on the overall covered fraction.

In the bond algorithm, Figs 4.11,4.14, The convergence is comparatively fast taking roughly one third of the iterations of the site algorithm. We see the same ability to pass over some energy barriers that was exhibited by the site algorithm though the effect on the measured values was less drastic, this is due to the fact,

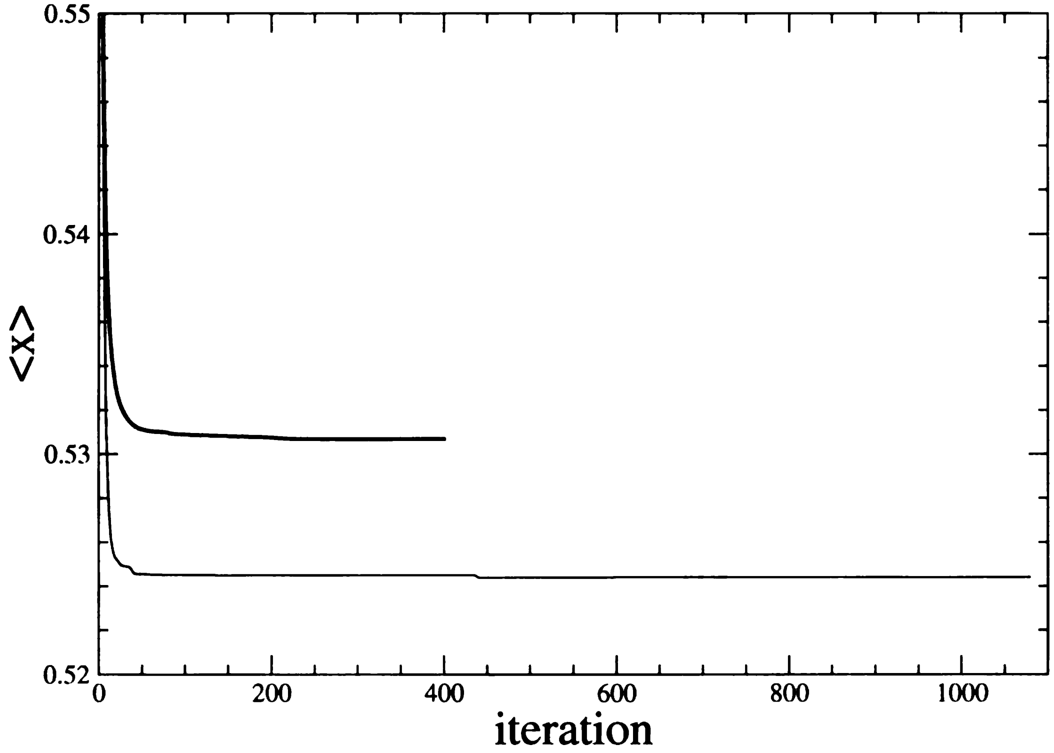


Figure 4.12: The minimum vertex cover as a function of iteration for a specific 10000 node triangular lattice using the site (thin line) and bond (thick line) LoPR algorithms generating one node list used for every iteration.

that in the bond algorithm significantly larger numbers of nodes are "degenerate".

All four convergence tests were run on the same lattices with the same initial node probabilities. The final values converged to in Fig 4.9 are $\langle x \rangle = 0.52424$ for the site algorithm, and $\langle x \rangle = 0.530982$ for the bond algorithm, in Fig 4.12 the simulations run with one node list converged to $\langle x \rangle = 0.524401$ for the site algorithm and $\langle x \rangle = 0.530678$. Neither algorithm correctly counts the degeneracy, this implies that neither algorithm fully explores the whole energy landscape, as such we cannot say they converge to global minima. The practice of repeated reordering of the node list, makes it easier for the algorithms to avoid local minima making the final value of $\langle x \rangle$ much less affected by the initial conditions. It however, does increase the running time, and also makes it less likely to converge to a $\delta\langle x \rangle$ value of much less than 5×10^{-9} on a $N = 10,000$ node graph. In general, good per-

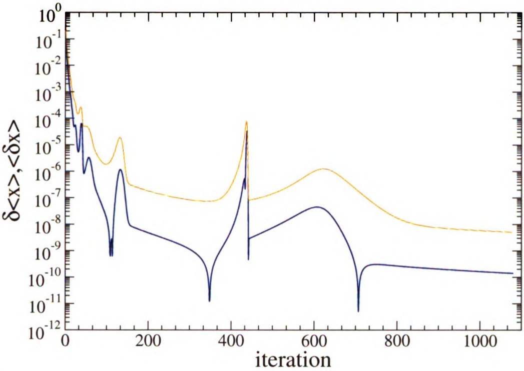


Figure 4.13: $\delta\langle x\rangle$ (orange) and $\langle\delta x\rangle$ (blue) as a function of iteration for the same 10,000 node triangular lattice and initial conditions, as the previous figure, using the vertex-LoPR algorithms generating one node list used for every iteration.

formance was found with both algorithms and with both methods of ordering the node list. In order to more fully compare the effect of ordering the node list, the algorithm was run over 10^5 configurations for the $N = 100$ node triangular lattice displayed in Fig 4.8. Table 4.1 shows a histogram of the results comparing the single ordered nodes list algorithms and the re-randomized node list algorithms for a lattice with a known exact solution, we can clearly see there was no appreciable difference between the methods of ordering the node list. Using the single list LoPR a slightly higher percentage of solutions, about one percent higher, found the correct ground state.

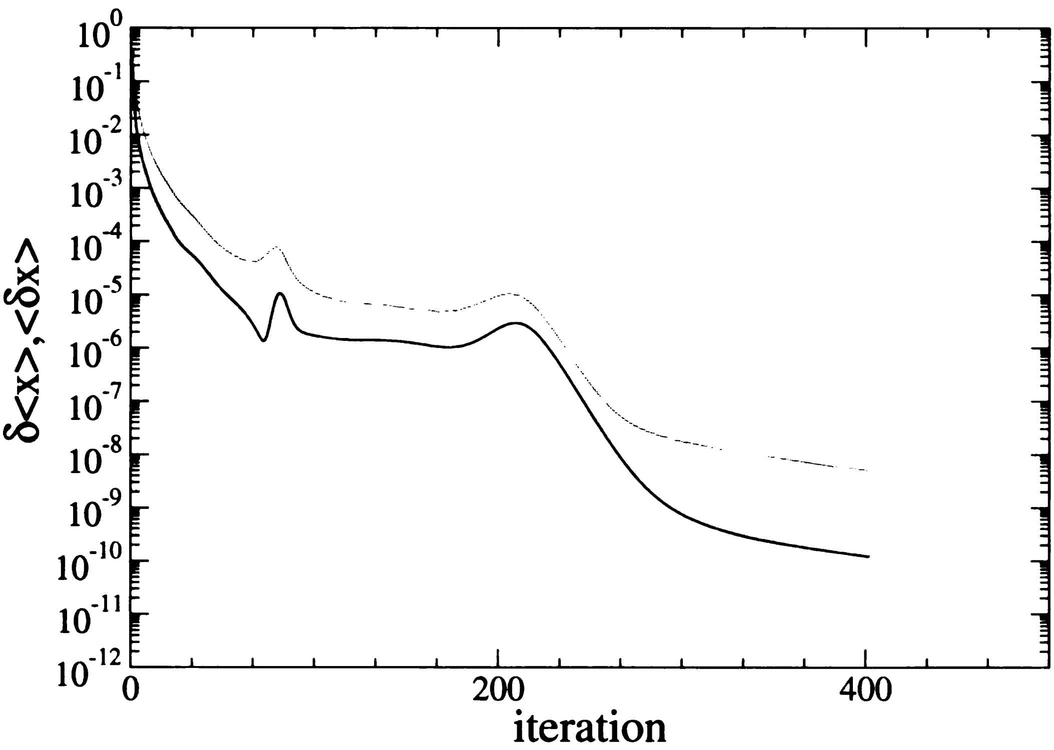


Figure 4.14: $\delta\langle x\rangle$ (blue) and $\langle\delta x\rangle$ (orange) as a function of iteration for the same 10000 node triangular lattice and initial conditions, as the previous figure, using the bond-LoPR algorithms generating one node list used for every iteration.

Table 4.1: Percent of occurrence of solutions for the LoPR algorithms on the 100 node triangular lattice shown in Figure (4.8). The results are averaged over 10,000 runs. The exact solution for the lattice $\langle x\rangle = 0.54$.

	Standard LoPR				Single List LoPR			
$\langle x\rangle$	vLoPR	DIG	bLoPR	DIG	vLoPR	DIG	bLoPR	DIG
0.53	0	0	0	0	0	0	0	0
0.54	55.975	56.66	0	68.505	56.7	57.65	0	69.17
0.55	39.555	40.965	0	31.495	39.02	39.985	0	30.88
0.56	4.365	2.275	100	0	4.2	2.28	100	0
0.57	0.1	0.1	0	0	0.085	0.085	0	0
0.58	0.005	0	0	0	0	0	0	0
0.59	0	0	0	0	0	0	0	0

Chapter 5

Core Percolation

Preconditioning of graphs is a time-honored method of simplifying an instance. This chapter deals with reductions of a graph brought about by the removal of leaves, and triangles. The subsequent remaining graph is known as the core, it is this part of the graph that contains the higher connectivity nodes, and remains computationally hard.

In this chapter, we will discuss the results of leaf removal and triangle removal. First, we will discuss the results of core percolation by leaf removal on an Erdös-Rényi style random graph, the 2-dimensional triangle and square lattices and the 3-dimensional FCC and simple cubic lattices. Then we will discuss the results of core percolation by triangle removal on the random, triangular and FCC lattices. The results of leaf removal on the triangle lattice has been published [40]. The results for leaf removal on the other lattices are in preparation [37] as are the results of triangle removal [38].

5.1 Core percolation by leaf removal on random graphs

As stated previously, leaf removal is a partial application of Tarjan and Trojanowski's algorithm for finding an MIS. It was first applied by Karp and Sipser to matching in 1981 [46]. Subsequently, Bauer and Golinelli [5], analyzed the behaviour of leaf removal on a random graph. Leaf removal has also been applied to simple Boolean networks [15]. Bauer and Golinelli found that above a critical concentration of bonds, there remained a core of nodes and edges that could not be removed by leaf removal. They observed that the core percolation threshold for leaf removal on a random graph occurs at the same critical concentration as replica symmetry breaking. In their paper, they analyzed behaviour at the critical point and drew the conclusion that core percolation is in a different universality class to that of standard percolation [5]. This section will draw an alternative conclusion, namely that core percolation by leaf removal on a random lattice is in the same universality class as standard percolation.

5.1.1 Generation of random graphs

All calculations on random graphs were made using the Erdös-Rényi model. In the Erdös-Rényi model edges are placed at random into the graph with the probability,

$$p = c/N. \tag{5.1}$$

Where, N is the number nodes, and the connectivity, $c = 2E/N$, is the average number of edges per node.

Simulations were done with sample sizes of $N = 100$, $N = 1000$, $N = 10^4$, $N = 10^5$, $N = 10^6$ all samples $N \leq 10^4$ were averaged over 10^4 configurations, graphs of $N = 10^5$ were averaged over 1000 configurations and graphs of $N = 10^6$ were averaged over 100 configurations.

5.1.2 Random graph results

Figs. 5.1-5.3 show the results for the giant cluster after leaf removal. Figure 5.1 is the probability of a node in the graph being on the giant cluster, and Figure 5.2 is the number of edges in the giant cluster as a fraction of the number of nodes. The probability a site is on the largest cluster,

$$P_C = \frac{N_C}{N} \tag{5.2}$$

and at percolation the number of edges in the largest cluster, E_C, scales in standard percolation as,

$$P_C \sim \frac{E_C}{N} \sim N^{\omega-1}, \tag{5.3}$$

where in standard percolation ω takes a value of $2/3$ [6]. Figure 5.3 shows the connectivity of the core as a function of the original connectivity of the graph. The connectivity of the core scales as [20],

$$c_C - 2 \sim N^{-\phi} \tag{5.4}$$

The graphs of P_C, E_C and c_C are in agreement with the results of Bauer and Golinelli [5].

In Figure 5.4, we apply the scaling relations to find ω, ϕ. Here we make the observation that the widths of the mean scale as the mean, thus we find $\omega = 0.66(2)$ and $\phi = 0.22(1)$, which are very similar to the values obtained by Bauer and Golinelli [5] (see table 5.1).

Figure 5.5 shows finite size scaling collapse in the critical regime of the giant

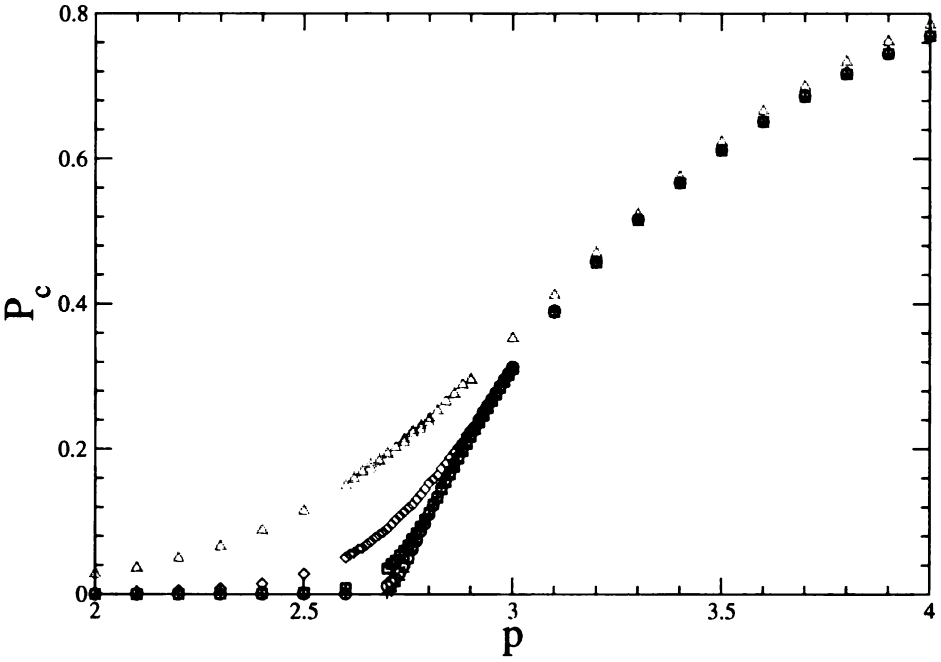


Figure 5.1: Plot of the probability of being on the core versus connectivity for random graphs. Sample sizes and number of realizations, starting from the top trace: N=100(10,000), N=1,000(10,000), N=10,000(10,000), N=100,000(1,000), N=1,000,000(100).

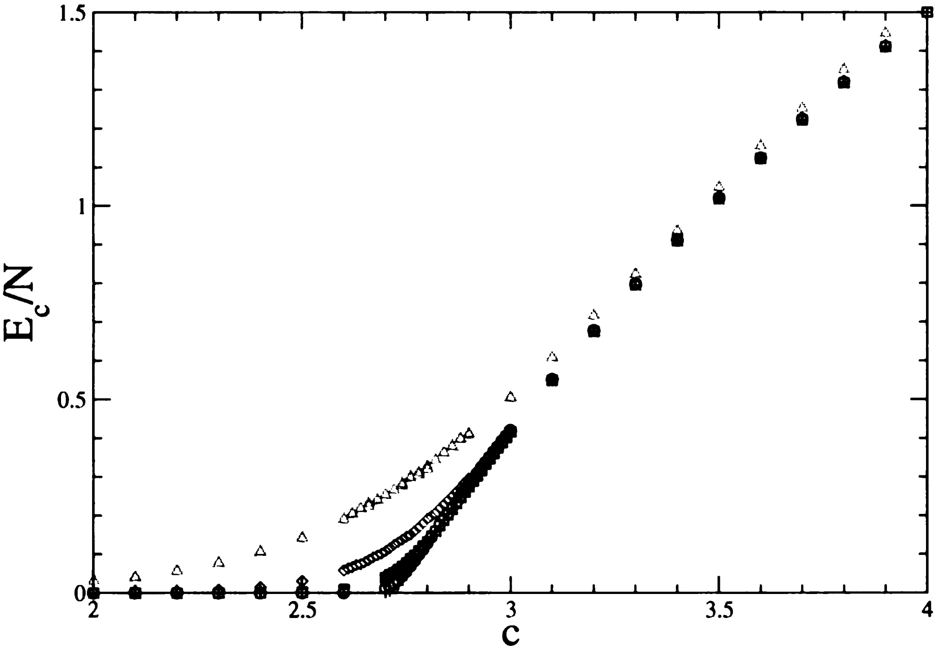


Figure 5.2: Plot of edges in the core cluster versus connectivity. Sample sizes (number of realizations), starting from the top trace: N=100(10,000), N=1,000(10,000), N=10,000(10,000), N=100,000(1,000), N=1,000,000(100).

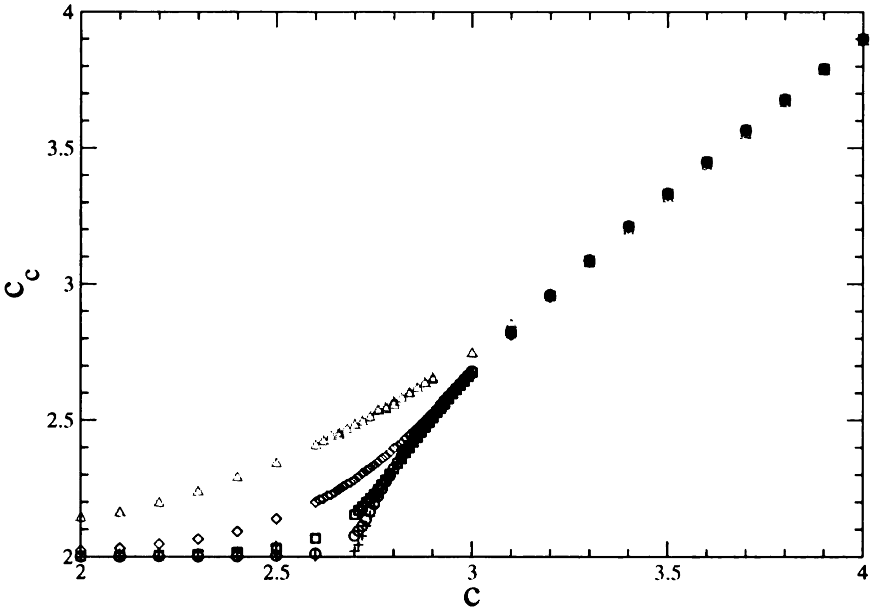


Figure 5.3: Plot of connectivity of the core versus connectivity of original graph. Sample sizes (number of realizations), starting from the top trace: N=100(10,000), N=1,000(10,000), N=10,000(10,000), N=100,000(1,000), N=1,000,000(100).

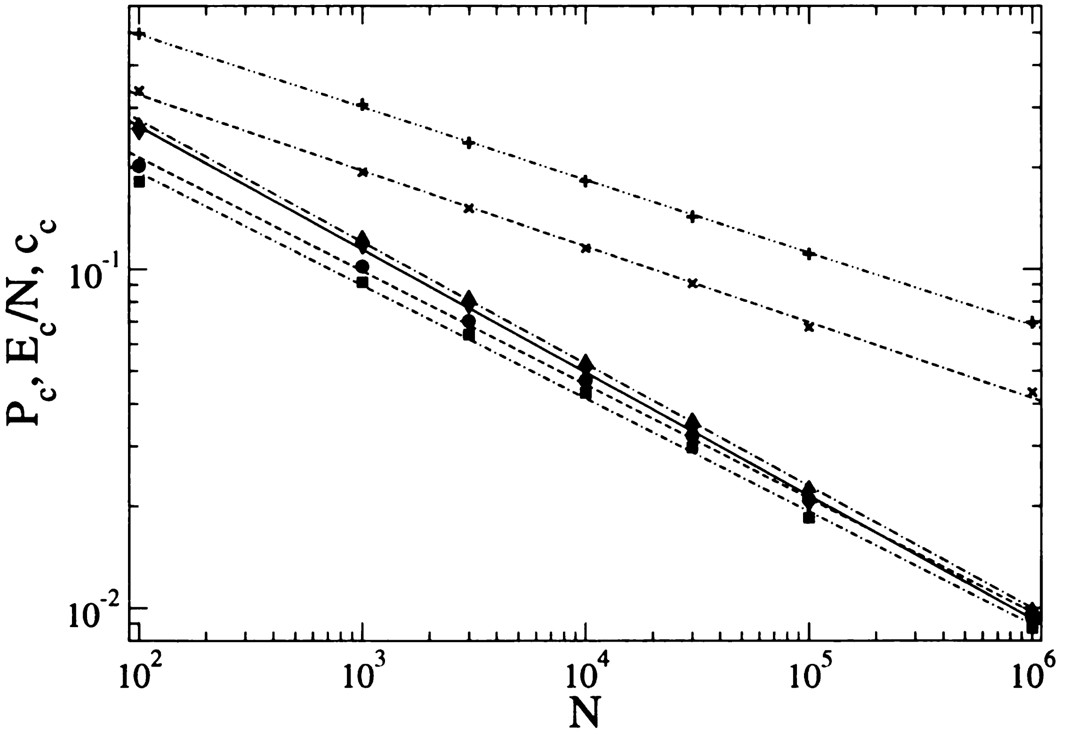


Figure 5.4: Finite size effects of connectivity (+), Edges in core (▼), P_C (•), the variance the connectivity (×), variance in the number of edges in the core (▼), variance of the probability of being on the core (■) at the transition, $c = e$. Sample size (number of realizations), N=100(10,000), N=1,000(10,000), N=3,000(10,000), N=10,000(10,000), N=300,000(10,000), N=100,000(1,000), N=1,000,000(200).

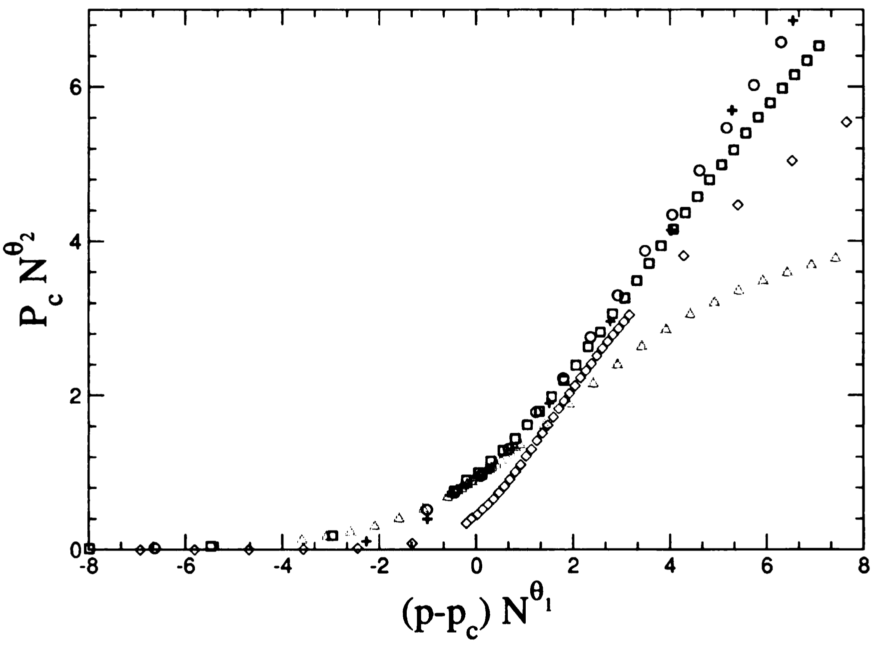


Figure 5.5: Finite size collapse fit with $\theta_1 = 0.35(2)$, $\theta_2 = 0.33(3)$. Sample sizes (number of realizations): $N = 100(10{,}000)(\triangle)$, $N = 1{,}000(10{,}000)(\diamond)$, $N = 10{,}000(10{,}000)(\square)$, $N = 100{,}000(1{,}000)(\circ)$, $N = 1{,}000{,}000(100)(+)$.

cluster using the scaling ansatz,

$$P_c N^{\theta_1} \sim (p - p_c) N^{\theta_2}. \tag{5.5}$$

Traditionally, $\theta_1 = \theta_2$. Here, θ_1 and θ_2 were allowed to vary independently to give a check in relation to the standard scaling ansatz, and to allow more freedom to find a good collapse. The collapse we found is good for values of $N > 10^5$, and adequate for $N = 10^4$. It is hard to fit smaller samples because of increased finite size effects. Upon collapse we have $\theta_1 = 0.35(2)$ and $\theta_2 = 0.33(3)$, verifying the scaling ansatz. In standard percolation,

$$\theta = 1 - \omega \tag{5.6}$$

The exponents we measured are listed in table 5.1. We see that our value of $\phi =$

0.22 is close to Bauer and Golinelli's which they believe to be 1/5 and given their relation of,

$$\omega = 1 - 2\phi, \tag{5.7}$$

convinces them that ω is closer to 3/5, placing core percolation by leaf removal in a new universality class rather than that of standard percolation. It is not simple to say that core percolation is not a new universality class, it is easy to say that what we have observed is consistent with leaf removal on a random graph being in the same universality class as standard percolation. It is likely that higher accuracy experiments in the future will deal with this question in more detail.

Table 5.1: Critical Exponents for core percolation on random graphs

exp.	standard	Bauer and Golinelli	Fay	
θ	1/3	2/5	0.33(4)	Figure 5.5
ω	2/3	3/5	0.66(2)	Figure 5.4
ϕ		1/5	0.22(1)	Figure 5.4

5.2 Core percolation by leaf removal on regular lattices

In this section we study the percolative properties of core percolation on regular lattices, specifically, the triangular, square, simple cubic and FCC lattices. First we will examine the properties of 2-d lattices and then the properties of 3-d lattices.

We focused on three percolative properties: (i) the spanning probability, P_S, the probability that a core cluster spans the sample, (ii) The probability a node is on that spanning cluster P_∞, (iii) the core cluster number $n_S(s,p)$ and the average cluster size $\langle s \rangle$. The scaling behavior of these quantities near a second order transition is given by [84],

$$P_S(p) \sim f_S(\delta p L^{1/\nu}) \tag{5.8}$$

$$P_\infty(p) \sim L^{-\beta/\nu} f_\infty(\delta p L^{1/\nu}) \tag{5.9}$$

$$n_s(p) \sim s^{-\tau} f_n(s^\sigma \delta p) L \to \infty \tag{5.10}$$

$$\langle s \rangle \sim \delta p^{-\gamma} f_a(\delta p L^{1/\nu}) \tag{5.11}$$

where $\delta p =\mid p - p_l \mid$, and p_l is the critical point for leaf removal. The scaling functions, f_s, f_∞, f_n, f_a contain the finite size scaling behaviors, and also allow us to determine the exponents, ν, β and γ. Eq. (5.10) allows us to determine the exponent τ in the infinite lattice limit at p_l. A check on the calculations may be performed since only two of the exponents are independent, and they are related by,

$$\nu = \frac{\beta}{2}\frac{\tau - 1}{\tau - 2}, \alpha + 2\beta + \gamma = 2, d\nu = 2 - \alpha \tag{5.12}$$

The standard percolation critical exponents for $d = 2$ are listed in table 5.3 and for $d = 3$ in table 5.5, with the standard bond percolation threshold occurring at $p_b = 0.34729$ on the triangular lattice, $p_b = 0.5$ on the square lattice, $p_b = 0.119$ on the FCC lattice, and $p_b = 0.2488$ on a simple cubic lattice [84]. The percolation thresholds are list in table 5.2. Raw data for the four lattices are presented in Fig 5.6, 5.7, 5.8, 5.9 and Figs. 4.2 and 4.3. On regular lattices, we can see that the core percolation thresholds occur at much higher connectivities than conventional percolation.

Table 5.2: Percolation thresholds for standard bond percolation (SP) and core percolation by leaf removal (LR) and triangle removal (TR).

lattice	SP, p_b	LR, p_l	TR, p_t
triangular	0.34729	0.4692(5)	0.580(2)
square	0.50000	0.6382(5)	—
FCC	0.119	0.202(1)	0.228(2)
cubic	0.2488	0.393(1)	—
	SP, c_b	LR, c_c	TR, c_t
random graph	1.00	$e \approx 2.7183$	$e \approx 2.7183$

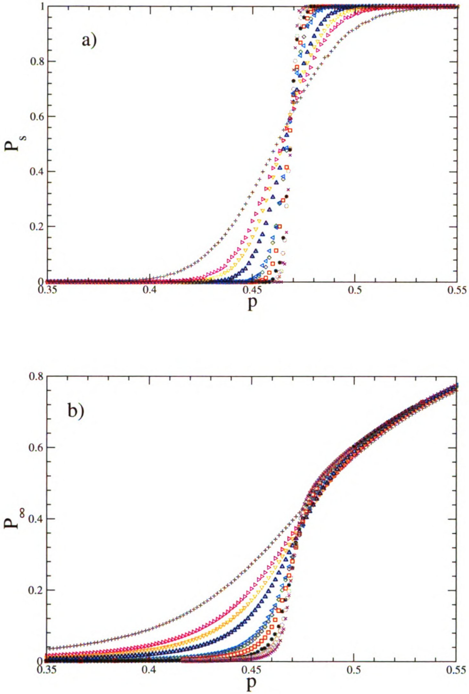


Figure 5.6: Plots of leaf removal on bond diluted triangular lattices. a) Spanning probability P_S as a function of the bond concentration p. b) The infinite cluster probability as a function of p. Sample sizes and number of configurations, $N = 1{,}000{,}000(100)$, $N = 500{,}000(2{,}000)$, $N = 250{,}000(1{,}000)$, $N = 100{,}000(10{,}000)$, $N = 50{,}000(10{,}000)$, $N = 40{,}000(10{,}000)$, $N = 10{,}000(20{,}000)$, $N = 5{,}000(20{,}000)$, $N = 3{,}025(10{,}000)$, $N = 1{,}000(20{,}000)$.

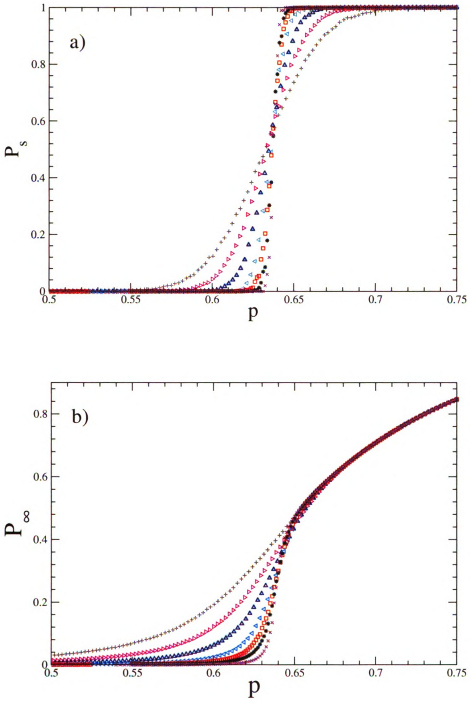


Figure 5.7: Plots of leaf removal on bond diluted square lattices. a) Spanning probability P_S as a function of the bond concentration p. b) The infinite cluster probability as a function of p. Both graphs were done over a range of graph sizes, N, and averaged over a number of configurations. Sample sizes and number of configurations, $N = 1,000,000(100)$, $N = 250,000(1,000)$, $N = 102,400(1,000)$, $N = 40,000(10,00)$, $N = 10,000(10,000)$, $N = 3,025(10,000)$, $N = 1,024(10,000)$.

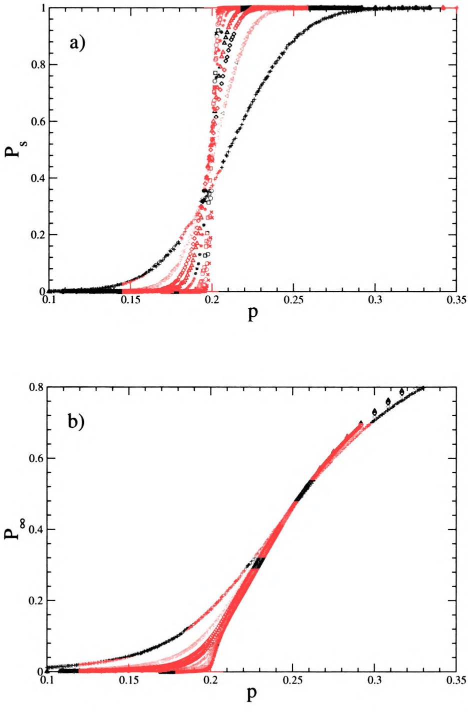


Figure 5.8: Plots of leaf removal on bond diluted FCC lattices. a) Spanning probability P_S as a function of the bond concentration p. b) The infinite cluster, P_∞, probability as a function of p. Both graphs over sizes $N = L^3$(and number of configurations) of $L = 5(10,000)$, $L = 10(10,000)$, $L = 15(10,000)$, $L = 20(10,000)$, $L = 30(10,000)$, $L = 50(1,000)$, $L = 70(1,000)$, $L = 100(100)$.

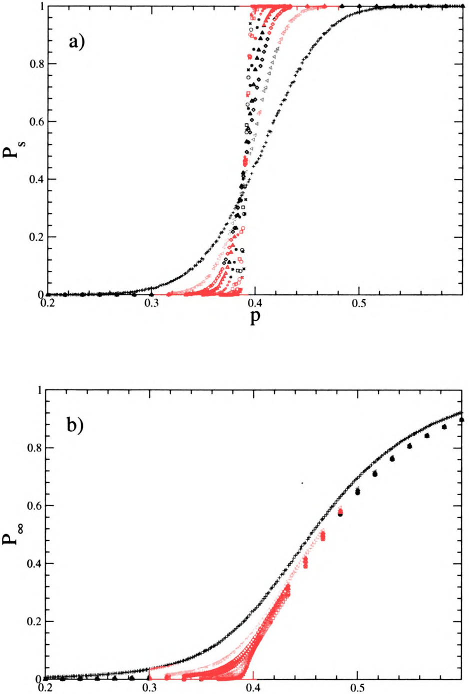


Figure 5.9: Plots of leaf removal on bond diluted simple cubic lattices. a) Spanning probability P_S as a function of the bond concentration p. b) The infinite cluster, P_∞, probability as a function of p. Both graphs over sizes $N = L^3$(and number of configurations) of $L = 5(10000)$, $L = 10(10000)$, $L = 15(10000)$, $L = 20(10000)$,$L = 30(10000)$, $L = 50(1000)$, $L = 70(1000)$, $L = 100(100)$.

5.2.1 Graph generation

All lattices were generated with free boundary conditions. The two-dimensional lattices were generated on an $L \times L$ substrate, where $L = \sqrt{N}$. If $\sqrt{N}$ was not an integer then L was rounded up (ie. $\sqrt{1000} = 31.6 \Rightarrow L = 32$). The final row in the sample is missing the number of nodes necessary to give the desired N. Thus an N that is a natural square such as $10,000$ is an 100×100 lattice, while a 1000 node lattice is 32×32 with 24 nodes missing in the final row. There was no difference in the results calculated on lattices constructed in this manner and lattices constructed from perfect squares. The FCC lattices and the cubic lattices were constructed in an $L \times L \times L$ box, such that $N = L^3$.

On all lattices edges were placed by the random insertion of bonds, each bond has a probability of,

$$p = \frac{cN}{2E_{max}} \tag{5.13}$$

of being present. E_{max} is the total number of edges that would be present in the lattice if there were no dilution, this number is dependent on the geometry of the lattice. Each bond is assigned a random probability, which if less than p, the edge is inserted into the graph. The FCC lattice is generated along the 1,1,1 direction by stacking triangular lattices. The average connectivity of a site for the lattices is $c = zp$, where z is the coordination number, $z = 6$, for triangular and cubic lattices, $z = 4$ for the square lattice and $z = 12$ for the FCC lattice.

5.2.2 Leaf removal on 2-d lattices

In Fig 5.10, an unbiased estimate of the exponent ν is found by using, $\delta p_l^2 = \langle p_l^2 \rangle - \langle p_l \rangle^2 \sim L^{-2/\nu}$, where $\langle p_l \rangle$ is the average value of p_l found by $\langle p_l \rangle = \int p dp dP_S/dp$ and its second moment$\langle p_l^2 \rangle = \int p^2 dp dP_S/dp$. Fig 5.10 is a log-log plot of δp_l vs L from which we find that $\nu = 1.34(3)$ for the triangular lattice, and $\nu = 1.39(3)$

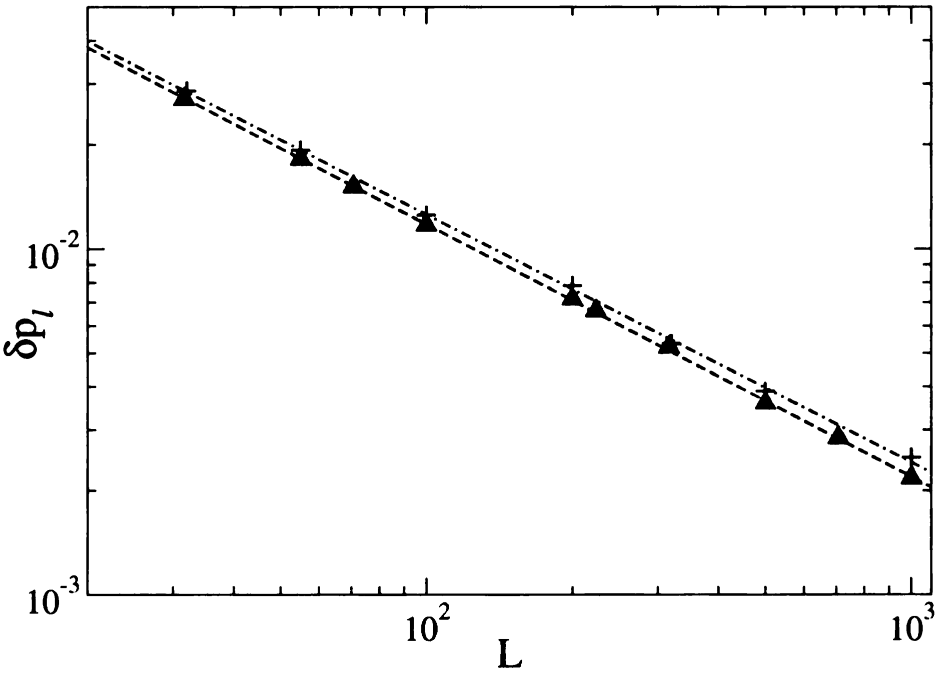


Figure 5.10: Plots of $\delta p_l^2 = \langle p^2 \rangle - \langle p \rangle^2$ as a function of lattice size L on a double logarithmic graph. The triangles are data from the triangular lattice, the dashed line is a best fit from which we extract the estimate $\nu = 1.37(3)$. The boxes are data from the square lattice from which we extract the estimate $\nu = 1.39(3)$.

for the square lattice. The values for ν are similar to each other, though the square lattice value is slightly higher than the conventional value for percolation. A summary of the exponents found can be seen in Table 5.3.

Determination of the critical threshold is presented in Fig 5.11. As the sample size increases, the value of $p_l(L)$ approaches the infinite lattice value and the width of the spanning probability δp_l approaches 0 [85] [32]. By plotting $p_l(L)$ as a function of $\delta p_l(L)$, we can make an estimate of $p_l(\infty)$. To determine a value for the intercept, the triangular lattice data was fit with a line, and the square lattice data was fit with a quadratic. From this fit we find the threshold for the triangular lattice at $p_l = 0.4690(4)$, and the threshold for the square lattice at $p_l = 0.6384(4)$.

In Fig 5.12, we analyze the finite size scaling behavior of the infinite cluster using higher-order finite size corrections. In correlated percolation problems, β can be sensitive to these corrections [66]. We now analyze the infinite cluster probabil-

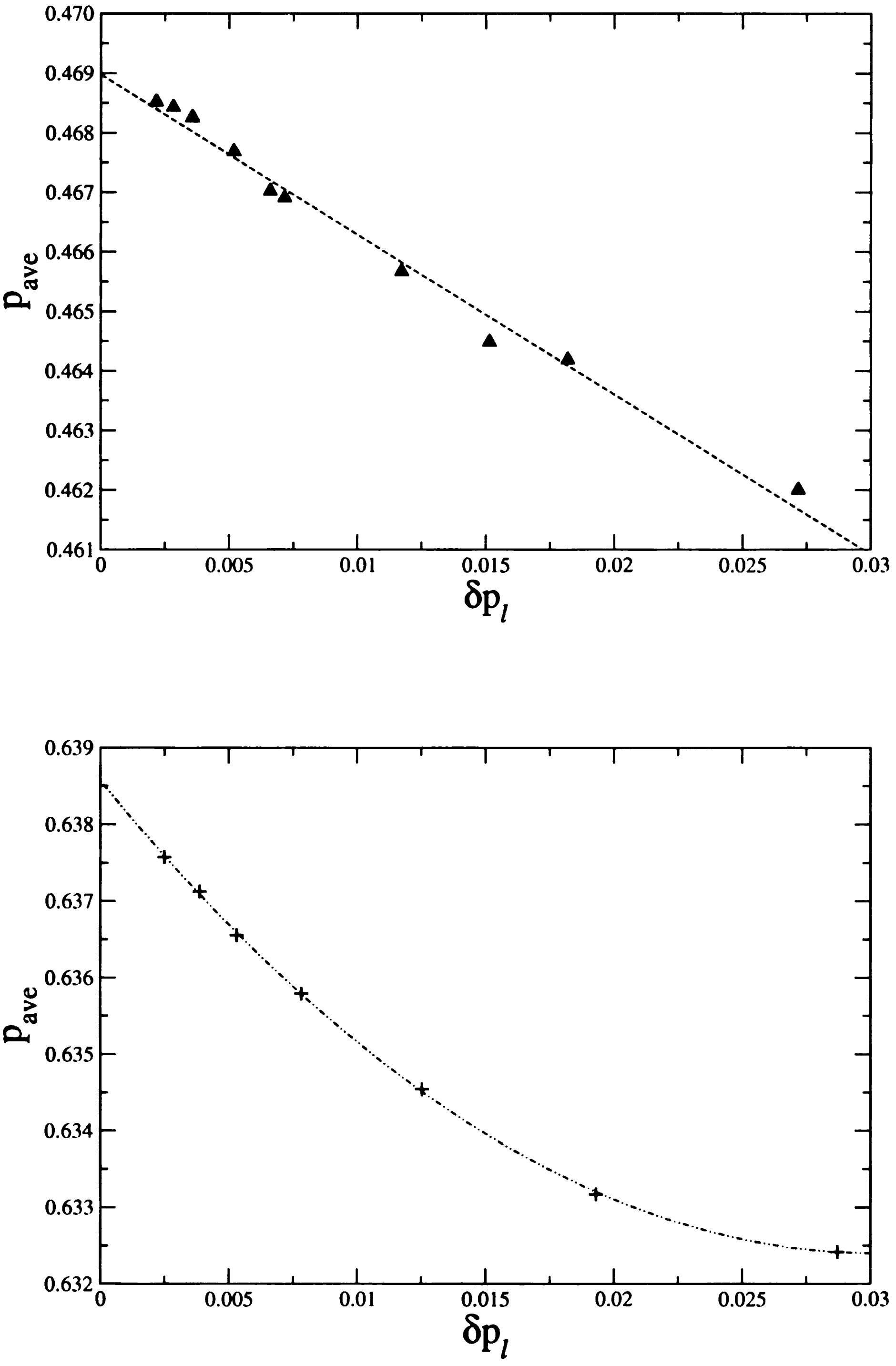


Figure 5.11: Plots of the average value of p_l as a function of δp_l, for each sample size we find $p_l(L)$ and $\delta p_l(L)$. δp_l goes to 0 as L goes to ∞. This allows us to extract the infinite lattice critical point. (a) Corresponds to the triangular lattice from which we extract $p_l = 0.4690(4)$. (b) Corresponds to the square lattice from which we extract $p_l = 0.6385(2)$.

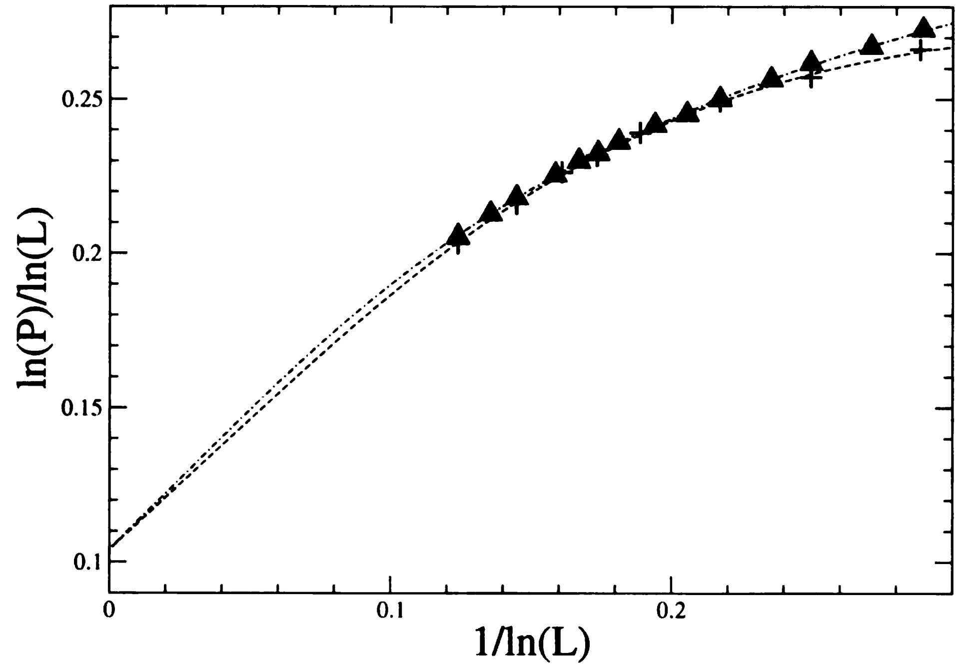


Figure 5.12: Analysis of the infinite cluster probability using a next to leading order finite size correction, as given by equation 5.14. The triangles are the data for the triangular lattice at $p_l = 0.4692$. The + are data for the square lattice at $p_l = 0.63825$, and in both cases a fit can easily be made for the standard percolation exponent $\beta/\nu = 0.104$. The triangular lattice has a correction to scaling exponent of $w = 0.35(5)$. The square lattice has a correction to scaling exponent of $w = 0.42(5)$. Sample sizes and number of realizations for the triangle lattice, $N = 1,000(20,000)$, $N = 1,600(20,000)$ $N = 3,000(20,000)$, $N = 4,900(20,000)$, $N = 10,000(20,000)$, $N = 16,900(20,000)$, $N = 30,000(40,000)$, $N = 62,500(20,000)$, $N = 100,000(10,000)$, $N = 160,000(10,000)$, $N = 300,000(5,000)$, $N = 1,000,000(1,000)$, $N = 2,560,000(1,000)$, $N = 10,240,000(100)$. Sample sizes and number of realizations for the square lattice, $N = 1,024(10,000)$, $N = 3,024(10,000)$, $N = 10,000(10,000)$, $N = 40,000(10,000)$, $N = 102,400(10,000)$, $N = 250,000(1,000)$, $N = 1,000,000(1,000)$, $N = 10,240,000(100)$.

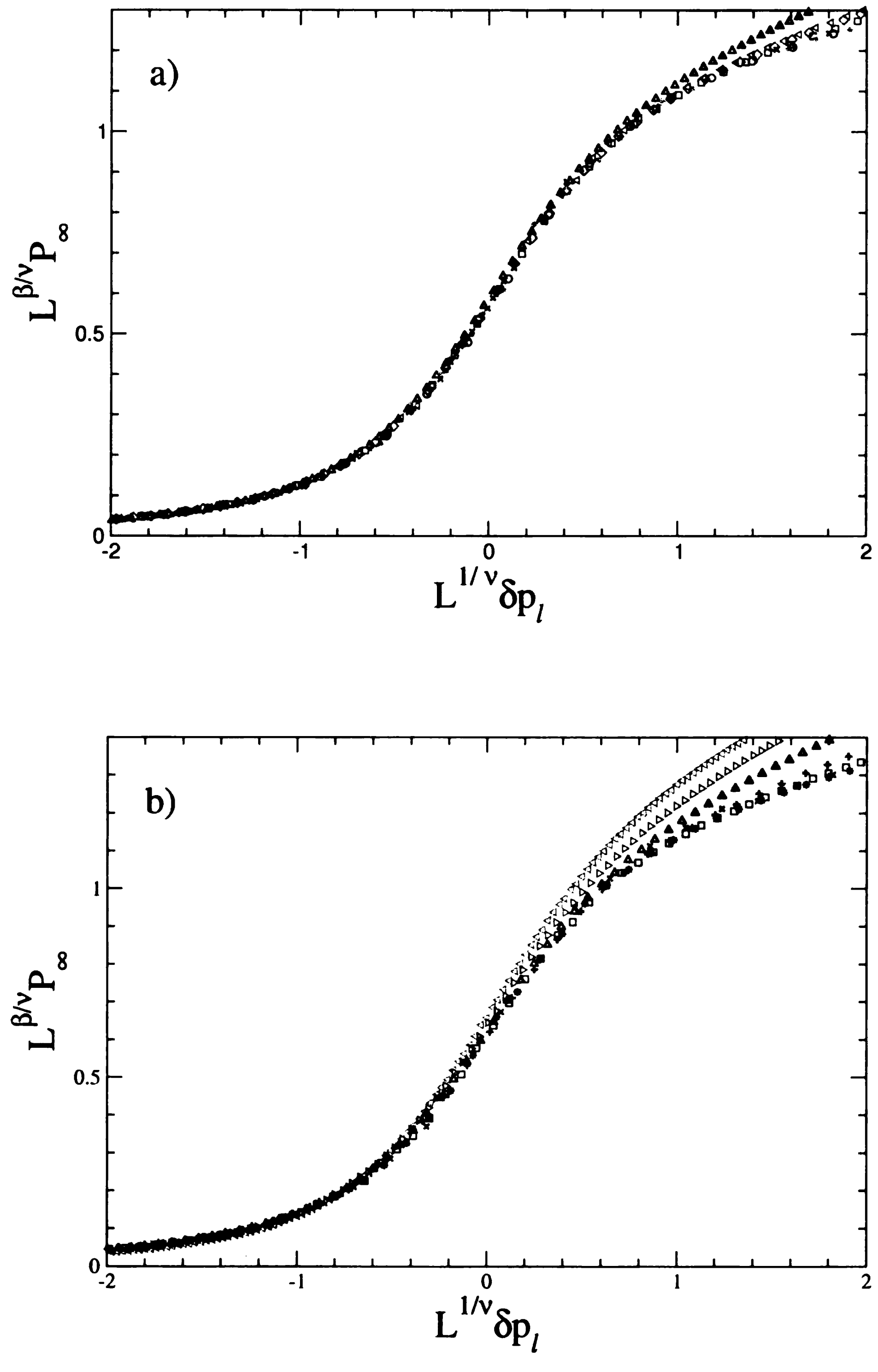


Figure 5.13: A scaling plot for the infinite cluster probability. (a) The best collapse for the triangular lattice was found with $p_l = 0.4692$, $\beta/\nu = 0.163$, $\nu = 1.35$. (b) The best collapse for the square lattice is found with $p_l = 0.3825$, $\beta/\nu = 0.146$, $\nu = 1.37$.

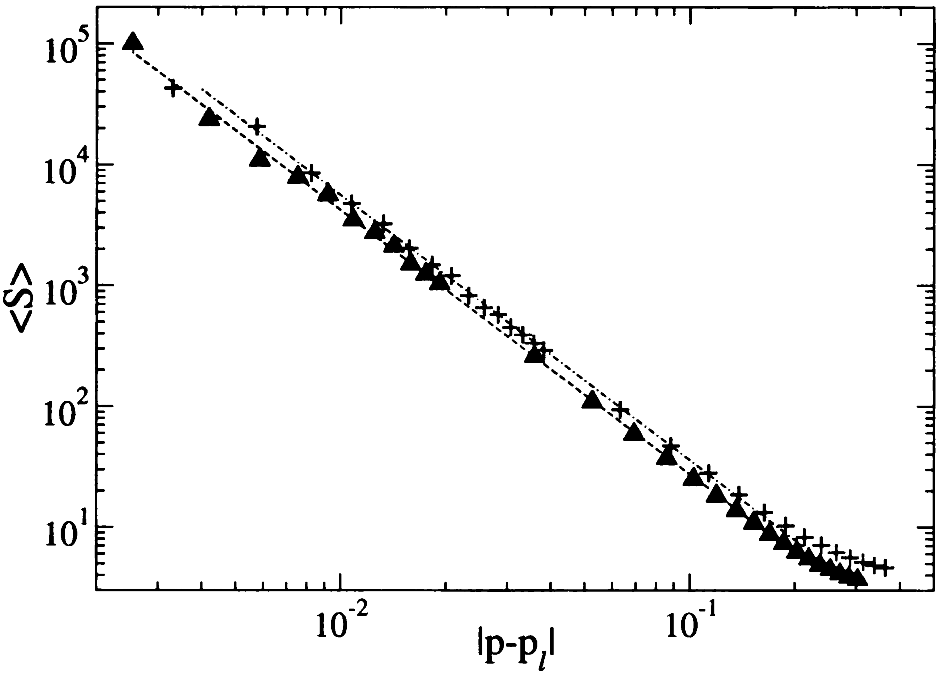


Figure 5.14: Plots of the average cluster size on approach to p_l from below for lattices of size $N = 10240000$ sites, from this plot we extract the exponent γ. The triangles correspond to the triangular lattice at $p_l = 0.4692$ yielding $\gamma = 2.16(3)$. The circles correspond tot he square lattice at $p_l = 0.63825$ yielding $\gamma = 2.19(3)$.

ity using the form [18],

$$P_\infty = aL^{-\beta/\nu}(1 + bL^{-w}). \tag{5.14}$$

A fit of the data is then made using $-\ln P_\infty / \ln L = \beta/\nu - \ln a / \ln L - bL^{-w} / \ln L$. It is difficult however to find the "best fit" for our samples, as any number of y-intercepts yielded reasonable fits. We proceeded on the assumption that core percolation was in the same universality class as conventional percolation. With that assumption, both the square and the triangular lattice can be fit with the standard percolation exponents $\beta/\nu = 0.104$, the correction to scaling exponents are then, $w = 0.35(5)$ for the triangular lattice, and $w = 0.42(5)$ for the square lattice. However, both lattices can be easily fit for a range of β/ν, implying that while this is consistent with the universality class of standard percolation, it is not sufficient to rule out a new universality class. In order to get more exact results, it

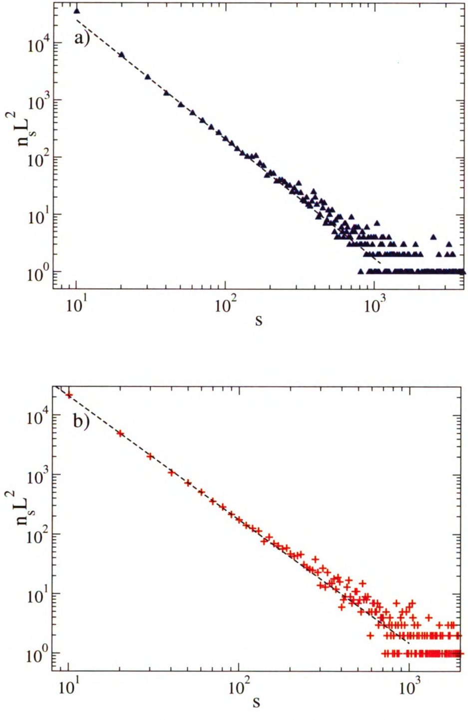


Figure 5.15: Double logarithmic plots of the cluster size distribution for lattices of $N = 10240000$ sites. (a) For a triangular lattice the dashed line has a slope of $-2.09(5)$. (b) For a square lattice the dashed line has a slope of $-2.07(2)$.

Table 5.3: Comparison of Core percolation exponents and Standard Percolation exponents for 2-d lattices. * values measured without the use of corrections to scaling.

exp.	d=2	triangular	square	comments
ν	$4/3 \sim 1.33$	1.37(2)	1.39(1)	Figure 5.10
ν	$4/3$	1.35(4)	1.37(4)	Figure 5.13
β	$5/36 \sim 0.139$	*0.20(2)	*0.20(2)	Figure 5.13
γ	$43/18 \sim 2.39$	*2.16(3)	*2.19(3)	Figure 5.14
τ	$187/91 \sim 2.05$	2.09(5)	2.07(2)	Figure 5.15

will be necessary to run larger sample sizes, but that becomes difficult as sizes of $N = 10,240,000$ with 100 realizations takes several days to run on our computers, and getting results for sizes even an order of magnitude larger becomes prohibitively difficult.

The scaling behavior of P_∞ may also be determined by the collapse of data in Fig 5.13. A very good collapse for the triangular lattice of sizes $N \geq 40000$ ($L \geq 200$) sites, may be found with $p_c = 0.4692(2)$, $\beta = 0.20(2)$, and $\nu = 1.35(4)$, And for the square lattices over the same range of sizes for $p_c = 0.3825(2)$, $\beta = 0.20(2)$, and $\nu = 1.37(4)$. The β found from the finite size collapse is higher than expected, and is explained by the difficultly in determining β in correlated percolation problems.

The cluster statistics are presented in Figs 5.14 and 5.15. The value of γ we find is smaller than the conventional percolation value of $\gamma = 2.16(3)$ for the triangular lattice and, $\gamma = 2.19(3)$ for the square lattice (see table 5.3). To find τ, one sample of $L = 3200(N = 10,240,000)$ was run as near to p_l was possible. The quantity n_s is defined as,

$$n_s = \frac{\text{number of clusters of size } s}{N}. \tag{5.15}$$

And the average cluster size $\langle S \rangle$ is defined as,

$$\langle S \rangle = \sum \frac{n_s s^2}{\sum n_s s} \tag{5.16}$$

From Fig 5.15 we find that $\tau = 2.09(5)$ for the triangular lattice and $\tau = 2.07(2)$ for the square lattice, which are consistent with the conventional value.

Core percolation on the 2-dimensional lattice is shown to be consistent with the universality class of standard percolation. The largest deviation from standard percolation is in the value found for γ (see Figure 5.14), which is significantly lower than the standard value. β is also a difficult prospect but the higher-order scaling corrections show that our results are consistent with the conventional value, similar finite size corrections occur in standard percolation (see appendix A).

5.2.3 Leaf removal on 3-d lattices

Leaf removal on 3-dimensional lattices shows behaviour similar to the 2-d lattices. On the 3-d lattices most exponents are close to the standard percolation values, however, ν for both lattices and β for the cubic lattice show deviations from the standard values. Fig 5.16 gives an estimate of $\nu = 1.00(5)$ for the simple cubic lattice and $\nu = 0.982(5)$ for the FCC lattice. These values are higher than the conventional 3-d value of 0.88. Appendix A presents the bond percolation results for the FCC and cubic lattices with free boundary conditions. The bond percolation results give $\nu = 0.93(4)$ for the cubic lattice and $\nu = 0.92(2)$ for the FCC lattice (see table A.2). A comparison of exponents can be seen in table 5.5. The critical threshold for the FCC and cubic lattices is shown in Fig 5.17, giving a value of $p_l = 0.202(2)$ for the FCC lattice and $p_l = 0.393(1)$ for the simple cubic after quadratic fits.

The higher-order finite size scaling plots of the FCC lattice and the cubic lattice appear in Fig 5.18, again with a fit constrained to intercept at the conventional value of $\beta/\nu = 0.4659$, they both show fits that appear reasonable, with the FCC lattice needing no corrections to scaling. From a linear fit we find $\beta/\nu = 0.471(6)$ for the FCC lattice. The simple cubic lattice has a correction to scaling exponent of

$w = 0.4(3)$.

The scaling plots for the collapse of the infinite cluster probability appear in Fig 5.19. For both lattice types we were able to find a reasonable collapse. Fig 5.19(a) shows the collapse of the FCC lattice at $p_l = 0.2023$, and the conventional percolation values of $\beta = 0.41(2)$, $\nu = 0.88(4)$ this collapse is for the 5 largest samples $L \geq 20$. Fig 5.19(b) shows the collapse for the same sample sizes of the cubic lattice at $p_l = 0.3921$. In this case, we see that the value for $\beta = 0.61(2)$ is higher than the standard value, while $\nu = 0.88(4)$ is equal to the conventional value.

When the results obtained here for the leaf removal and the results for standard bond percolation are compared, on 3-d lattices with free boundary conditions, similar values are obtained implying that the deviation seen in ν is related to finite size effects and the boundary conditions.

The cluster statistics are presented in Figs 5.20 and 5.21. In the 3-d case, the value of γ we find is much closer to the conventional percolation value of $\gamma = 1.80$, than we saw in the 2-d cases. We find $\gamma = 1.74(2)$ for the FCC lattice and, $\gamma = 1.7(2)$ for the cubic lattice. Likewise, we find that $\tau = 2.17(8)$ for the FCC lattice and $\tau = 2.23(6)$ for the cubic lattice, which are consistent with the conventional values. A summary of the exponents on the 3 dimensional lattices can be found in table 5.5.

Table 5.4: Comparison of core percolation exponents and standard percolation exponents for 3-d lattices

exp.	d=3	FCC	cubic	comments
ν	0.88	0.982(5)	1.00(5)	Figure 5.16
ν	0.88	0.88(4)	0.88(4)	Figure 5.19
β	0.41	0.41(2)	0.61(2)	Figure 5.19
γ	1.8	1.74(2)	1.7(2)	Figure 5.20
τ	2.18	2.17(8)	2.23(6)	Figure 5.21

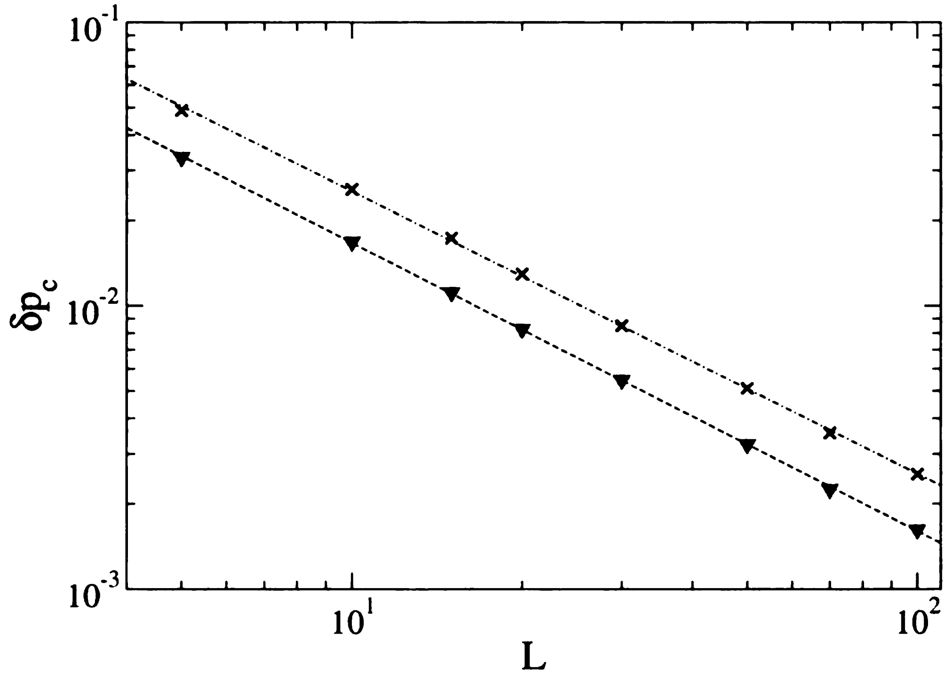


Figure 5.16: Plots of $\delta p_l^2 = \langle p^2 \rangle - \langle p \rangle^2$ as a function of lattice size L on a double logarithmic graph. The down facing triangles are the data from the FCC lattice with give an estimate of $\nu = 0.982(5)$. The cubic lattice data is the crosses which give an estimate of $\nu = 1.00(5)$.

5.3 Summary of core percolation by leaf removal

On all lattices studied core percolation by leaf removal appears to be in the same universality class as standard percolation. The exponent β was difficult to obtain because of finite size effects, but higher-order scaling corrections showed it to be consistent with the standard percolation value. The value for ν found is close to standard percolation. On the 2-d lattices ν was measured as slightly higher than the standard value. On the 3-d lattices the value of ν measured in Figure 5.16 was significantly higher than the standard percolation value. However, the value of ν extracted from the finite size collapse was equal to the standard percolation value. The discrepancy is believed to be a product of edge effects from the free boundary conditions and finite size effects, as shown by similar results in Appendix A. The exponent γ as calculated for the triangular and square lattices was low, but it

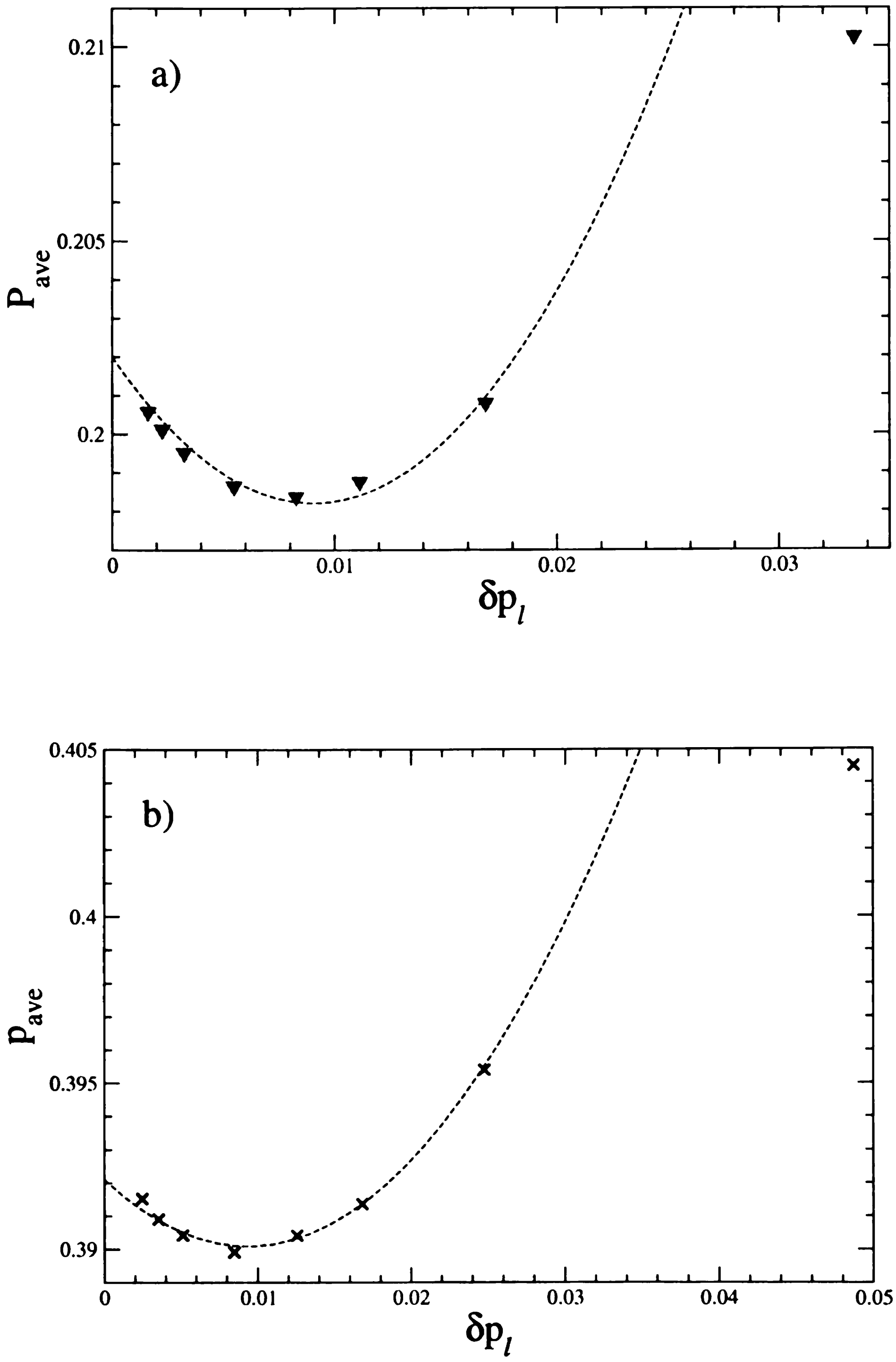


Figure 5.17: Plots of the average value of p_l as a function of δp_l, for each sample size we find $p_l(L)$ and $\delta p_l(L)$. (a) Corresponds to the FCC lattice from which we extract $p_l = 0.202(1)$. (b) Corresponds to the cubic lattice from which we extract $p_l = 0.3925(5)$.

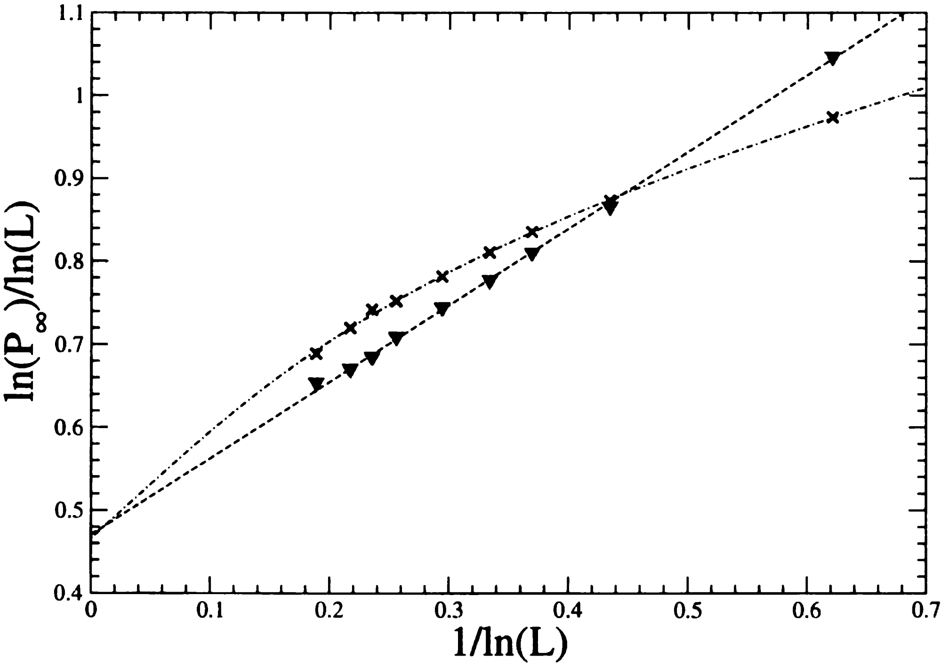


Figure 5.18: Analysis of the infinite cluster probability using a next to leading order finite size correction, as given by equation(5.14). The triangles are the data for the FCC lattice at $p_l = 0.2023$. The crosses are data for the cubic lattice at $p_l = 0.3928$, and in both cases a fit can easily be made for the standard percolation exponent $\beta/\nu = 0.4659$. A linear fit for the FCC lattice gives $\beta/\nu = 0.471(6)$. The correction to scaling exponent for the simple cubic lattice is $w = 0.4(3)$. Sample sizes and number of realizations. $L = 5(10,000)$, $L = 10(10,000)$, $L = 15(10,000)$, $L = 20(10,000)$,$L = 30(10,000)$, $L = 50(10,000)$, $L = 70(10,000)$, $L = 10(1,000)$,$L = 200(100)$.

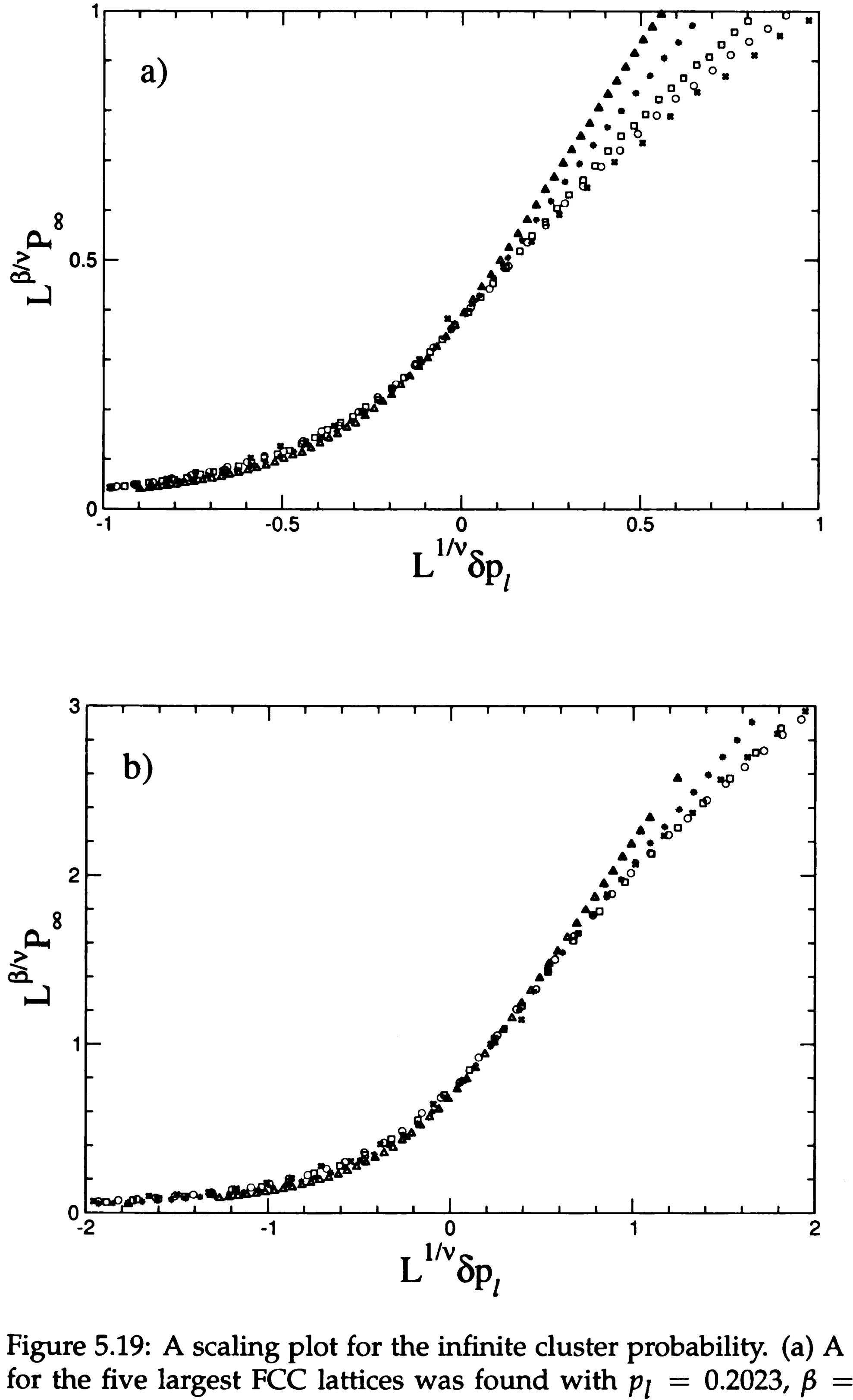


Figure 5.19: A scaling plot for the infinite cluster probability. (a) A good collapse for the five largest FCC lattices was found with $p_l = 0.2023$, $\beta = 0.41(2)$, $\nu = 0.88(4)$. (b) A good collapse for the five largest cubic lattices is found with $p_l = 0.3921$, $\beta = 0.61(2)$, $\nu = 0.88(4)$.

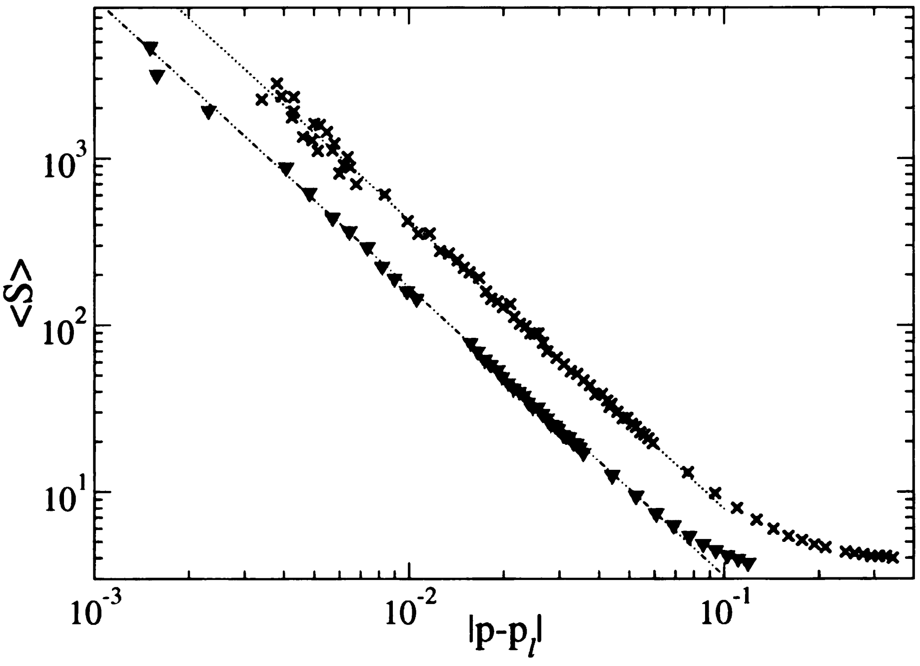


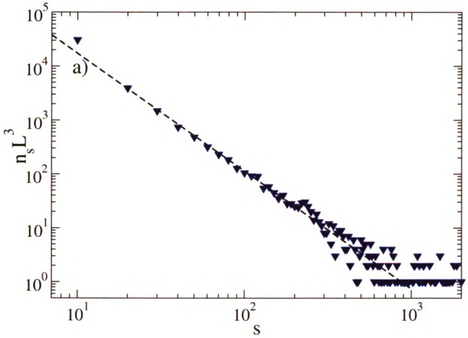
Figure 5.20: Plots of the average cluster size on approach to p_l from below for lattices of size $N = 8 \times 10^6$ sites. Plots for the FCC lattice ($\bigtriangledown$) yield $\gamma = 1.74(2)$. (b) Plots for the square lattice ($\times$) yield $\gamma = 1.7(2)$.

was very close for the FCC and cubic lattices. τ was found to be consistent with standard percolation values on all graphs.

5.4 Core percolation by triangle removal

Core percolation by triangle removal appears to be in the same universality class as standard percolation. Figure 5.22 and Figure 5.23, show a typical second order phase transition in P_∞. The transition on the triangular lattice occurs at $p_t = 0.580(2)$ and at $p_t = 0.228(2)$ on the FCC lattice (see Table 5.2). Figure 5.25 shows the approach to $p_t(\infty)$ as a function of $(\delta p_t(L))^2 = \langle p_t(L)\rangle^2 - \langle p_t(L)^2\rangle$, from which an estimate of the critical point, p_t is extracted.

Critical exponents were determined using the same procedure as for bond percolation and core percolation by leaf removal. Figure 5.24 shows the plot to extract the exponent ν which gives a value of $\nu = 1.33(6)$ for the triangular lattice, which



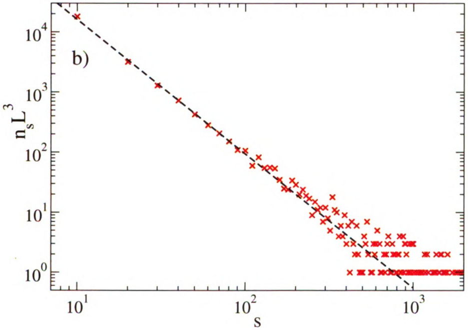


Figure 5.21: Double logarithmic plots of the cluster size distribution for lattices of $N = 8 \times 10^6$ sites. (a) For a FCC lattice we find a $\tau = 2.20(7)$ at $p_c. = 0.2023$ (b) τ on a simple cubic lattice is 2.23(6) at $p_c = 0.3928$.

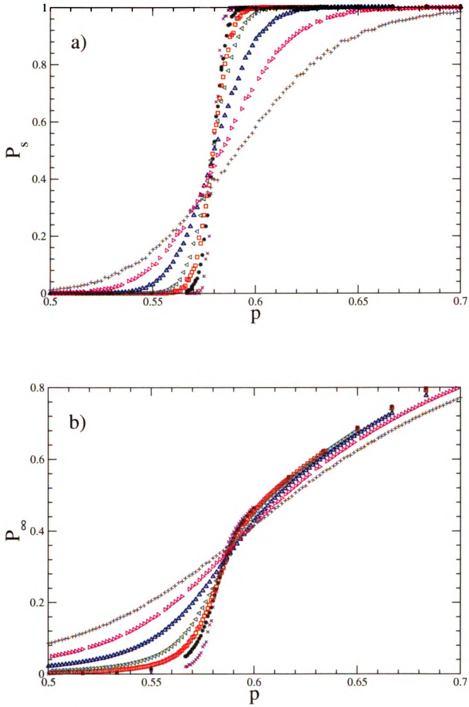


Figure 5.22: Plots of the core after triangle removal on a bond diluted triangular lattice. a) P_S as a function of bond concentration. b) P_∞ vs connectivity. Sample sizes and number of configurations, $N = 1,000,000(100)$, $N = 250,000(1,000)$, $N = 102,400(1,000)$, $N = 40,000(10,00)$, $N = 10,000(10,000)$, $N = 3,025(10,000)$, $N = 1,024(10,000)$.

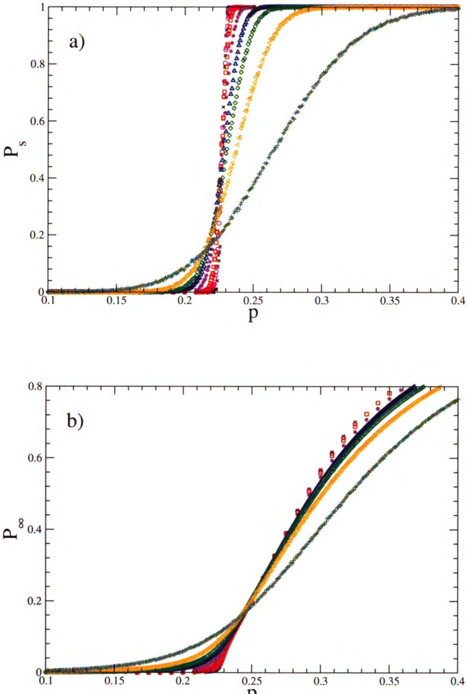


Figure 5.23: Plots of the core after triangle removal on a bond diluted triangular lattice. a) P_S as a function of bond concentration. b) P_∞ vs connectivity. Sample sizes and number of configurations, $L = 5(10,000)$, $L = 10(10,000)$, $L = 15(10,000)$, $L = 20(10,00)$, $L = 30(10,000)$, $L = 50(1,000)$, $N = 70(1,000)$, $N = 100(100)$.

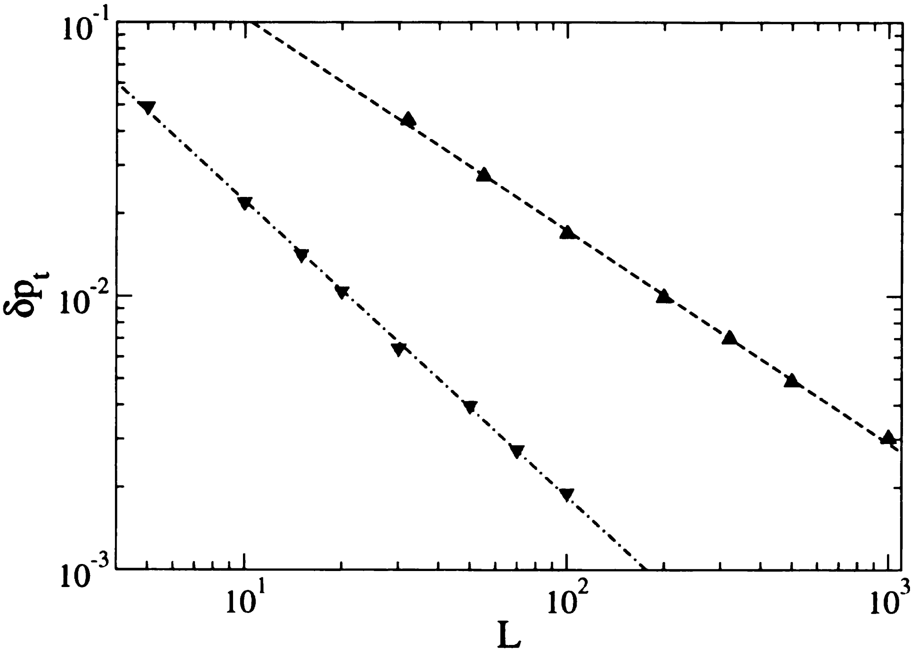


Figure 5.24: log-log plots of δp_t as function of L on after triangle removal on the triangular and FCC lattices. A value of $\nu = 1.33(6)$ was measured for the triangular lattice, and $\nu = 0.92(2)$ was measured for the FCC lattice.

matches exactly the standard value for ν. On the FCC lattice we got a value of $\nu = 0.92(2)$, which is higher than the standard value of $\nu = 0.88$, but equal to the value measured for standard percolation in Appendix A.

Figure 5.26 shows the correction to scaling analysis of the infinite cluster. The triangular lattice may be fit with values consistent with those of standard percolation, as with leaf removal there is a great deal of freedom in the fit. Lattices up to the size of $N = 10^6$ nodes were run, which is an order of magnitude smaller than those run for leaf removal (for leaf removal $N \approx 10^7$). This is due to the time required for triangle removal on lattices near $N \approx 10^7$, with one realization taking several days.

Finite size scale collapses are shown in Figure 5.27, which show that for the FCC and triangular lattices, the value of ν obtained for the best collapse is equivalent to the standard percolation value. As in the collapse for core percolation by leaf removal, it is observed that the value for β is higher than the standard value. The

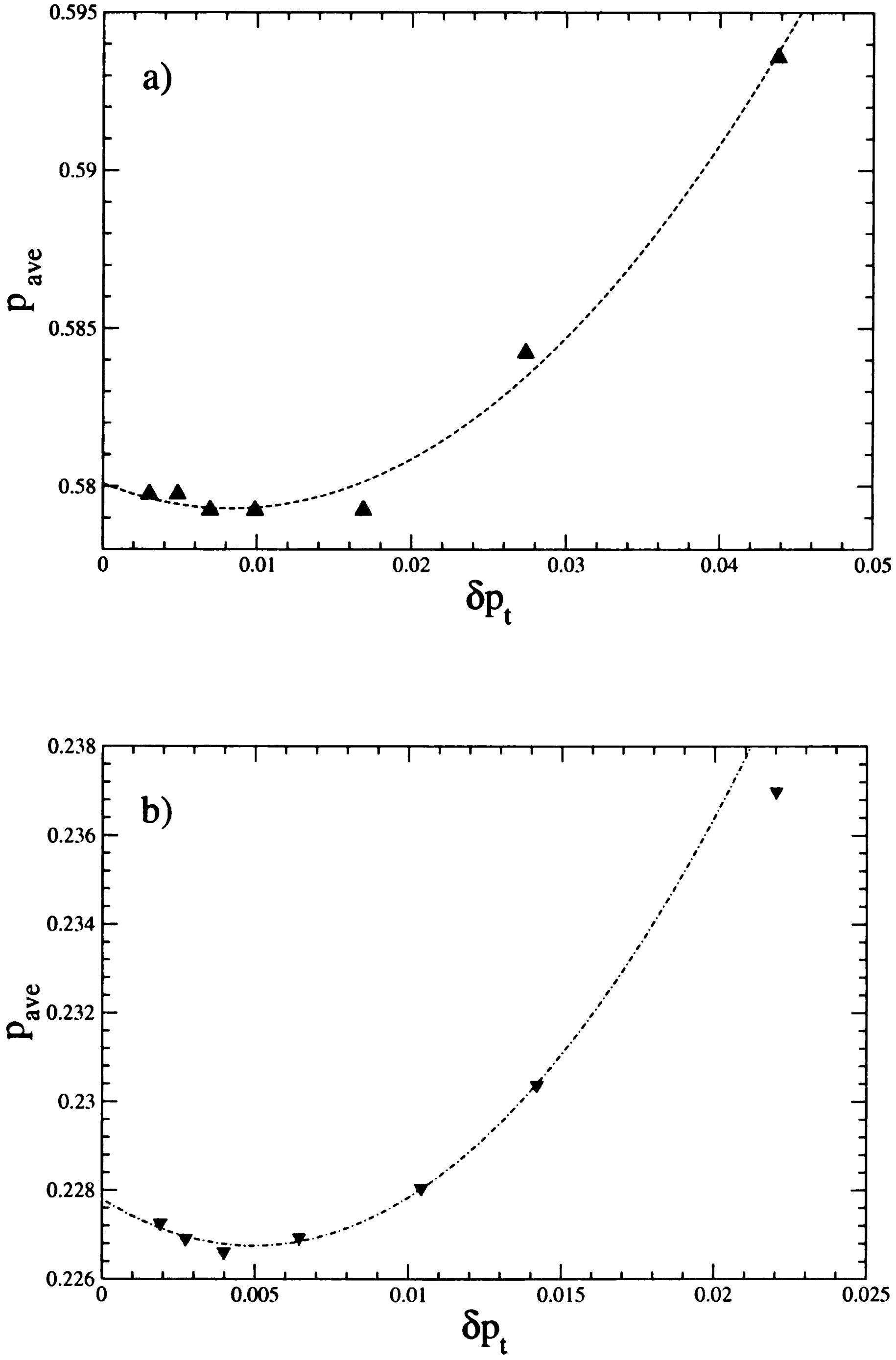


Figure 5.25: Plots of p_t as a function of δp_t after triangle removal on the triangular and FCC lattices From (a) we extract a value $p_t = 0.580(2)$ for the triangular lattice and on the FCC lattice (b) a value of $p_t = 0.228(2)$ is obtained.

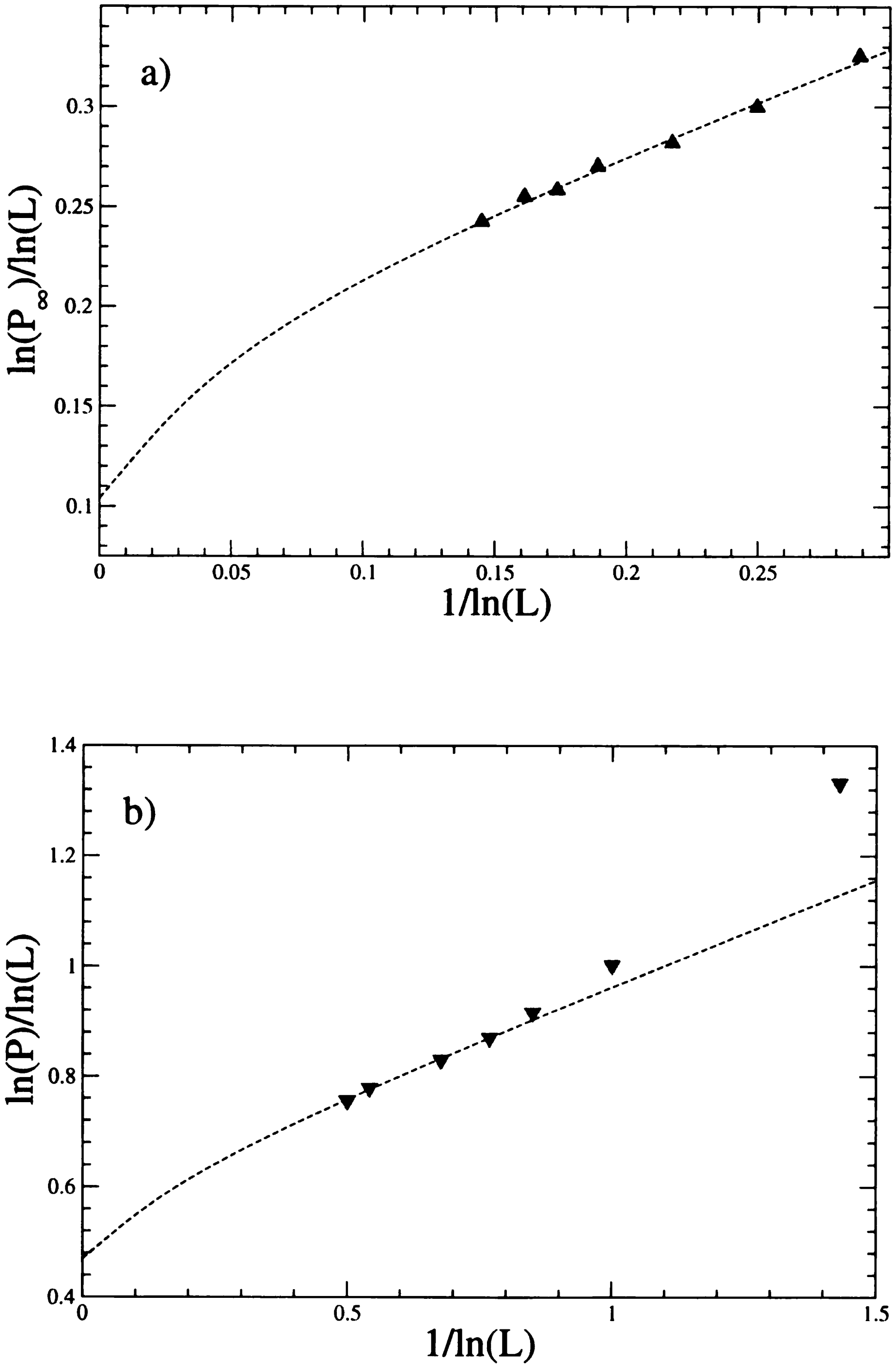


Figure 5.26: Higher order finite size scaling after triangle removal. a) Results for the triangular lattice. b) Results of the FCC lattice.

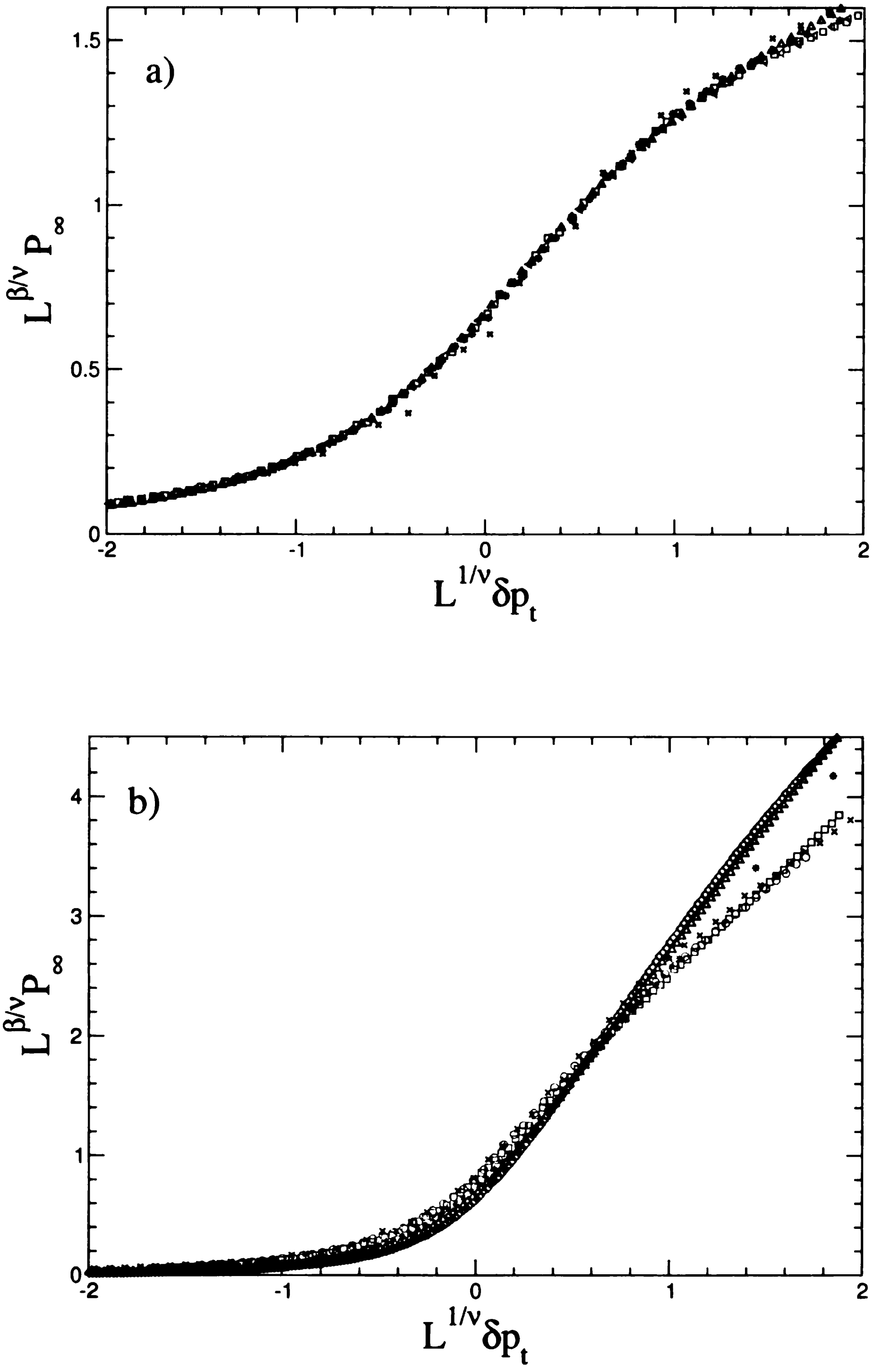


Figure 5.27: Finite size collapse of P_∞ on the FCC and the triangular lattice after triangle removal. (a) The collapse for the triangular lattice was obtained for values of $p_t = 0.579, \beta = 0.27, \nu = 1.33$. (b) The collapse for the FCC lattice was obtained for values of $p_t = 0.228, \beta = 0.61, \nu = 0.88$.

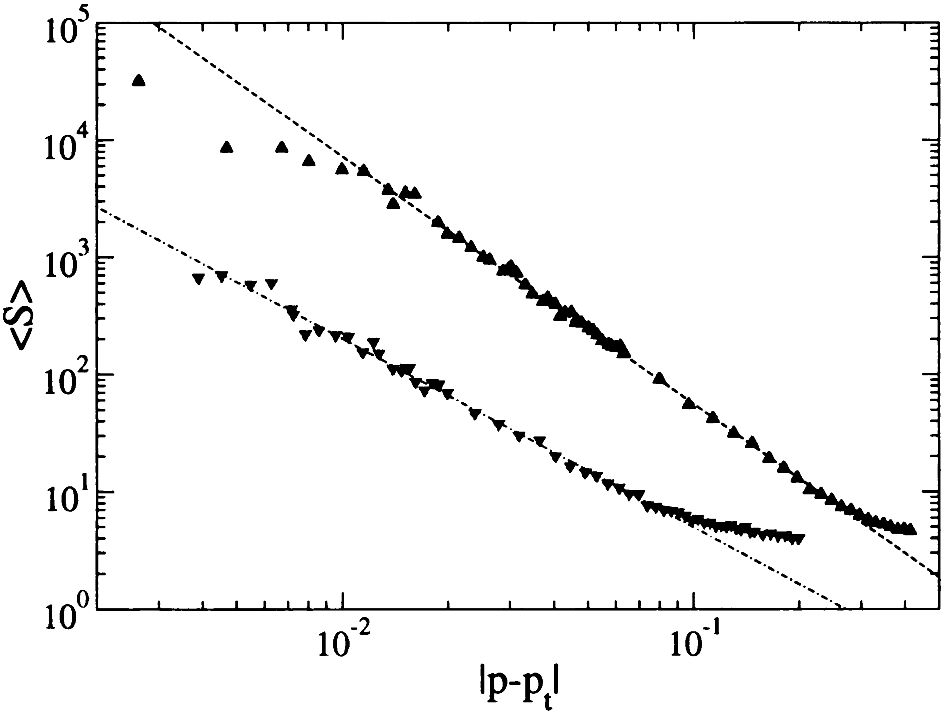


Figure 5.28: The Average cluster size on approach to p_t after triangle removal. For both lattices each point represents one realization of lattices of size $N = 10^6$ nodes. (a) Shows the triangular lattice with $p_t = 0.580$ giving a value of $\gamma = 2.13(4)$. (b) Shows the FCC lattice with $p_t = 0.228$ giving a value of $\gamma = 1.61(3)$.

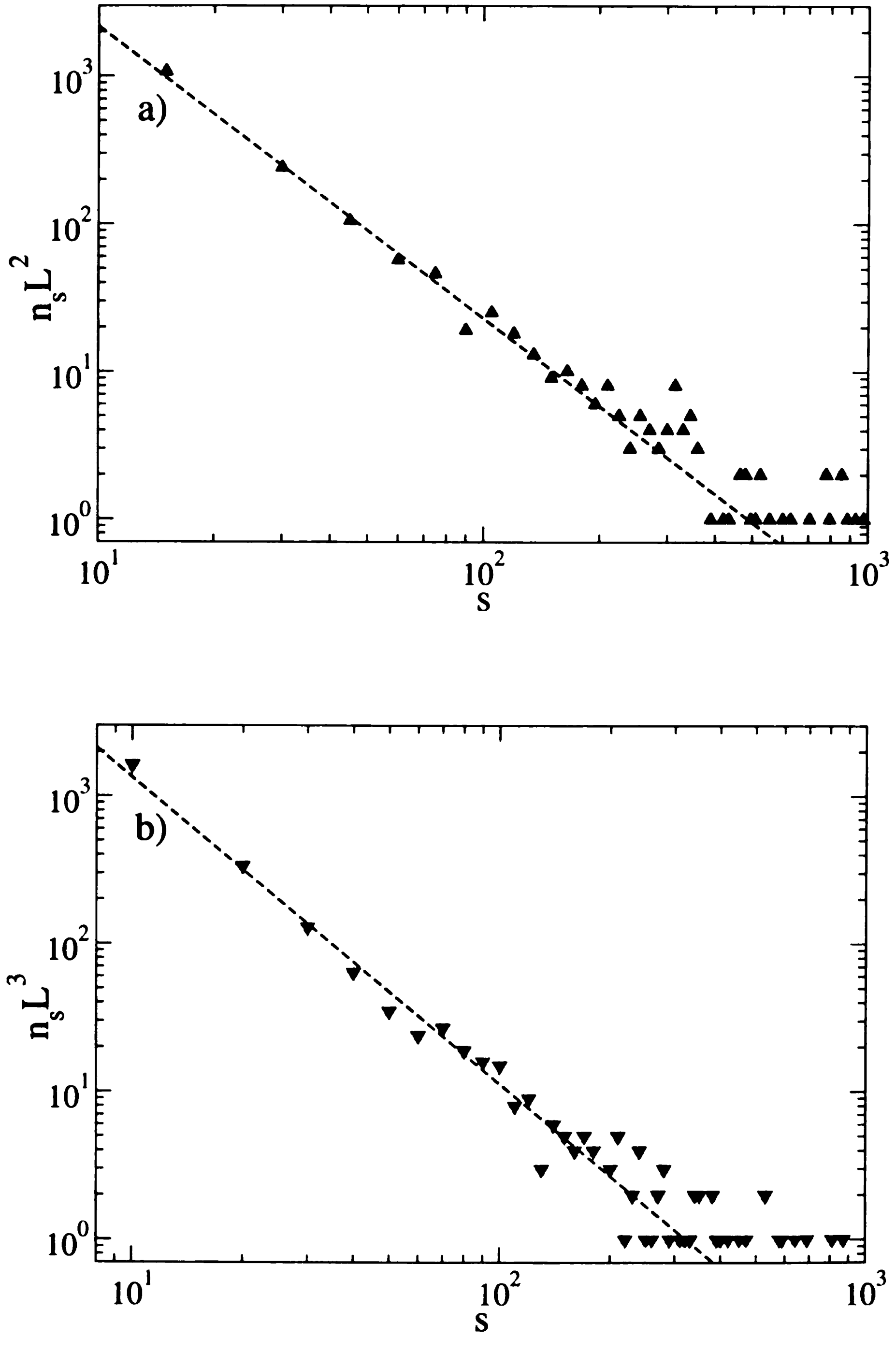


Figure 5.29: Cluster size distribution for the triangular lattice after triangle removal. (a) Gives the value of $\tau = 1.98(8)$ for the triangular lattice of size $N = 10^6$ at $p_t = 0.580$. (b) Shows the cluster size distribution for an FCC lattice of size $N = 10^6$ at $p_t = 0.230$ giving a value of $\tau = 2.07(8)$.

values found from triangle removal are higher than those found by leaf removal. The triangular lattice sees an increase in β of roughly 40% and increase of 50% on the FCC lattice, over the leaf removal values. The value of β for triangle removal on the FCC lattice compares favorably to the values of β on the cubic lattice for leaf removal, this of course is only a valid comparison if as for standard percolation, only the dimensionality of the lattice matters in the universality of the critical exponents. As stated previously it is difficult to obtain β in correlated percolation problems.

The value of γ is obtained from Figure 5.28. On the triangle lattice a value of $\gamma = 2.13(4)$ is observed, which, while lower than the standard percolation value, compares favorably to the value of γ for the triangular lattice from leaf removal. The value of $\gamma = 1.61(3)$ on the FCC lattice which is slightly lower than the values obtained for leaf removal and for standard percolation. This exponent is sensitive to the critical point, p_t, used in its calculation, which may account for the discrepancy. Finally, τ obtained for the triangular lattice and FCC is consistent with the values for bond percolation and leaf removal. On the triangular lattice it was necessary to increase the box size of the histogram, in order to reduce the noise enough to generate a fit.

Table 5.5: Comparison of Core percolation exponents and Standard Percolation exponents after triangle removal

exp.	d=2	d=3	triangular	FCC	
ν	1.33	0.88	1.33(6)	0.92(2)	Fig 5.24
ν			1.33(4)	0.88(4)	Fig 5.27
β	0.139	0.41	0.27(2)	0.66 (2)	Fig 5.27
γ	2.39	1.8	2.13(4)	1.61(3)	Fig 5.28
τ	2.05	2.18	1.98(8)	2.07(8)	Fig 5.29

5.4.1 Triangle removal on a random graph

Triangle removal on a random graph resulted in an extremely small shift of the critical point. This is due to the very small number of triangles present in a random graph. Figs. 5.30-5.32 show the results for the giant cluster after leaf removal. From Fig 4.2 we can see that the difference between triangle removal on a random graph and leaf removal is minimal at best, as expected. In Figure 5.34, we see the collapse of triangle removal data is similar to the collapse obtained for the leaf removal data, Figure 5.5. Figure 5.33(a) shows the finite size effect of the connectivity just as we had in the leaf removal case, with similar fits. Figure 5.33(b) shows the deviation between the leaf removal and triangle removal at small sample sizes, indicating an increase in the finite size effects This supports our claim that as the dimensionality increases the difference between the transitions for leaf removal and triangle removal disappears.

5.4.2 Summary of triangle removal

Core percolation by triangle removal appears to be in the same universality class as core percolation by leaf removal and standard percolation. For all types of core percolation it is difficult to obtain values of β consistent with standard percolation, this is believed to be due to a lack of larger samples and the difficulty in determining β in correlated percolation problems. At lower dimensions, there is a sizeable difference between the leaf removal critical point and the triangle removal critical point, as the dimensionality of the graph increases the difference between the two goes to zero.

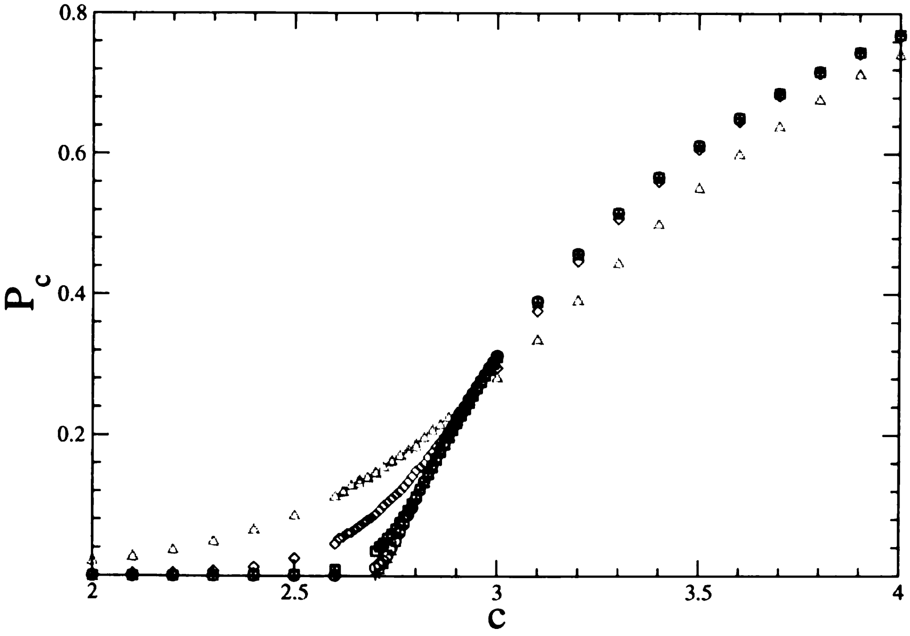


Figure 5.30: Plot of the probability of being on the core verses connectivity after triangle removal for random graphs. Samples sizes and number of realizations. N=100(10,000), N=1,000(10,000), N=10,000(10,000), N=100,000(1,000), N=1,000,000(100).

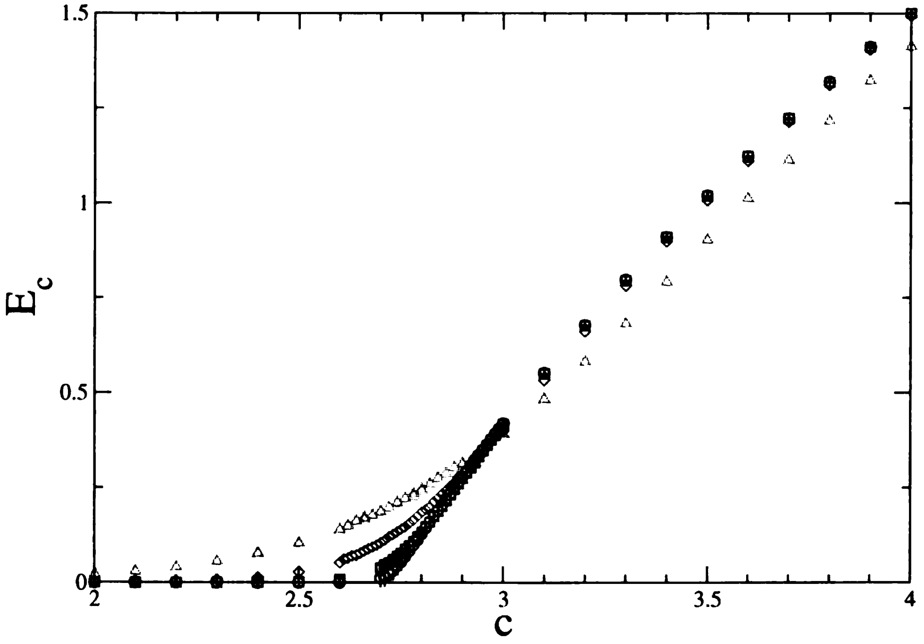


Figure 5.31: Plot of edges in the core cluster verses connectivity after triangle removal. Samples sizes (number of realizations), N=100(10,000), N=1,000(10,000), N=10,000(10,000), N=100,000(1,000), N=1,000,000(100).

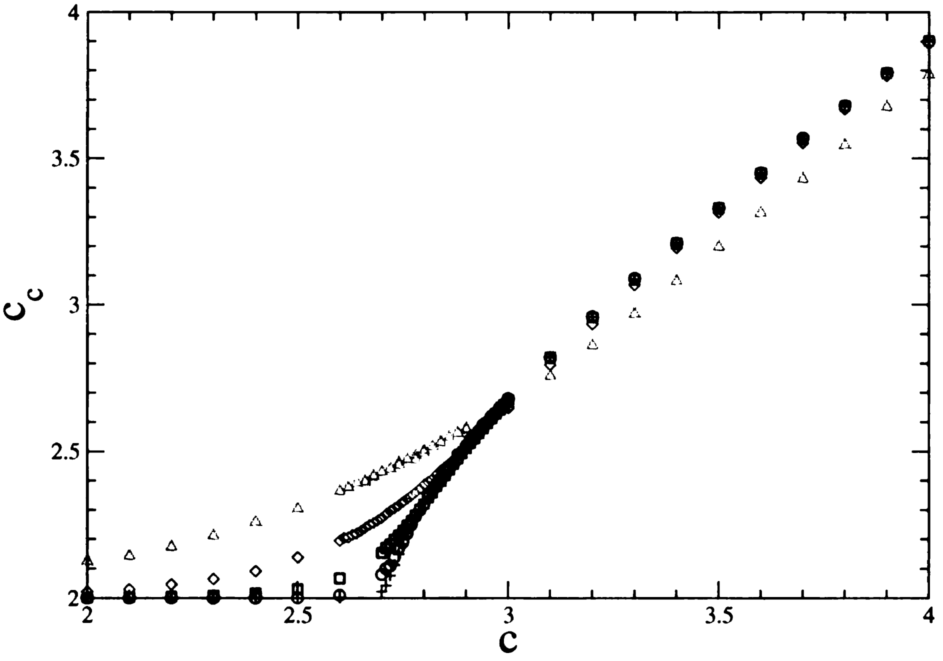
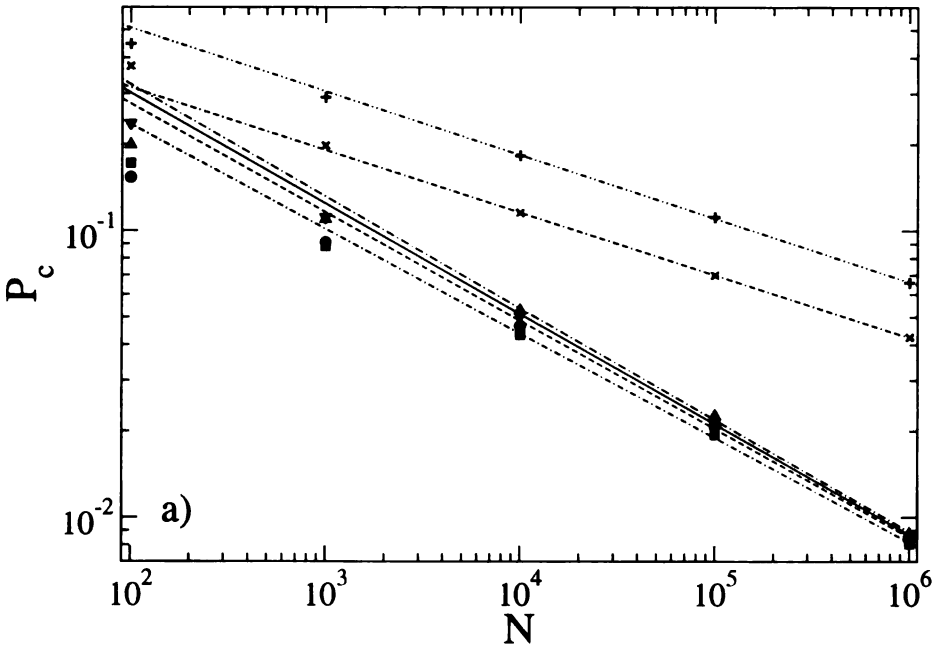


Figure 5.32: Plot of connectivity of core verses connectivity of original graph after triangle removal. Samples sizes (number of realizations), N=100(10,000), N=1,000(10,000), N=10,000(10,000), N=100,000(1,000), N=1,000,000(100).



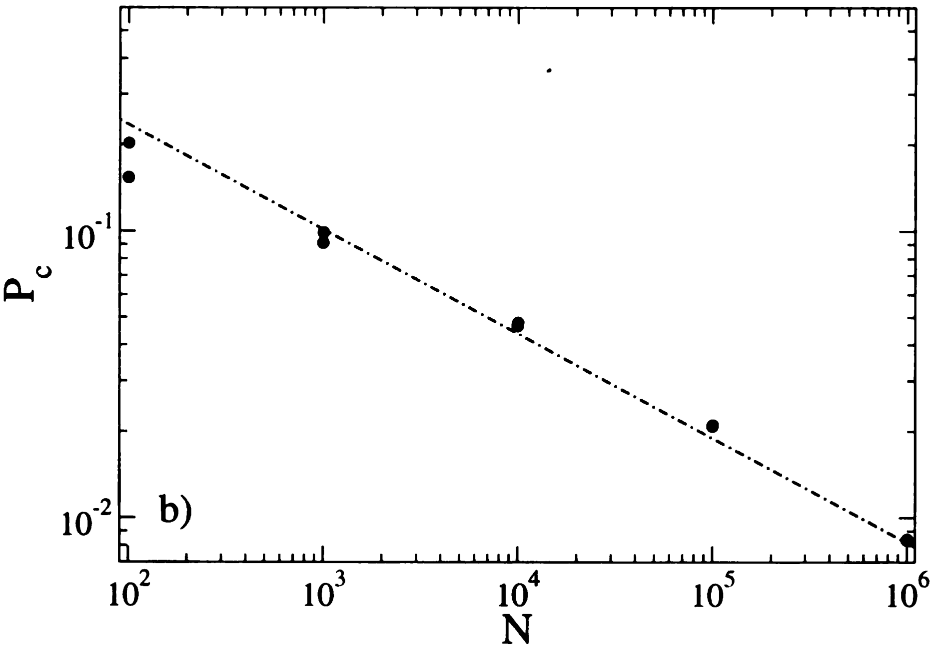


Figure 5.33: a) Finite size effects of connectivity (+), Edges in core (▼), P_C (●), the variance the connectivity (×), variance in the number of edges in the core (▼), variance of the probability of being on the core (■) at the transition, $c = e$. b) shows the finite size effects of the probability of being on the core after both triangle removal (black) and the leaf removal (green). Samples sizes (number of realizations), N=100(10,000), N=1,000(10,000), N=3,000(10,000), N=10,000(10,000), N=300,000(10,000), N=100,000(1,000), N=1,000,000(200).

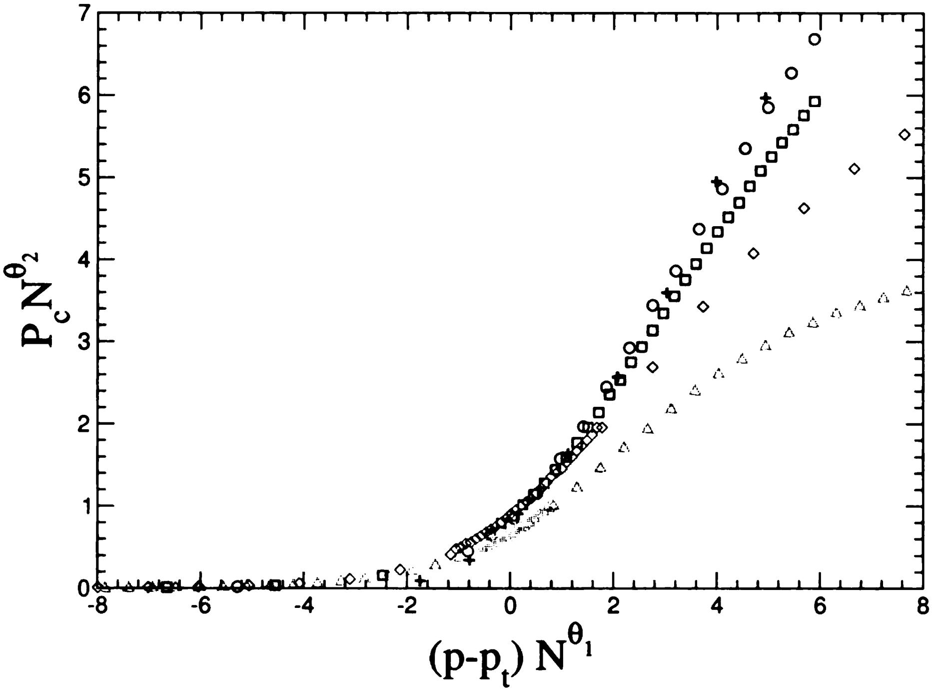


Figure 5.34: Finite size collapse of the core after triangle removal on the random graph, fit with $\theta_1 = 0.35(2)$, $\theta_2 = 0.33(3)$.

Chapter 6

Local Probability Recursion

Hard computational problems, like the hard-core lattice gas, display glassy behaviour. The aspects of the hard computational problems that are important physically include the onset of glassy behaviour, and degeneracy of ground states. The issue of degeneracy brings forth the ideas of fluctuations, the frozen fraction, and clusters. We know that in spin glasses and frustrated lattice gases there are a large number of ground states. The frozen fraction is defined by the overlap of the states, the nodes in the frozen fraction, are nodes that are in the same state (covered or uncovered) for all ground states. $\langle P \rangle$, which measures the density of the MIS, is the coarsest measure of the lattice gas, corresponding to its density. The Hamiltonian of the lattice gas is,

$$H = \sum_{\{ij\}} J_{ij} n_i n_j - \mu \sum n_i. \tag{6.1}$$

The density of the lattice gas within the replica symmetric assumption is,

$$\rho = \int_{-\infty}^{\infty} dh g(h) \frac{\exp(h)}{1+\exp(h)}. \tag{6.2}$$

For more on the origin of this function see Eq. (B.43) in Appendix B. I want to make a note here about the meaning of the function $g(h)$. Weigt and Hartmann describe its physical interpretation as, "take a site i, then its average local occupation number $\langle x_i \rangle_\mu$ in the presence of the chemical potential μ can be written as $\exp[h_i]/(1+\exp[h_i])$ using an effective chemical potential h_i accounting for all interactions on i. $g(h)$ can now be constructed as the histogram of these effective chemical potentials." This is similar to the probability distribution of the order parameter, typical of spin glasses.

To study these problems, in the absence of suitable exact algorithms, we use heuristics that capture or give an insight into the physics of the problem. In this chapter, we present the results for the local probability algorithms presented in chapter 4. These algorithms capture some of the physics, namely good results for the order-parameter distribution, or the site probabilities. However, they fail to correctly find the frozen fraction. This chapter contains the results for a site based heuristic, and a bond based heuristic. The site based heuristic which we have termed vertex-LoPR is based on iterating Eq.(4.15) until a steady state value of the average site probability is found. The edge algorithm known as bond-LoPR iterates equations Eqs. (4.27)-(4.40), which ensure that each bond is covered. These are used to generate the site probabilities in Eq. (4.36).

The lattices used in these calculations were constructed using the same method as in Chapter 5. The MIS site probabilities, P_i, were calculated using double precision on 32-bit Linux PCs, and converged to an accuracy of 5×10^{-8}.

Figs 6.1, 6.3, 6.5, 6.7, 6.9 show histograms of the site probabilities for various large lattices. For the histograms, average connectivities, of $c = 1.0, 2.0, 4.0$ were chosen to give graphs of the same size a similar number of bonds. The graphs illustrate the variability in the site probabilities typical of hard computational problems and that there is a nontrivial order parameter distribution. There are pronounced peaks at the binary values $P = 1$ and $P = 0$. The strong peaks at 0 and 1 show that the vLoPR algorithm finds a greater number of integer probability or "frozen" sites as compared to bLoPR. On all bond algorithm frozen fraction graphs we see probability spikes at roughly $P \sim 0.35, 0.5, 0.7$ which correspond to those seen by Weigt and Hartmann and attributed to "dangling ends of the infinite cluster" [94].

6.1 LoPR on the triangular lattice

The calculations for the histogram of site probabilities were performed on $N = 40,000$ node triangular lattices with free boundary conditions, with one sample for each value of the average coordination number. In figure 6.1(a), we observe that as the average coordination of the graph increases the degenerate continuum decreases for vLoPR. As the average coordination increases, the individual site probabilities become less random, and more confined to specific values. This is shown by the strong peak at slightly below $P = 0.4$, and the less prominent peaks near $P = 0.08$, 0.23, 0.42, 0.48. When finding the MIS on a triangle we would expect each node to appear in the MIS $1/3$ of the time, so we would expect a peak at $P = 1/3$, the peak is slightly higher than this value near $P = 0.4$. As stated in chapter 4 this value is a result of the algorithm (note: The limiting cover probability on a triangle, $\langle x \rangle = 0.61803$ means that $\langle P \rangle = 1 - 0.61803 = 0.38197$.)

For the bond algorithm the degeneracy increases with increasing c. In the results of bLoPR, there exist peaks in the $Frequency(P)$ at $P = 0.35, 0.5, 0.7$. The behaviour of the number of nodes in the degenerate state, (the degenerate continuum), can be understood from the frozen fraction, $\langle F \rangle$, as plotted in Figure 6.2(b), for as $\langle F \rangle$ decreases the number of nodes in the degenerate state must increase. From both vLoPR and bLoPR it is observed that as the number of nodes in the degenerate continuum increases the roughness of $Frequency(P)$ decreases.

The average value of the MIS found from the site and bond algorithms are presented in Figure6.2(a) along with the average MIS found from the transfer matrix

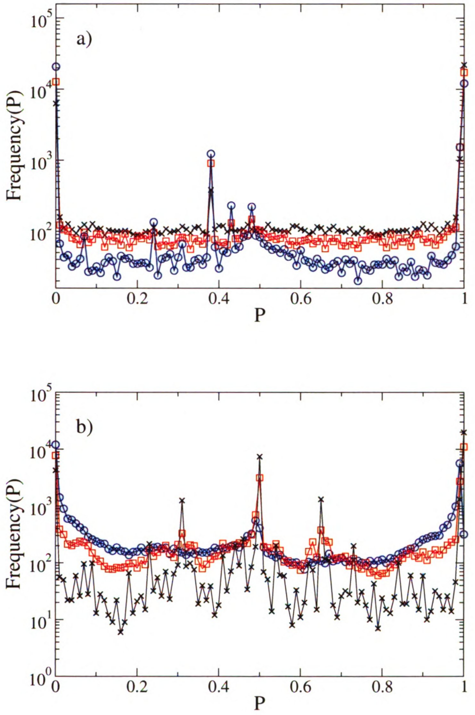


Figure 6.1: Distribution of the MIS probabilities $Frequency(P)$ for triangular lattices with $N = 40,000$ sites and with free boundaries at $c = 1.0(\times)$, $c = 2.0(\square)$, $c = 4.0(\circ)$ (a) vertex LoPR and (b) bond LoPR.

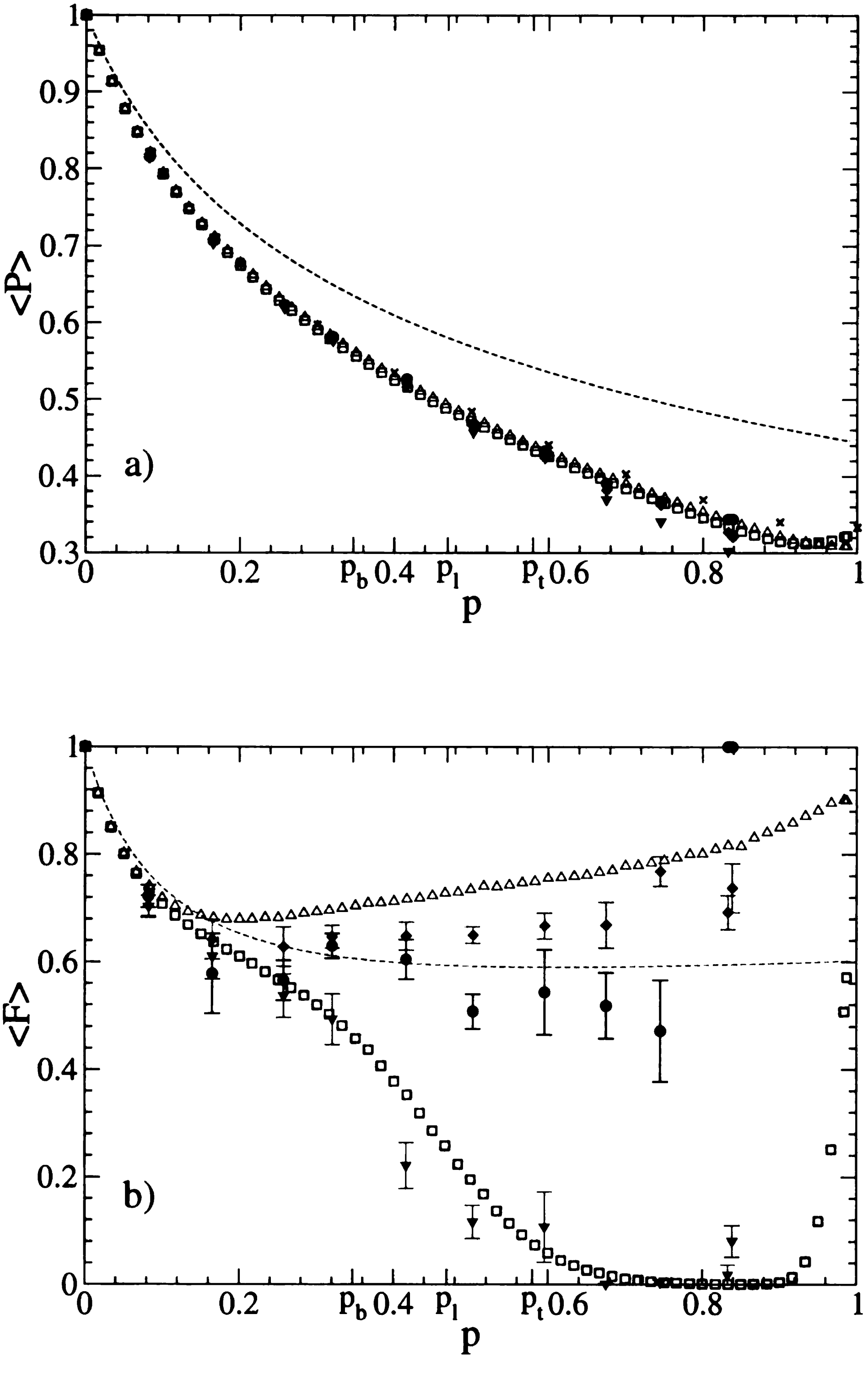


Figure 6.2: (a) The average MIS per site $\langle P \rangle$ and (b) the frozen fraction of sites $\langle F \rangle$ for N=10,000 triangular lattices averaged over 100 configurations. The data are for site LoPR(△), bond LoPR (□), transfer matrix in (a) (×), the symmetric Bethe approximation for z=6 (dotted line) and exact results on 64 node lattices (•) averaged over 10 configurations, vertex LoPR on 64 node lattices (♦) and bond LoPR on 64 node lattices (▼) . In this figure, p_b is the connectivity percolation threshold while p_l is the leaf removal threshold, and p_t is the triangle removal threshold.

method as prepared by Ji-Wu Liu, another student working with Dr. Duxbury. The figures contain the results of the symmetric Bethe lattice (z=6) result of Eqs (3.35)-(3.37), and the exact and LoPR results for $N = 64$ nodes lattices. The average value of the MIS was calculated for $N = 10^5$ with the MIS averaged over 100 instances. The exact values on the 64 node lattice, and their corresponding LoPR values, were carried out with free boundary conditions, and the results were averaged over 10 realizations. We can clearly see that when $p < 0.3$ the vLoPR and the transfer matrix results are very close, while the bond algorithm is slightly lower than either vLoPR or the transfer matrix for nearly all values of p. When one compares the LoPR algorithms on the same lattices as the exact solver (64 node lattices) (solid circle, solid diamond, solid triangle) we see a similar result, with the LoPR algorithms only at most a few percent different. LoPR is also a significant improvement over the symmetric Bethe approximation.

The average fraction of frozen sites as calculated on triangular lattices is presented in Figure6.2(b). Again we have the results of the site, bond, Bethe lattice and exact results on small lattices along with results for LoPR on $N = 10,000$ node triangular lattices. We can see that as the connectivity increases the frozen fraction for the site algorithm decreases to a minimum near the bond percolation threshold, then increases over the rest of the connectivity range. This increase mirrors the decrease in degeneracy that was observed in the histogram of site probabilities for the site algorithm. For the bond algorithm the frozen fraction decreases, corresponding to the increase in the degeneracy on Figure 6.1(b). In the frozen fraction, as calculated by the bond algorithm, there are two inflection points. The first appears near the bond percolation critical point and the second near the core percolation critical point. It is interesting to observe that while the value of the MIS found from the LoPR algorithms is quite close, the frozen fraction is not close to the correct value. This implies that the LoPR algorithms do not fully explore the solution space, yet the algorithms are able to find near optimal states.

6.2 LoPR on the square lattice

The histogram of site probabilities of the square lattice were calculated on $N = 40,000$ node lattices with free boundary conditions, with one realization for each value of average coordination number. In Fig 6.3(a) behaviour similar to the triangular lattice is observed, namely a decrease in the degenerate continuum and an increase in roughness with an increase in average coordination. The primary differences are the absence of a peak near $P = 0.4$, and that $Frequency(P)$ appears symmetric around $P = 0.5$. The first of these differences speaks to the lack of triangles and odd numbered loops, and the latter to the bipartite nature of the lattice. The main structure that is apparent is a slight arch centered around $P = 0.5$.

In the bond algorithm, Fig 6.3(b) the degenerate continuum increases as the connectivity increases from $c = 1.0$ to $c = 2.0$ and then decreases as the connectivity increases to $c = 4.0$, This can be understood by examining the frozen fraction $\langle F \rangle$ in Fig 6.4(b). At $c = 1.0$ there is a high level of roughness that is likely due to the small number of edges relative to the number of nodes. When there are small clusters the bLoPR algorithm is able to resolve the small clusters, leading to spikes in the $Frequency(P)$ related to recurring structures. We can see that as the number of edges increases the roughness tends to decrease.

Fig 6.4 shows the value of the MIS and the frozen fraction on a square lattice as calculated by the two LoPR procedures. Results for $N = 10^5$ node square lattices averaged over 100 realizations at each connectivity are presented. The exact and vLoPR and bLoPR results from $N = 64$ node square lattices averaged over 10 configurations are also shown, along with the Bethe lattice calculation for $z = 4$. For most values of p the probability of a site being in the MIS calculated by

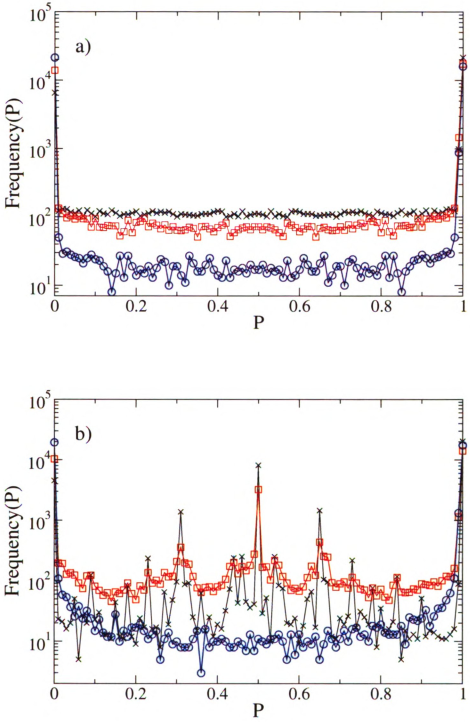


Figure 6.3: Distribution of the MIS probabilities Frequency(P) for square lattices of $N = 40,000$ sites with free boundaries at c=1.0(×), c=2.0(□), c=4.0(○): (a) vertex LoPR and (b) bond LoPR

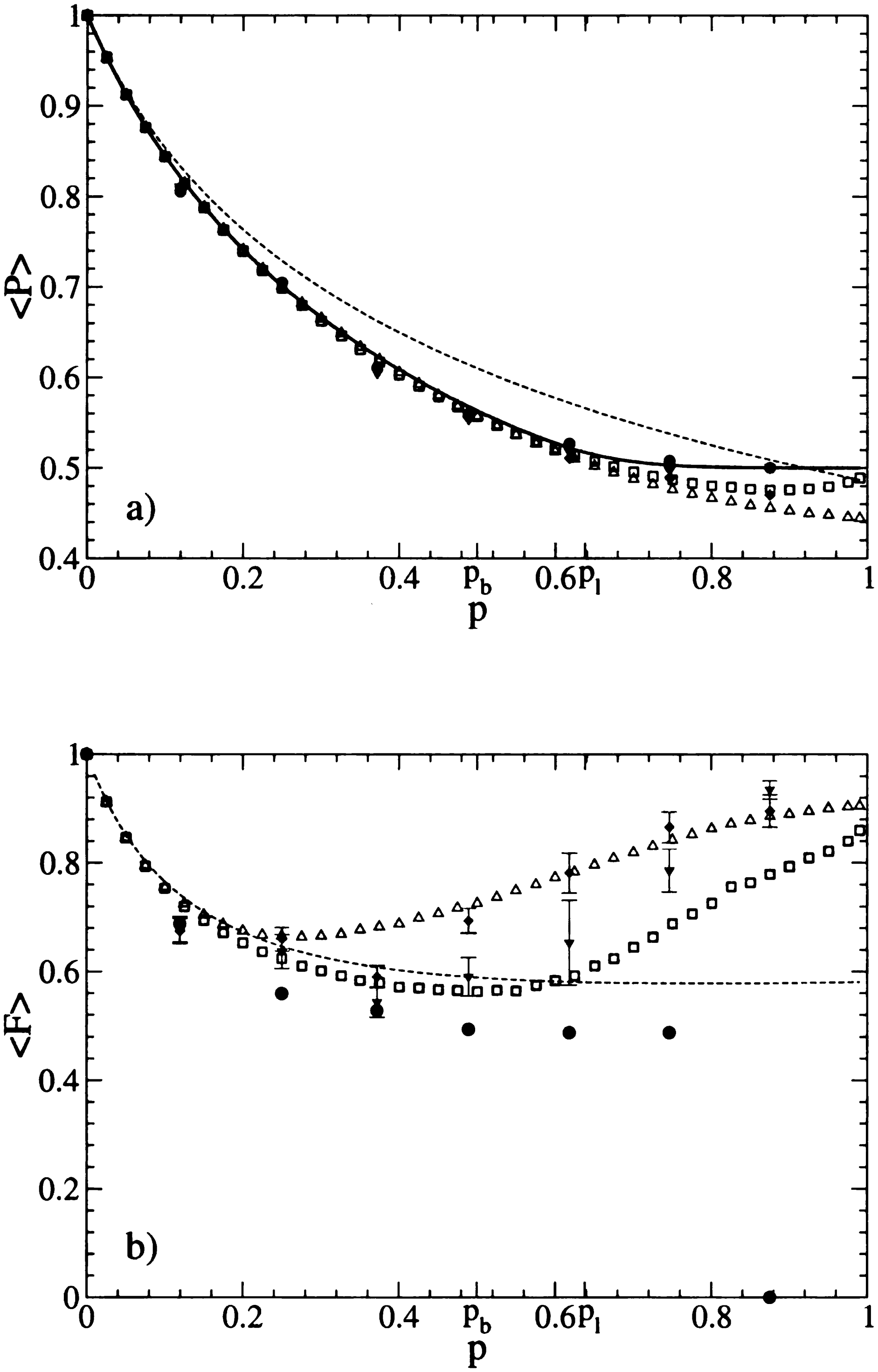


Figure 6.4: (a) The average MIS per site $\langle P \rangle$ and (b) the frozen fraction of sites $\langle F \rangle$ for N=10,000 square lattices averaged over 100 configurations. The data are for site LoPR(△), bond LoPR (□), transfer matrix in (a) (×), the symmetric Bethe approximation (dotted line) and exact results on 64 node lattices (•) averaged over 10 configurations, vertex LoPR on 64 node lattices (♦) and bond LoPR on 64 node lattices (▼). The solid line is the lower bound MIS found through leaf removal. In this figure, p_b is the connectivity percolation threshold while p_l is the core percolation threshold.

both LoPR algorithms are close to the exact values, with the bond algorithm being slightly lower for smaller values of p and being better at higher values. bLoPR performs better than vLoPR when $p > p_b$. The Bethe lattice with $z = 4$ deviates significantly from the LoPR values.

From the geometry of the problem, we know that once the graph has its full complement of edges the fraction of independent sites is $P = 0.5$ exactly, and the frozen fraction is $F = 0$. The lattice is bipartite, without disorder and with sub-lattices of identical cardinality. In this case a maximum independent set may be constructed from one of the two identical sub-lattices. Each sub-lattice has a cardinality of $N/2$. As such, there are no intersecting solutions and $F = 0$. We expect that as p increases the cardinality of the MIS should tend to $P = 0.5$ and the frozen fraction should go to 0. We see that the bond algorithm trends toward the correct answer for the MIS at high connectivity, just as it did on the triangular lattice.

To find the size of the MIS using core percolation on a bipartite lattice two things need to be known. The first is that when leaf removal is applied the correct value for the MIS of the removed parts of the lattice is known. The second issue is how one finds the lower bound on the core. As such we make the following conjectures, about the cardinality of a dilute hypercubic lattice.

1. The MIS of hypercubic lattices without leaves is unfrustrated and equal to the cardinality of the largest sub-lattice of the graph.

2. The MIS on site or bond diluted hypercubic lattices is frustrated, and this frustration is due only to the leaves. Leaf removal reduces the graph to those treatable using conjecture 1.

3. conjectures 1 and 2 imply that asymptotically the exact MIS of a diluted hypercubic lattice is equal to the number of leaves removed, plus 1/2 the num-

ber of vertices in the core.

The MIS should be a monotonically decreasing function for all connectivities, one cannot add edges and increase the number of independent nodes. Also, on a bipartite graph, $\langle P \rangle$ should never be below $\langle P \rangle = 0.5$. The max independent set is bounded below by the larger of the two sub-lattices. To be fully accurate we need to consider that the graph is constructed such that it is a loose connection of bipartite clusters; the MIS on each cluster being the largest sub-lattice of that cluster. Then the MIS is the sum of the IS for each cluster. Using this information we can construct a lower bound by adding the number of independent nodes found through leaf removal to one-half the number of nodes in the core. This gives a lower bound without measuring the size of the sub-lattices, and is expected to be a good approximation as illustrated by the solid line in Figure 6.4(a).

The frozen fraction as calculated by vLoPR does not go to 0 as $p \rightarrow 1$. $\langle F \rangle$ for the site algorithm displays similar behaviour as on the triangular lattice. However, unlike the triangular lattice the minimum in $\langle F \rangle$ does not occur near the bond percolation threshold. The frozen fraction of bLoPR, in contrast to its behaviour on the triangular lattice, decreases to a minimum near the bond percolation critical point before increasing. The shape of the frozen fraction as calculated by bLoPR is similar to that calculated by vLoPR, though with a broader peak. The exact results for $\langle F \rangle$ on the $N = 64$ node lattices is significantly lower than either of the LoPR results on the same lattices. At the highest value of p on the 64 node exact lattice we see the frozen fraction go to 0, while large sections of the lattice in the LoPR algorithms remain frozen. At the maximum connectivity all bonds are present in the 64 node lattice.

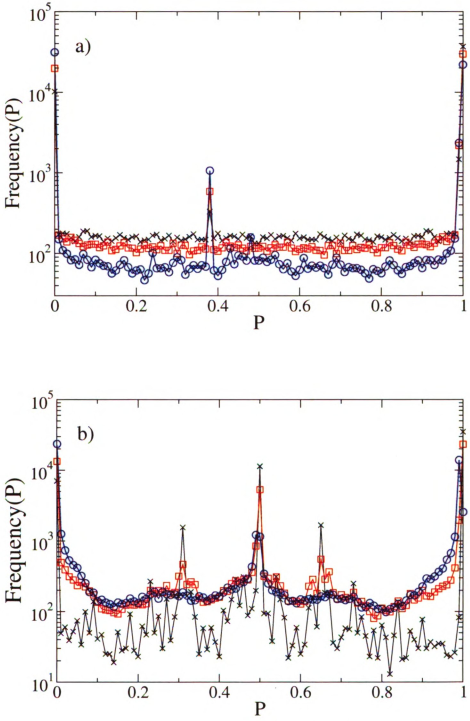


Figure 6.5: Distribution of the MIS probabilities Frequency(P) for FCC lattices of N=64,000 sites with free boundaries at c=1.0(×), c=2.0(□), c=4.0(○): (a) vertex LoPR and (b) bond LoPR.

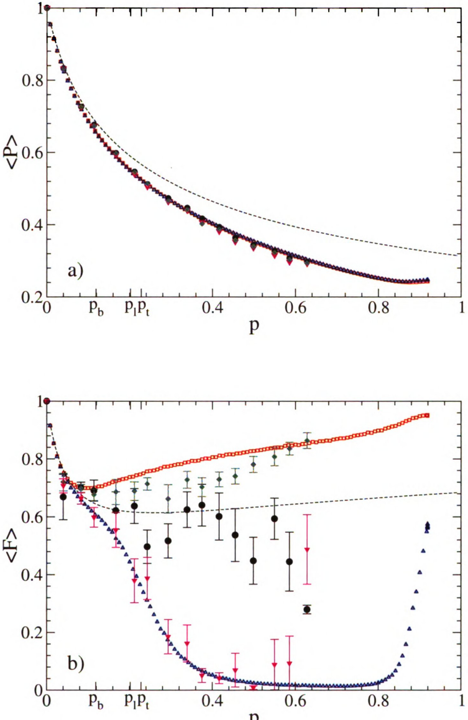


Figure 6.6: (a) The average MIS per site $\langle P \rangle$ and (b) the frozen fraction of sites $\langle F \rangle$ for N=8,000(L=20) FCC lattices averaged over 100 configurations. The data are for vertex LoPR(△), bond LoPR (□), the symmetric Bethe approximation (dotted line) and exact results on 64 node lattices (●) averaged over 10 configurations, vertex LoPR on 64 node lattices (♦) and bond LoPR on 64 node lattices (▼). In this figure, p_b is the connectivity percolation threshold while p_l is the leaf removal threshold, and p_t is the threshold for triangle removal.

6.3 LoPR on the FCC lattice

Not surprisingly, the behaviour of the degenerate continuum on FCC lattices is similar to its behaviour on the triangular lattice. In Fig 6.5(a) we see that the degenerate continuum decreases with increasing connectivity when using vLoPR. This is also accompanied by an increase in the roughness of the *Frequency*(P). There also exists a peak near $P = 0.4$ that appears only for the FCC and triangular lattices. The other smaller peaks seen in the triangular lattice are not observed. An examination of the distribution of site probabilities from the bond algorithm, For bLoPR, Fig 6.5(b) shows the three peaks related to dangling ends. An increase in the degeneracy in the *Frequency*(P), as the connectivity increases, is also observed. On the FCC lattice the degenerate continuum appears to increase as we traverse the connectivity from $c = 1.0$ to $c = 2.0$, then it appears to level out for $c = 4.0$. These calculations were performed on $N = 64,000$ node FCC lattices with free boundary conditions, over one configuration at each value of p.

Fig 6.6(a) shows the fraction of nodes in the MIS for all values of p. Here, the LoPR results are close to the exact values, the bond LoPR algorithm being slightly lower. The first deviation of LoPR from the exact solutions appears to be near the bond percolation critical point, p_b. At higher connectivities the bond based procedure improves its performance, giving the correct answer on fully filled lattices.

$\langle F \rangle$ for the FCC lattice behaves as it does on the triangular lattice, like the triangular lattice the FCC lattice shows a minimum in $\langle F \rangle$ for vLoPR near the bond percolation threshold, p_b. At $p \approx p_b/2$ the frozen fraction from the LoPR algorithms begins to diverge from the frozen fraction on the Bethe lattice. The frozen fraction calculated by bLoPR shows the first inflection point near p_b. Unlike the triangular lattice, the second inflection point of the bLoPR frozen fraction point not appear to be near the leaf removal threshold, p_l. These calculations were carried out on $N = 8000$ node FCC lattices with free boundary conditions averaged over

100 configurations. The exact results are for $N = 64$ node lattices averaged over 10 configurations.

6.4 LoPR on the simple cubic lattice

On the cubic lattice the degenerate continuum decreases as a function of connectivity for the vLoPR algorithm. There is also a small increase in the roughness of $Frequency(P)$, as can be seen in Figure 6.7(a). The cubic lattice shows the same apparent symmetry and slight arch about $P = 0.5$ that the square lattice shows, though the arch is narrower on the cubic lattice. In Figure 6.7(b) we see that as the connectivity increases the degenerate continuum of bLoPR increases for $c = 2.0$, then decreases again for $c = 4.0$. This is mirrored as we expect in the $\langle F \rangle$, in Figure 6.8(b), since the degenerate continuum is related to the number of frozen nodes. The calculations were carried out on one configuration of a $N = 64,000$ node simple cubic lattice, with free boundary conditions.

The results for $\langle P \rangle$ and $\langle F \rangle$ as shown in Fig 6.8, were calculated for $N = 8000$ node lattices averaged over 100 realizations, exact, vLoPR and bLoPR calculations from averages of 10 realizations on $N = 64$ node lattices and the replica symmetric solution for $z = 6$. Examining the results $\langle P \rangle$ in Fig 6.8(a) one sees the divergence between vLoPR and bLoPR begins near the leaf removal threshold, p_l. $\langle P \rangle$ on the cubic lattice like the square lattice also deviates below the correct value of $\langle P \rangle = 0.5$, and the deviation from the exact value occurs at some point after the leaf removal threshold. Like the square lattice, the geometry of the lattice tells us that at high connectivities the covered fraction $P_I = 0.5$ and the frozen fraction $\langle F \rangle = 0$. The simple cubic lattice also displays the same behaviour in the MIS as compared to the lower bound on the MIS from leaf removal, deviating significantly at higher connectivities. The frozen fraction from the bond algorithm on the cubic lattice

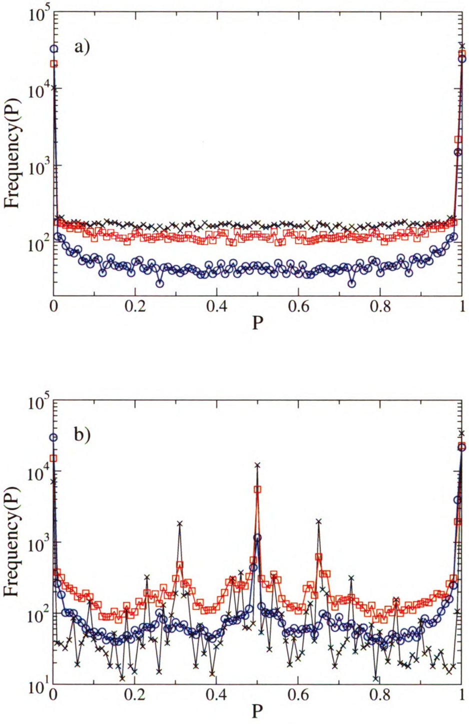


Figure 6.7: Distribution of the MIS probabilities Frequency(P) for simple cubic lattices of $N = 64,000$ sites with free boundaries at c=1.0(×), c=2.0(□), c=4.0(○): (a) vertex LoPR and (b) bond LoPR.

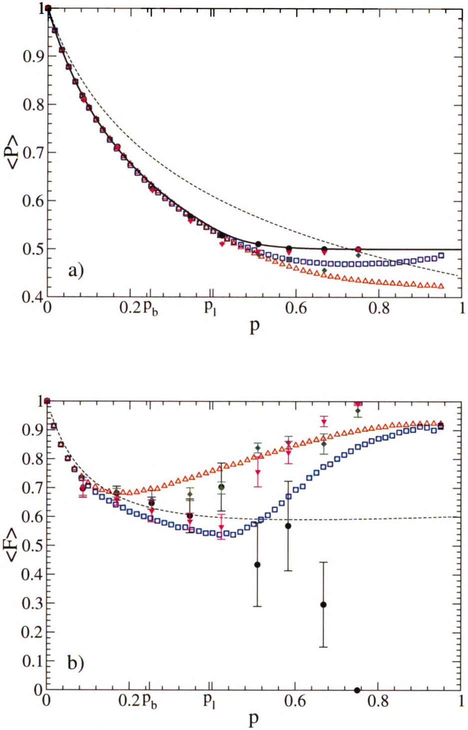


Figure 6.8: (a) The average MIS per site $\langle P \rangle$ and (b) the frozen fraction of sites $\langle F \rangle$ for N=8,000 (L=20) for the simple cubic lattice averaged over 100 configurations. The data are for site LoPR(△), bond LoPR (□), the symmetric Bethe approximation (dotted line) and exact results on 64 node lattices (•) averaged over 10 configurations, vertex LoPR on 64 node lattices (♦) and bond LoPR on 64 node lattices (▼). The solid line is the lower bound MIS as calculated by leaf removal. In this figure, p_b is the connectivity percolation threshold while p_l is the core percolation threshold.

demonstrates a similar increase near the leaf removal threshold, p_l, although with a more pronounced cusp. At high connectivities, the bond and vertex algorithm approach the same value for the frozen fraction. LoPR nearly freezes the entire lattice at the highest connectivity $N = 64$ node lattice, where the exact solver gives a $\langle F \rangle = 0$. This implies that with the greater number of solutions and high energy barriers between solutions, the LoPR has difficulties finding all ground state configurations.

6.5 LoPR on the random graph

On the random graph we see that as the connectivity increases the degenerate continuum decreases for the vertex algorithm and increases for the bond algorithm, Fig 6.9. Examining the results from the vertex algorithm, we observe dips in the $Frequency(P)$ near $P \approx 0.2, 0.4, 0.6, 0.8$. This is likely due to the increase in the number of loops interfering with the tree like structure of the clusters. The results from the bond algorithm shows that the roughness which is present at $c = 1.0$ has largely disappeared by $c = 2.0$ and diminishes even more by $c = 4.0$. The degenerate continuum appears to be roughly symmetric about $P = 0.5$ as on the square and cubic lattices. This is likely due to the bipartite tree like structure of dilute random graphs. The calculations were performed on $N = 40000$ node random graphs, with one configuration at each connectivity.

Displayed in Figure 6.10 are the results for the probability of a site being on the MIS and the frozen fraction. Here, we see very good results for the vertex algorithm and bond algorithm as compared to the exact results. The random graph is different from the previous lattices since the connectivity range tested never gets close to "filling" the lattice as such there are less short range loops in the graph. The deviation between the replica symmetric solution and the LoPR solutions occur

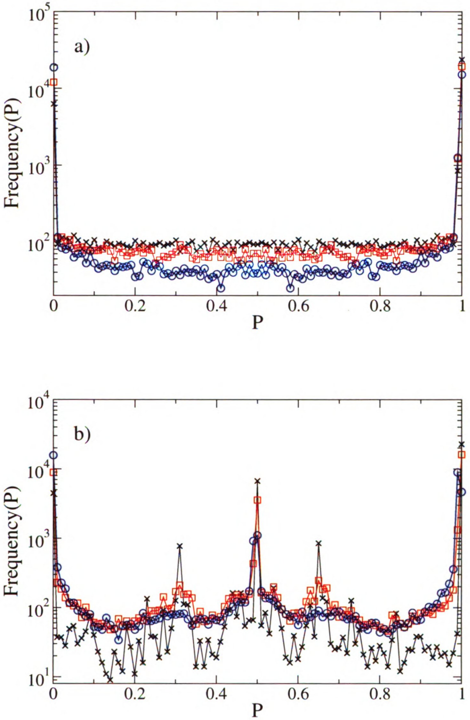


Figure 6.9: Distribution of the MIS probabilities Frequency(P) for random graphs of N=40,000 sites at c=1.0(×), c=2.0(□), c=4.0(○): (a) vertex LoPR and (b) bond LoPR.

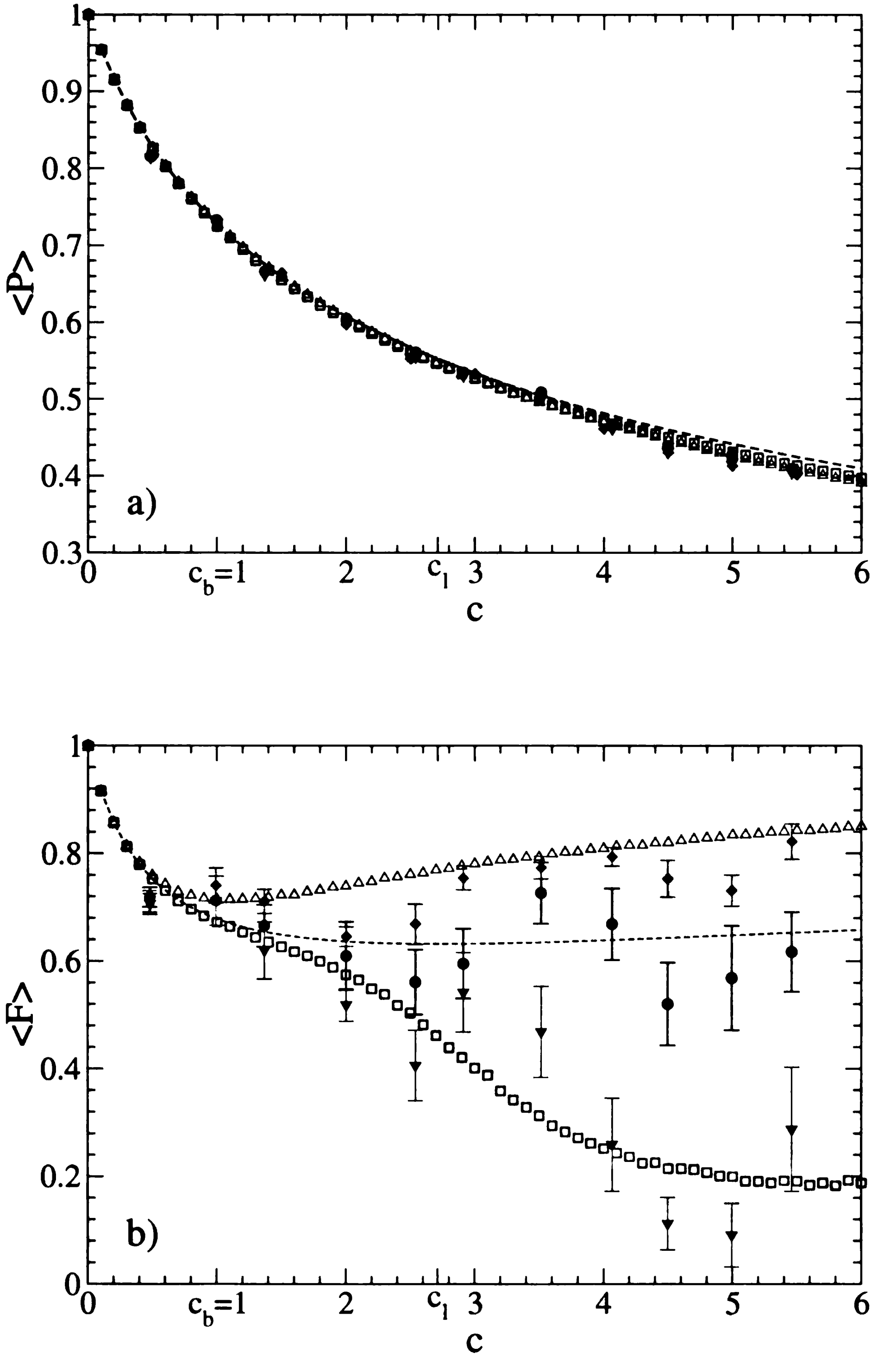


Figure 6.10: (a) The average MIS per site $\langle P \rangle$ and (b) the frozen fraction of sites $\langle F \rangle$ for N=10,000 random graphs averaged over 100 configurations. The data are for site LoPR(△), bond LoPR (□), the symmetric Bethe approximation ($*$) and exact results on 64 node lattices (●) averaged over 10 configurations, vertex LoPR on 64 node lattices (◆) and bond LoPR on 64 node lattices (▼). In this figure, p_b is the connectivity percolation threshold while p_l is the core percolation threshold.

near the leaf removal threshold, c_l. The replica symmetric solution is good until the $c_l = e$, above the leaf removal threshold we find that both LoPR algorithms give a better solution.

The LoPR calculated frozen fraction appears to have more in common with the triangular and FCC lattice than with the square and cubic lattices. This is likely due to the omission of triangles and odd loops from the square and cubic lattices. The data does not show the increase in the frozen fraction from the bond LoPR algorithm that we see in triangular and FCC lattices, at high connectivities. The random graph displays a similar two inflection point curve in the frozen fraction as calculated by bLoPR. Similar to the triangle lattice the first inflection point appears around the bond percolation threshold, c_b and the second after the leaf removal threshold, c_l. The calculations were carried out on $N = 10000$ node random lattices averaged over 100 configurations, and the exact results on $N = 64$ node random lattices, averaged over 10 configurations.

6.6 Summary

On all lattices we see individual node probabilities are affected by the type of algorithm, the specific geometries of the lattice, and the occurrence of small structures specifically, dangling ends and triangles. vLoPR appears to be affected by triangles, but not leaves, while bLoPR is affected by leaves. Triangles give rise to a peak in the order parameter distribution near $P \approx 0.4$ for the FCC and triangular lattices on vLoPR. There also appears to be a small amount of structure in the $Frequency(P)$ on the cubic and square lattices near $P \approx 0.5$, due to small complete bipartite structures. vLoPR is less sensitive to leaves, but more sensitive to small loops such as triangles, as the individual probabilities are affected by highly correlated sites. bLoPR includes a more complex treatment which is dependent upon the next nearest neighbor.

In the calculation of the probability of a site being in the MIS, the bond LoPR procedure shows better performance at high connectivities than the vertex algorithm, with both the vertex and the bond algorithm out performing the symmetric Bethe approximation. On random graphs, LoPR methods yielded values for the average MIS that are precise for $p < p_l$, where the symmetric Bethe approximation are expected to be exact [94] and are accurate to within a few percent for $p > p_l$ where replica symmetry breaking is important. The LoPR method provides a good lower bound on the MIS. In terms of finding the MIS on bipartite graphs, a lower bound may be found most easily using leaf removal, where the lower bound is equal to the number of leaves removed, plus 1/2 the number of vertices in the core.

Estimating the frozen fraction is more difficult, as neither the site nor the edge algorithm provides correct predictions. The site algorithm consistently over estimates the frozen fraction, while the bond algorithm generally underestimates the frozen fraction. The exception to this being at high connectivity on bipartite graphs, where both algorithms freeze more nodes than the exact solution.

Chapter 7

Conclusion

7.1 Conclusion

In this thesis, we have examined the ground state of the geometrically frustrated hard-core lattice gas. Geometrical frustration on diluted lattices leads to glassy behaviour. On bipartite lattices, the hard-core lattice gas maps to the diluted anti-ferromagnet in a field. Finding the ground state of the hard-core lattice gas is difficult and has been found to map to the computationally difficult class of problems known as NP-complete. In order to model the ground state we reduced the complexity of the lattice by removing small cliques, such as leaves and triangles. Further, we found that a local heuristic can model the density of the ground state, and reproduce the non-trivial order-parameter distribution.

We have examined the percolative properties of leaf and triangle removal. Both leaf-removal and triangle removal run in polynomial time. Similar methods for larger cliques would exist, though they would eventually become prohibitively difficult to solve, because clique is also NP-complete. For core percolation, there exists a series of transitions, for each order of clique removed, that appear to coalesce as the dimensionality grows. The results of core percolation in Chapter 5 show that triangle and leaf removal are in the same universality class as standard percolation. Nevertheless, several exponents showed deviation from the standard values, but these compared favorably to bond percolation calculations made on similar lattices (see appendix A).

The LoPR algorithm was shown to be robust, providing near extremal values for the Vertex Cover. The algorithms run very quickly by limiting the number of times the order of iteration is randomized. The bond algorithm generally out performed site LoPR. Larger numbers of nodes were frozen by the site algorithm. Neither algorithm correctly captured the frozen fraction. This means that the algorithm is able to find near optimal solutions, but does not explore the complete solution space. This doesn't appear to have a major effect on results of the MIS as the correct solution appears to be found over 50 percent of the time, after a distinct instance is generated for the $N = 100$ triangular lattice tested. Both the site and bond algorithms out perform the replica symmetric solution. In general, the LoPR family of algorithms provide a quick method to get the ground state of MIS/VC within a few percent.

7.2 Further Work

One of the things that needs to be done in the future, is to continue the analysis of core percolation by studying in more detail the effects of the boundary conditions on the critical exponents. This would be particularly useful for the three dimensional lattices. Larger sample sizes for the lattices already studied may be come accessible as computer speed increases. This would allow greater examination of the critical exponents affected by finite size effects, particularly β.

The analytic evaluation of the bond-LoPR method needs to be examined in still greater detail to determine if a generalized equation can be written to describe higher moments. Analytic work on the LoPR algorithms might also provide rigorous lower bounds on MIS.

There are several possible extensions to LoPR; the algorithms could be tested on different lattices. Perhaps a hybrid algorithm joining site and bond LoPR might have a better approximation to the frozen fraction and find the correct probabilities on a triangle. An algorithm built on finding the correct probability of triangles, might provide interesting results. We might try constraining the covered fraction in the LoPR algorithms. This might allow us to more fully explore the solution space, by comparing the frozen nodes over a number of similar solutions. Site-LoPR with a single randomized node list runs quickly and could be used to generate a "solution". We could then flip the probabilities and converge to a second solution, which could be continued until a "best" solution is found. This would likely increase the probability of finding a true minimum. This might also be used to explore the free energy landscape of the LoPR algorithms.

It would be interesting to examine the behaviour of core percolation and the LoPR method on scale free random graphs, and small world graphs.

APPENDICES

Appendix A

Percolation

A.1 Introduction

Percolation is a process where nodes in a lattice are joined randomly. This is a growth process by which elements of a graph (such as edges or nodes) are placed into a lattice with some probability p. In bond percolation, the probability, p, is a function of the average coordination of a site, c, and in the case of a regular lattice, the coordination of the lattice, z,

$$p = \frac{c}{z}. \tag{A.1}$$

The average coordination can be calculated by,

$$c = \frac{2E}{N}, \tag{A.2}$$

where, E is the number of edges in the lattice, and N is the number of nodes in the lattice.

At any concentration there will be a finite probability that the largest cluster will span the sample, P_S. For an infinite sample, the probability,

$$P_S(p) = \begin{cases} 0 & p < p_c \\ 1 & p \geq p_c. \end{cases} \tag{A.3}$$

For finite systems there is a transition region between 0 and 1.

We can also define a probability that a node is on the spanning cluster, P_∞, the infinite cluster probability.

A.2 The infinite cluster on the Bethe lattice

Here we will approximate the infinite cluster near the transition point using a Bethe lattice. The Bethe lattice has α connections to each subsequent layer, and the total coordination of each level is $z = \alpha + 1$. Each bond occurs with probability p and is absent with probability $(1 - p)$ Then we can write the probability, P_{k+1} that a site at level k+1, is connected to the boundary as,

$$P_{k+1} = \sum_{l=1}^{\alpha} \binom{\alpha}{l} (pP_k)^l (1 - pP_k)^{\alpha - l}. \tag{A.4}$$

Where, pP_k is the probability that the kth level is node is connected to the boundary and the bond to the k+1 level is present (with the zeroth level being the boundary). As the size of the lattice goes toward infinity, the probability converges to a steady state value that we will call, P.

$$\begin{aligned} P &= \sum_{l=1}^{\alpha} \binom{\alpha}{l} (pP)^l (1 - pP)^{\alpha - l}, & \text{(A.5)} \\ &= 1 - (1 - pP)^{\alpha}, & \text{(A.6)} \\ &= 1 - \left(1 - \frac{\alpha p P}{\alpha}\right)^{\alpha}. & \text{(A.7)} \\ & & \text{(A.8)} \end{aligned}$$

If we take the limit of α as it goes to infinity,

$$
\begin{aligned}
P &= \lim_{\alpha \to \infty} \left(1 - \left(1 - \frac{\alpha p P}{\alpha}\right)^{\alpha}\right), && \text{(A.9)} \\
&= 1 - \exp(-\alpha p P). && \text{(A.10)}
\end{aligned}
$$

Then if we make an expansion of the exponential to ,

$$\exp(-\alpha p P) = 1 - \alpha p P + \frac{(\alpha p P)^2}{2!} - \frac{(\alpha p P)^3}{3!} + \ldots, \qquad \text{(A.11)}$$

taking the first three terms and substituting them into our probability, we have,

$$
\begin{aligned}
P &= 1 - 1 + \alpha p P - \frac{(\alpha p P)^2}{2!} && \text{(A.12)} \\
&= P\left(\alpha p - \frac{(\alpha p)^2 P}{2}\right), && \text{(A.13)}
\end{aligned}
$$

This leads to,

$$1 = \alpha p - \frac{(\alpha p)^2 P}{2}. \qquad \text{(A.14)}$$

Which has the solution,

$$P = -(p - \frac{1}{\alpha})\frac{2}{\alpha p^2}. \qquad \text{(A.15)}$$

Thus the probability of a node being on the infinite cluster scales as,

$$P_{\infty} \propto (p - p_c)^{\beta}, \qquad \text{(A.16)}$$

with $p_c = \frac{1}{\alpha}$, and $\beta = 1$ for the Bethe lattice.

A.3 Mean cluster size

Besides the infinite lattice probability, $P_\infty(p)$ there are three other quantities of interest; $P_S(p)$, the spanning probability, $n_S(p)$, the normalized cluster number, and $\langle S\rangle$ the mean cluster size. The normalized cluster distribution is defined as the number of size s clusters per lattice site. This leads to the definition of the mean cluster size as;

$$\langle S\rangle = \sum \frac{n_S s^2}{\sum n_S s}. \tag{A.17}$$

A.4 Finite-size scaling in percolation

Making the finite size scaling ansatz, we may say that any property of the graph $\chi(p, L)$ scales as,

$$\chi(p, L) = (p - p_c)^{-x} X((p - p_c)L^{1/v}). \tag{A.18}$$

We use finite size scaling to relate the behaviour at the critical point infinite samples to the behaviour of finite samples [23]. Some scaling relations that we have used in this dissertation are [84] [12],

$$\begin{aligned}
n_S(p_c) &\propto s^{-\tau} f_n(s^{\sigma}(p - p_c))L \to \infty, && \text{(A.19)}\\
\langle S\rangle &\propto \text{p-p\_c}^{-\gamma} f_a((p - p_c)L^{1/v}), && \text{(A.20)}\\
P_\infty(p_c, L) &\propto L^{-\beta/v} f_\infty(L^{1/v}(p - p_c)), && \text{(A.21)}\\
P_S(p_c, L) &\propto f_S(L^{-1/v}(p - p_c)), && \text{(A.22)}\\
\Delta &\propto L^{1/v}, && \text{(A.23)}\\
P_\infty &\propto (p - p_c)^{\beta}, && \text{(A.24)}\\
& && \text{(A.25)}
\end{aligned}$$

where Δ is the width of the transition in P_S,

$$\Delta^2 = \langle p^2 \rangle - \langle p \rangle^2. \tag{A.26}$$

A simple power law gives a straight line on a log-log plot, but for some correlated percolation processes, like rigidity percolation [66], we need to examine a next to leading order correction of the form.

$$\chi(p_c, L) = C_1 L^{-e}(1 + C_2 L^{-w}) \tag{A.27}$$

The scaling functions f_S, f_∞, f_n, f_a include the finite size scaling behaviors and enable determination of β, ν, and γ. Eq. (A.19) also gives the scaling behaviour in the infinite lattice limit at p_C and will enable determination of τ. Only two of the exponents are independent, as such a check can be provided by various exponent relations.

$$\nu = \frac{\beta\tau - 1}{2\tau - 2}, \alpha + 2\beta + \gamma = 2, d\nu = 2 - \alpha, \tag{A.28}$$

where d is the dimension.

A.5 Bond percolation on regular lattices with free boundary conditions

Table A.1: Standard Percolation Exponents

exponent	d=2	sq	triangular	d=3	cubic	FCC
ν	4/3	1.35(1)	1.37(3)	0.88	0.93(4)	0.92(2)
γ	43/18	2.26(7)	2.28(7)	1.8	1.7(1)	1.76(8)
τ	187/91	2.06(1)	2.08(2)	2.18	2.24(1)	2.24(2)

Here we present bond percolation values calculated for simulations run under similar conditions to those in Chapter 5. The accepted values of standard percola-

Table A.2: Comparison of Percolation Exponents

	ν_b	ν_l	ν_t
Sq.	1.35(1)	1.39(1)	–
Tri.	1.37(3)	1.37(1)	1.33(6)
Cubic	0.93(4)	1.00(5)	–
FCC	0.92(2)	0.982(5)	0.92(2)
	γ_b	γ_l	γ_t
Sq.	2.26(7)	2.19(3)	–
Tri.	2.28(7)	2.16(3)	2.13(6)
Cubic	1.7(1)	1.7(2)	–
FCC	1.76(8)	1.74(2)	1.61(3)
	τ_b	τ_l	τ_t
Sq.	2.06(1)	2.07(2)	–
Tri.	2.08(2)	2.09(5)	1.98(8)
Cubic	2.24(1)	2.23(6)	–
FCC	2.24(2)	2.17(8)	2.07(8)

tion, may be easily looked up in any number of books, including "Introduction to percolation theory" by Dietrich Stauffer and Amnon Aharony. The primary reason for doing these calculations was to examine the critical exponents ν, τ, and γ, of the 3-d lattices for free boundary conditions.

Figure A.1 provides an unbiased estimate of the exponent ν by using, $\delta p_c^2 = \langle p_c^2 \rangle - \langle p_c \rangle^2 \sim L^{-2/\nu}$. The results are recorded in Table A.2. The value of ν measured for the 2-d lattices is consistent with the standard percolation value and the leaf removal values. The values of ν calculated for the 3-d lattices are between the standard values and the leaf removal values. The values are very close to the value of ν found for triangle removal.

In Figure A.3 we calculated the exponent γ from the average cluster size, S, on approach to p_c from below. The value of γ calculated in the 3-d cases are consistent with the standard percolation value of $\gamma = 1.8$ $(d = 3)$. In 2-d, he standard percolation value of $\gamma = 2.38$ is higher than the values measured $\gamma_{sq} = 2.26(7)$ and $\gamma_{tri} = 2.27(7)$. The values measured here fall between the standard value and the core percolation values. The values for γ are strongly affected by finite size effects, as the slope is dependent upon the transition point which in finite samples is a function of the size of the samples.

Figs. A.4 and A.5 show the plots of $n_S L^d$ verses s used to calculated τ For the 3-dimensional lattices we measure a $\tau = 2.24(2)$ for bond percolation, which is consistent with the leaf removal values of $\tau = 2.17(8)$ for the FCC and $\tau = 2.23(6)$ for the simple cubic lattice. This suggests that the value we calculated for leaf removal is consistent with standard percolation. The value of τ for triangle removal on the FCC lattice is lower than both the standard quoted value and the value calculated here, it is however consistent with the standard value. The 2-d values are consistent with the leaf removal values.

After examining the critical exponents obtained by the same methods as used in the core percolation calculations, we can see that the values are similar. Any discrepancies between the standard values and the values measured here, and in core percolation, can be explained through finite-size effects and the results of the boundary conditions.

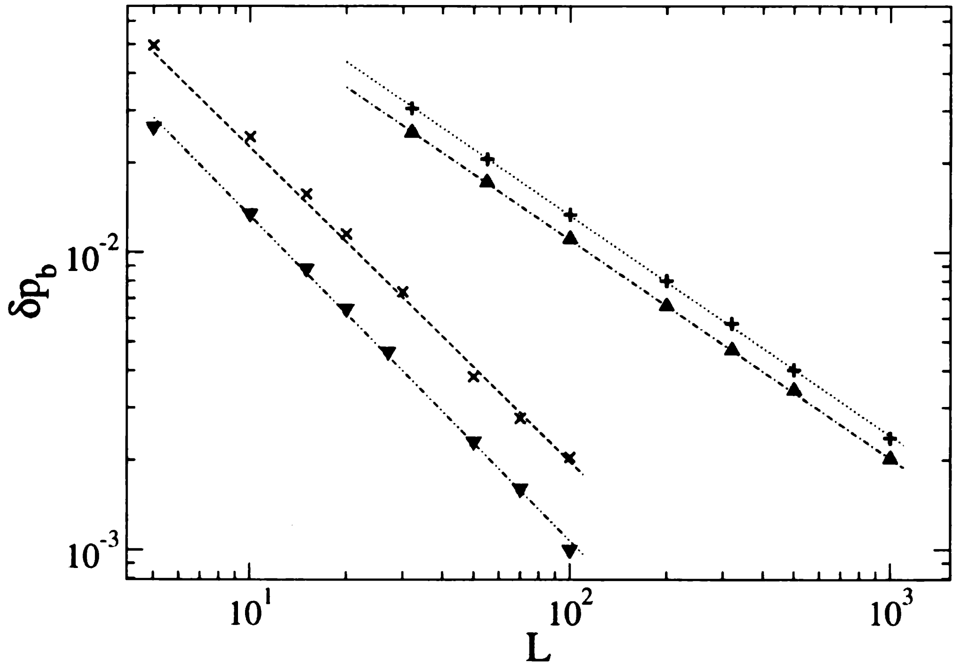


Figure A.1: Plots of $\delta p_l^2 = \langle p^2 \rangle - \langle p \rangle^2$ as a function of lattice size L on a double logarithmic graph. The $\triangle$ are data from the triangular lattice, the dashed line is a best fit from which we extract the estimate $\nu = 1.35(1)$. The + are data from the square lattice from which we extract the estimate $\nu = 1.37(3)$. The ∇ are data from the FCC from which we extract the estimate $\nu = 0.92(2)$. The $\times$ are data from the cubic lattice from which we extract $\nu = 0.93(4)$.

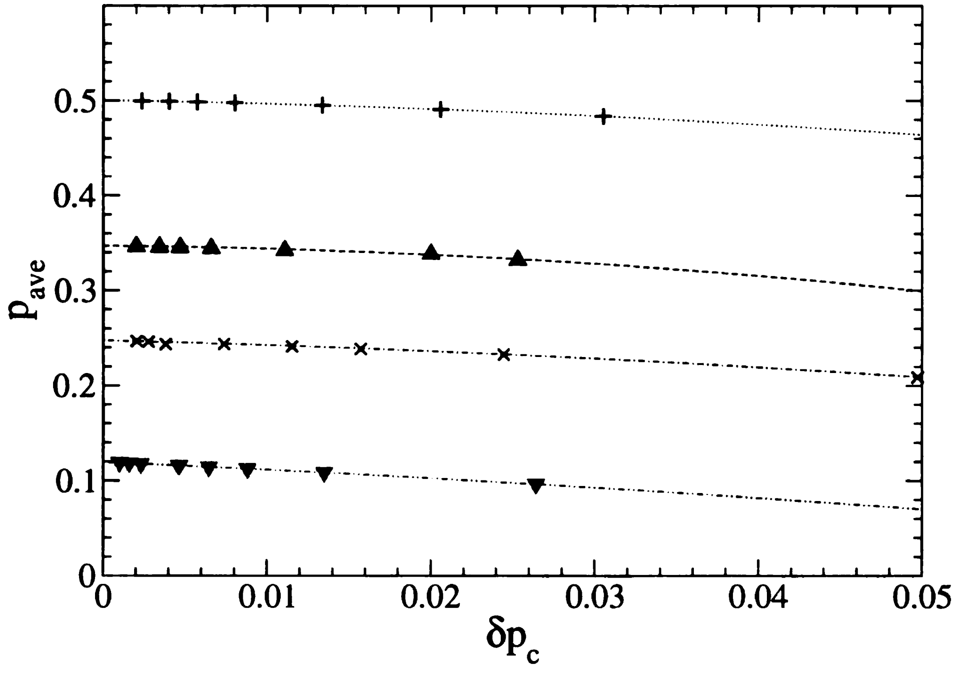


Figure A.2: Plots of the average value of p_l as a function of δp_l, for each sample size we find $p_l(L)$ and $\delta p_l(L)$. δp_l goes to 0 as L goes to ∞. This allows us to extract the infinite lattice critical point.

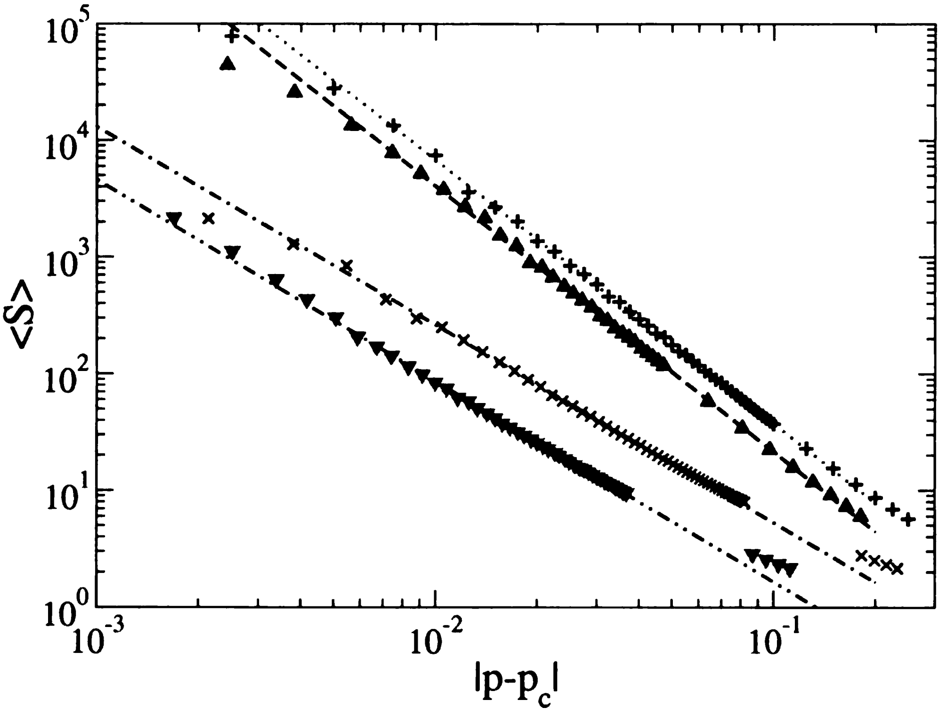
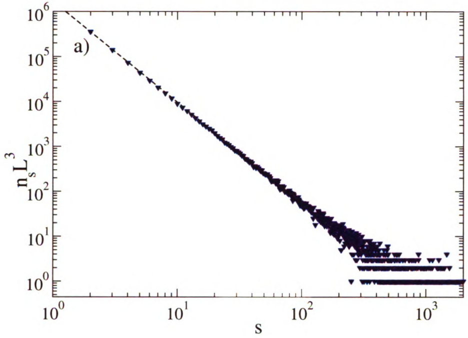


Figure A.3: Plots of the average cluster size on approach to p_l from below for lattices of size $N = 10,240,000$ sites, from this plot we extract the exponent γ. The $\triangle$ correspond to the triangular lattice yielding $\gamma = 2.28(7)$. The $+$ correspond to the square lattice yielding $\gamma = 2.26(7)$. The ∇ correspond to the FCC lattice yielding $\gamma = 1.76(8)$. The $\times$ correspond to the cubic lattice yielding $\gamma = 1.7(1)$.



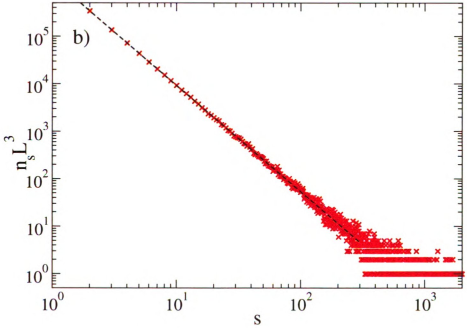


Figure A.4: plots of the cluster size distribution for lattices of $N = 8,000,000$ sites. (a) The FCC lattice yields $\tau = 2.24(2)$. (b) The cubic lattice yields $\tau = 2.24(1)$.

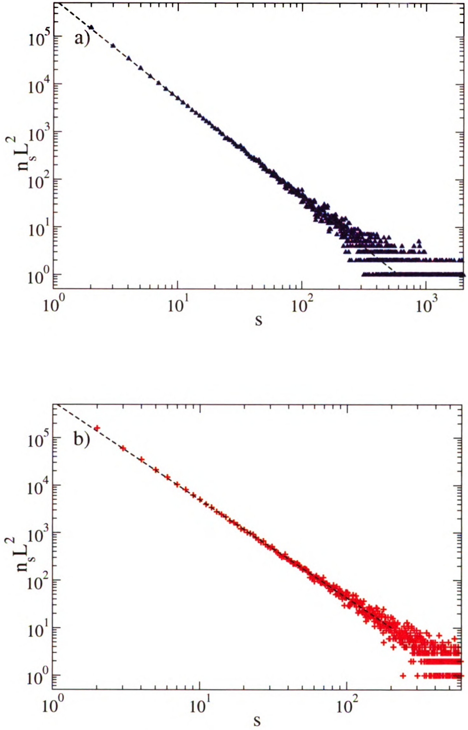


Figure A.5: plots of the cluster size distribution for lattices of $N = 10,240,000$ sites. (a) The triangular lattice yields $\tau = 2.08(2)$. (b) The square lattice yields $\tau = 2.06(1)$.

Appendix B

Replica Symmetric Calculations

This appendix contains an expansion of the math done in Sherrington's and Kirkpatrick's paper titled "Solvable Model of a Spin-Glass" [83] and the application of the replica method to the lattice gas [94].

B.1 Replica Method on the Ising Spin Glass [83]

We start with the Ising Model Hamiltonian with zero field, the system has N Ising spins, $S_i = \pm 1$,

$$H = -\frac{1}{2} \sum_{i \neq j} J_{ij} S_i S_j. \tag{B.1}$$

At this point we make n replica's of the partition Function, which we denote with the label α,

$$Z^n = \sum_{S_i^\alpha} \exp\left(\frac{\beta}{2} \sum_{i \neq j} \sum_{\alpha} J_{ij} S_i^\alpha S_j^\alpha\right). \tag{B.2}$$

The average over the disorder for Z^n is,

$$\langle Z^n \rangle = \int P(J_{ij}) Z^n dJ_{ij}. \tag{B.3}$$

We need the average value of the replicated Partition function in order to utilize the replica trick,

$$\langle \ln Z\rangle = \lim_{n\to 0}\frac{\langle Z^n\rangle - 1}{n}. \tag{B.4}$$

The form for the Probability density $P(J_{ij})$ for a specific J_{ij} is a Gaussian,

$$P(J_{ij}) = \frac{1}{J\sqrt{2\pi}}\exp(-\frac{(J_{ij}-J_0)^2}{2J^2}). \tag{B.5}$$

With the Product of the probability densities for all unique (ij) pairs giving,

$$\begin{aligned} P(J_{ij}) &= (\frac{1}{J\sqrt{2\pi}})^{(n(n-1)/2)}\exp(-\sum_{i\langle j}\frac{(J_{ij}-J_0)^2}{2J^2}), & \text{(B.6)} \\ &= (\frac{1}{J\sqrt{2\pi}})^{(n(n-1)/2)}\exp(-\frac{1}{2}\sum_{i\neq j}\frac{(J_{ij}-J_0)^2}{2J^2}). & \text{(B.7)} \end{aligned}$$

Combining (B.2),(B.3) and (B.7) we get,

$$\begin{aligned} \langle Z^n\rangle &= (J\sqrt{2\pi})^{-\frac{n^2}{2}}\int\sum_{S_i^\alpha}\prod_{ij}\exp(-\frac{J_0^2}{4J^2}) \\ &\quad \exp\left(-\frac{J_{ij}^2}{4J^2}+\left(\frac{\beta}{2}\sum_{\alpha}S_i^\alpha S_j^\alpha+\frac{J_0}{2J^2}\right)J_{ij}\right)dJ_{ij}. \end{aligned} \tag{B.8}$$

and using the Gaussian identity,

$$\int_{-\infty}^{\infty}\exp(-ax^2+bx)dx = \sqrt{\frac{\pi}{a}}\exp(\frac{b^2}{4a}), \tag{B.9}$$

we can evaluate $\langle Z^n\rangle$ to get our replicated partition function, and the free energy now looks like,

$$\langle Z^n\rangle = Z_R = \sum_{S_i^\alpha}\prod_{ij}\left[\exp\left(\frac{J^2}{4(kT)^2}(\sum_{\alpha}S_i^\alpha S_j^\alpha)^2+\frac{J_0}{2kT}\sum_{\alpha}S_i^\alpha S_j^\alpha\right)\right]. \tag{B.10}$$

Now, rearrange the sums in the exponential,

$$\sum_{i \neq j} (\sum_{\alpha} S_i^{\alpha} S_j^{\alpha})^2 = 2 \sum_{\alpha \langle \beta} (\sum_{i} S_i^{\alpha} S_i^{\beta})^2 . + N^2 n - N n^2 . \tag{B.11}$$

After this the partition function becomes,

$$\begin{aligned} Z_R &= \sum_{S_i^{\alpha}} \Big[\prod_{\alpha \langle \beta} \exp \Big[(\frac{J^2 N^2 n}{4(kT)^2}) - (\frac{J^2 n^2 N}{4(kT)^2}) + (\frac{J_0 N n}{2kT}) \\ &\quad + (\frac{J^2}{2(kT)^2}) (\sum_{i} S_i^{\alpha} S_i^{\beta})^2 + (\frac{J_0}{2kT}) (\sum_{i} S_i^{\alpha})^2 \Big] \Big] . \end{aligned} \tag{B.12}$$

The terms with N are dropped because they vanish in the thermodynamic limit to leave our partition function as,

$$\begin{aligned} Z_R &= \exp \Big(\frac{J^2 N^2 n}{4(kT)^2} \Big) \sum_{S_i^{\alpha}} \Big[\prod_{\alpha \langle \beta} \exp \Big(\frac{J^2}{2(kT)^2} (\sum_{i} S_i^{\alpha} S_i^{\beta})^2 \Big) \\ &\quad \prod_{\alpha} \exp \Big(\frac{J_0}{2kT} (\sum_{i} S_i^{\alpha})^2 \Big) \Big] . \end{aligned} \tag{B.13}$$

Which is the result arrived at by Sherrington and Kirkpatrick. Decompose this using Eq. (B.9), where for the $y_{\alpha\beta}$ integral $a = N/2$ and $b^2 = \frac{J^2 N}{(kT)^2} (\sum_i S_i^{\alpha} S_j^{\beta})^2$, and for the x_{α} integral , $a = N/2$ and $b^2 = \frac{J_0 N}{(kT)} (\sum_i S_i^{\alpha})^2$, to get,

$$\begin{aligned} Z_R &= \exp(\frac{\tilde{J}^2 N n}{4(kT)^2}) \sum_{S_i^{\alpha}} \int \Big[\prod_{\alpha \langle \beta} (\frac{N}{2\pi})^{1/2} dy_{\alpha\beta} \Big] \int \Big[\prod_{\alpha} (\frac{N}{2\pi})^{1/2} dx_{\alpha} \Big] \\ &\quad \Big(\exp(-N \sum_{\alpha \langle \beta} \frac{y_{\alpha\beta}^2}{2} + \frac{\tilde{J}}{kT} \sum_{\alpha \langle \beta} \sum_{i} S_i^{\alpha} S_i^{\beta} y_{\alpha\beta}) \\ &\quad + \exp(-N \sum_{\alpha} \frac{x_{\alpha}^2}{2} + (\frac{\tilde{J}_0}{kT})^{1/2} \sum_{\alpha} \sum_{i} S_i^{\alpha} x_{\alpha}) \Big) . \end{aligned} \tag{B.14}$$

Where $\tilde{J} = JN^{1/2}, \tilde{J}_0 = J_0 N$. At this point we choose to solve for the replica symmetric solution, which means setting all $y_{\alpha\beta} \rightarrow y$ and $x_\alpha \rightarrow x$. Now we make the statement,

$$\begin{aligned} Z_R &= \exp(\frac{\tilde{J}^2 N n}{4(kT)^2}) \int [\prod_{\alpha\langle\beta} (\frac{N}{2\pi})^{1/2} dy] \int [\prod_{\alpha} (\frac{N}{2\pi})^{1/2} \\ & \qquad dx] \exp\Big(- N\Big(\frac{(n^2 - n)y^2}{4} + \frac{nx^2}{2}\Big)\Big)(Z_1^n)^N. \qquad \text{(B.15)} \end{aligned}$$

Where Z_1^n is equal to,

$$\begin{aligned} Z_1^n &= \sum_{S^\alpha} \exp\left(\frac{\tilde{J}}{kT} \sum_{\alpha<\beta} S^\alpha S^\beta y + \sqrt{\frac{\tilde{J}_0}{kT}} \sum_{\alpha} S^\alpha x\right), \\ &= \sum_{S^\alpha} \exp\left(\frac{\tilde{J} y}{2kT} (\sum_{\alpha} S^\alpha)^2 - \frac{\tilde{J} y n}{2kT} + \sqrt{\frac{\tilde{J}_0}{kT}} \sum_{\alpha} S^\alpha x\right). \qquad \text{(B.16)} \end{aligned}$$

Then we use the Gaussian trick to decompose the this once again, to get,

$$\begin{aligned} Z_1^n &= \frac{\exp(-\frac{\tilde{J} y n}{2kT})}{\sqrt{2\pi}} \int \exp\big(- \frac{z^2}{2}\big) \\ & \qquad \prod_{\alpha} \left(\sum_{S^\alpha} \exp\left(\Big(\sqrt{\frac{\tilde{J} y}{kT}} z + \sqrt{\frac{\tilde{J}_0}{kT}} x\Big) \sum_{\alpha} S^\alpha \right) \right) dz. \qquad \text{(B.17)} \end{aligned}$$

Now we can finally sum over $S^\alpha = \pm 1$, this makes Z_1^n now,

$$Z_1^n = \frac{\exp(-\frac{\tilde{J} y n}{2kT})}{\sqrt{2\pi}} \int \exp(-\frac{z^2}{2}) \left(2\cosh\left(\sqrt{\frac{\tilde{J} y}{kT}} z + \sqrt{\frac{\tilde{J}_0}{kT}} x \right) \right)^n dz. \qquad \text{(B.18)}$$

Insert this into our Replicated Partition function and get,

$$\begin{aligned} Z_R &= \int [\prod_{\alpha\langle\beta}(\frac{N}{2\pi})^{1/2} dy] \int [\prod_{\alpha}(\frac{N}{2\pi})^{1/2} dx] \exp(\frac{\tilde{J}^2 Nn}{4(kT)^2}) \exp(-\frac{\tilde{J}yNn}{2kT}) \\ & \exp(-N(\frac{(n^2-n)y^2}{4} + \frac{nx^2}{2})) + \exp\Big[N \ln(2\pi)^{-1/2} \\ & \int dz \exp(-\frac{z^2}{2}) \Big(2\cosh\big(\sqrt{\frac{\tilde{J}y}{kT}} z + \sqrt{\frac{\tilde{J}_0}{kT}} x\big)\Big)^n\Big]. \end{aligned} \tag{B.19}$$

Now, evaluate the integrals over x and y by the method of steepest descent. But first let us define $y \rightarrow (\frac{\tilde{J}}{kT})q$ and $x \rightarrow m(\frac{\tilde{J}_0}{kT})^{1/2}$ this gives,

$$\begin{aligned} Z_R &= \int \Big[\prod_{\alpha\langle\beta}(\frac{N}{2\pi})^{1/2} \frac{\tilde{J}^2}{(kT)^2} dq\Big] \int \Big[\prod_{\alpha}(\frac{N\tilde{J}_0}{2\pi kT})^{1/2} dm\Big] \\ & \exp\Big(-(\frac{\tilde{J}^2 Nn}{4(kT)^2}(1-q)^2 + \frac{\tilde{J}_0 Nn}{2kT} m^2)\Big) \exp\Big(\frac{\tilde{J}^2 Nn^2}{4(kT)^2} q^2\Big) \\ & + \exp\Big[N \ln(2\pi)^{-1/2} \int dz e^{-\frac{z^2}{2}} \Big(2\cosh\big(\frac{\tilde{J}}{kT} q^{1/2} z + \frac{\tilde{J}_0}{kT} m\big)\Big)^n\Big] \end{aligned} \tag{B.20}$$

$$\begin{aligned} F &= -kT \lim_{n\rightarrow 0} n^{-1} \Big[\int \Big[\prod_{\alpha\langle\beta}(\frac{N}{2\pi})^{1/2} \frac{\tilde{J}^2}{(kT)^2} dq\Big] \int \Big[\prod_{\alpha}(\frac{N\tilde{J}_0}{2\pi kT})^{1/2} dm\Big] \\ & \exp\Big(-(\frac{\tilde{J}^2 Nn}{4(kT)^2}(1-q)^2 + \frac{\tilde{J}_0 Nn}{2kT} m^2)\Big) \exp\Big(\frac{\tilde{J}^2 Nn^2}{4(kT)^2} q^2\Big) \\ & + \exp\Big[N \ln(2\pi)^{-1/2} \int dz \exp(-\frac{z^2}{2}) \\ & \Big(2\cosh\big(\frac{\tilde{J}}{kT} q^{1/2} z + \frac{\tilde{J}_0}{kT} m\big)\Big)^n\Big] - 1\Big]. \end{aligned} \tag{B.21}$$

Now we do the integrals by the method of steepest descent to get the coupled equations,

$$
\begin{aligned}
-\frac{N}{2}(\frac{\tilde{J}}{kT})^2(1-q) + N\frac{\partial \ln Z_1^{n'}}{\partial q} &= 0, \qquad \text{(B.22)} \\
-N(\frac{\tilde{J}_0}{kT})m + N\frac{\partial \ln Z_1^{n'}}{\partial m} &= 0. \qquad \text{(B.23)}
\end{aligned}
$$

where we define,

$$
Z_1^{n'} = (2\pi)^{-1/2}\int dz \exp(-\frac{z^2}{2})\left(2\cosh\left(\frac{\tilde{J}}{kT}q^{1/2}z + \frac{\tilde{J}_0}{kT}m\right)\right)^n. \tag{B.24}
$$

Then

$$
\begin{aligned}
q &= 1 - \frac{1}{\sqrt{2\pi}}\int dz \exp(-z^2/2)\operatorname{sech}^2\left(\frac{\tilde{J}}{kT}q^{1/2}z + \frac{\tilde{J}_0}{kT}m\right), \qquad \text{(B.25)} \\
m &= \frac{1}{\sqrt{2\pi}}\int dz \exp(-z^2/2)\tanh\left(\frac{\tilde{J}}{kT}q^{1/2}z + \frac{\tilde{J}_0}{kT}m\right). \qquad \text{(B.26)}
\end{aligned}
$$

The free energy becomes,

$$
\begin{aligned}
F = -kT\Bigg[&- \frac{\tilde{J}^2 N}{4(kT)^2}(1-q)^2 - \frac{\tilde{J}_0 N}{2kT}m^2) \\
&+ N\ln(2\pi)^{-1/2}\int dz \exp(-z^2/2) \\
&2\cosh\left(\frac{\tilde{J}}{kT}q^{1/2}z + \frac{\tilde{J}_0}{kT}m\right)\Bigg]. \qquad \text{(B.27)}
\end{aligned}
$$

and the entropy is,

$$
\begin{aligned}
\frac{S}{N} = &-\left[\frac{\tilde{J}_0 m^2}{2T} + (\frac{\tilde{J}^2}{4kT^2})(1-q)(1+3q) - \frac{Hm}{T}\right] \\
&+ k(2\pi)^{-1/2}\int dz \exp(-\frac{1}{2})\ln(2\cosh Z). \qquad \text{(B.28)}
\end{aligned}
$$

The entropy will be nonphysical when $T \to 0, H = 0, m = 0$.

B.2 Replica solution to the lattice gas [94]

We define an indicator function,

$$\chi = \prod_{(ij)} (1 - y_i y_j) \tag{B.29}$$

and a partition function,

$$\Xi = \sum_{y_i=0,1} \chi \exp(\mu \sum_i y_i). \tag{B.30}$$

The average density at a site is,

$$\rho = \frac{1}{\Xi}\frac{\partial \Xi}{\partial \mu} |_{\mu \to \infty} \tag{B.31}$$

Using the identity,

$$\frac{1}{\Xi} = \lim_{n \to 0} \Xi^{n-1} \tag{B.32}$$

we shall find the average value of ρ, by averaging Eq. (B.31) using (B.32). The average replicated partition function is given by,

$$\overline{\Xi^n} = \sum_{y_i^a=0,1} \exp(\mu \sum_{i,a} y_i^a)\Big(\prod_{i \neq j}[(1 - \frac{c}{N}) + \frac{c}{N}\prod_a (1 - y_i^a y_j^a)]\Big) \tag{B.33}$$

where $a = 1, ..., n$ is the replica index. Since we are interested in $c \approx 1$, we can exponentiate to find,

$$\overline{\Xi^n} = \sum_{y_i^a=0,1} \exp(\mu \sum_{i,a} y_i^a - (cN/2) + (c/N) \sum_{i \neq j} \prod_a (1 - y_i^a y_j^a)) \tag{B.34}$$

Now introduce the variable,

$$x(\vec{y}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{a}\delta_{y_i^a, y_i} \tag{B.35}$$

with

$$\sum_{y} x(\vec{y}) = 1. \tag{B.36}$$

Notice that we have defined a special notation for $\vec{y}_i = \{y_i^1, ...y_i^a, ..., y_i^n\}$. It is a vector of replicated variables at a single site and thus has n components, each of which can take on the values 0 or 1. The order parameter $x(\vec{y})$ is the fraction of sites which are in configuration $\vec{y}$. It is a functional order parameter as each of the 2^n components of x can take on a continuum of values between 0 and 1. Using this order parameter we rewrite the partition function as,

$$\overline{\Xi^n} = \int_0^1 \prod_{y^a=0,1}^{2^n} dx(y^a)\exp(-NF[x(\vec{y})]). \tag{B.37}$$

where

$$\begin{aligned} F[x(\vec{y})] &= \frac{c}{2} - \mu\sum_{\vec{y}} x(\vec{y})(\vec{y}\cdot\vec{1}) - \frac{c}{2}\sum_{\vec{y}\vec{y}'} x(\vec{y})x(\vec{y}')\prod_{a}(1 - y^a y^{a'}) \\ &\quad + \sum_{\vec{y}} x(\vec{y})\ln x(\vec{y}) + \lambda(\sum_{\vec{y}} x(\vec{y}) - 1). \end{aligned} \tag{B.38}$$

In this expression $\vec{y}\cdot\vec{1} = \sum_{a=1}^{n} y^a$. Eq. (B.38) includes the combinatorial factor $N!/\prod_{\vec{y}}[x(\vec{y})N]!$ due to the permutations of $x(\vec{y})$. We now use the method of steepest descents with $x(\vec{y})$ being our set of continuous variables. We have,

$$\frac{\partial F}{\partial x} = -\mu\vec{y}\cdot\vec{1} - c\sum_{\vec{y}'} x(\vec{y}')\prod_{a}(1 - y^a y^{a'}) + \ln x(\vec{y}) + 1 + \lambda = 0 \tag{B.39}$$

The symmetric ansatz assumes that x is invariant under permutations of the components of $\vec{y}$. This implies that

$$x(\vec{y}) \to x(\sum_{a=1}^{N} y^a) \tag{B.40}$$

It is useful to define the exponential transform,

$$x(\vec{y}) = \int_{-\infty}^{\infty} dh g(h) \frac{\exp(h \sum_{a=1}^{N} y^a)}{(1+e^h)^n}. \tag{B.41}$$

which ensures that the constraint (B.36) holds, provided,

$$\int_{-\infty}^{\infty} g(h) dh = 1. \tag{B.42}$$

Within the symmetric assumption, the average density ρ is the same for all replicas, so we have,

$$\rho = \langle y_1 \rangle = \sum_{\vec{y}} x(\vec{y}) y_1 = \int_{-\infty}^{\infty} dh g(h) \frac{\exp(h)}{1+\exp(h)}. \tag{B.43}$$

Defining $s = \sum_a y^a$ and rearranging equation (B.39) yields,

$$x(\vec{y}) = \exp[-\lambda - 1 + \mu s + c \sum_{\vec{y}'} x(\vec{y}') \prod_{a'} (1 - y^a y^{a'})] \tag{B.44}$$

Using Eq. (B.41), we have,

$$\begin{aligned} \int dh g(h) \frac{\exp(hs)}{(1+\exp(h))^n} &= \exp\Bigg[-\lambda + 1 + \mu s + c \int dh' g(h') \\ &\qquad \exp\Big(-s\ln\big(1+\exp(h')\big)\Big)\Bigg] \end{aligned} \tag{B.45}$$

The only puzzling piece of this expression is the last term in the exponential on the RHS. To see how this arises start with the last term on the RHS of Eq. (B.44) (the one proportional to c),

$$\begin{aligned}
&\sum_{\vec{y}'} \int dh' g(h') \frac{\exp(h's')}{(1+\exp(h'))^n} \prod_{a'} (1 - y^a y^{a'}) \\
&= \int dh' g(h') \frac{1}{(1+\exp(h'))^n} \prod_{a'} \sum_{y^{a'}=0,1} \exp(h' y^{a'})(1 - y^a y^{a'})
\end{aligned} \tag{B.46}$$

Each term in the product can be summed to give $1 + (1 - y^a)e^{h'}$. This term is one if $y^a = 1$, and it is $1 + e^{h'}$ if $y^a = 0$. Using this information, Eq. (B.46) reduces to,

$$= \int dh' g(h') \frac{(1+\exp(h'))^{n-s}}{(1+\exp(h'))^n}, \tag{B.47}$$

which is the same as the last term in the exponential on the RHS of Eq. (B.45). Now we take the limit $n \to 0$ to find,

$$\int dh g(h) e^{hs} = \exp[-\lambda + 1 + \mu s + c \int dh' g(h') \exp(-s\ln(1+\exp(h')))])] \tag{B.48}$$

Note that if $s = 0$, we must have, $c - \lambda + 1 = 0$, which implies that,

$$\lambda = c + 1 \tag{B.49}$$

It is useful to make the following changes of variable,

$$k = h/\mu \quad ; \quad v = s\mu \tag{B.50}$$

We then have,

$$\mu \int_0^\infty dk g(k\mu) \exp(kv) = \exp(v - c + cz(v)) \tag{B.51}$$

where

$$z(v) = \mu \int dk' g(k'\mu) \exp(-\frac{v}{\mu} \ln[1 + \exp(k'\mu)]) \tag{B.52}$$

Now note that in the limit $\mu \to \infty$ we have,

$$\frac{1}{\mu}\ln(1+\exp(k'\mu)) \to k'\theta(k') \tag{B.53}$$

This implies that,

$$\begin{aligned} z(v) \to \mu \int_{-\infty}^{\infty} dk' g(k'\mu)\exp(-k'v\theta(k')) = \\ \mu \int_{0+}^{\infty} dk' g(k'\mu)\exp(-k'v) + \mu \int_{-\infty}^{0^+} dk' g(k'\mu). \end{aligned} \tag{B.54}$$

Now we assume that,

$$g(k\mu) = \frac{1}{\mu}\sum_{m=-1}^{\infty} g_m \delta(k+m). \tag{B.55}$$

Substituting into Eq. (B.51) (and using Eq. (B.54)) we have,

$$\sum_{m=-1}^{\infty} g_m \exp(-mv) = \exp(v - c + c[g_{-1}\exp(-v) + g_0 + \sum_{m=1}^{\infty} g_m]) \tag{B.56}$$

From the normalization condition on $g(h)$ (Eq. (B.42)), we obtain,

$$\int_{-\infty}^{\infty} g(h)dh = 1 \quad \Rightarrow \quad g_{-1} + g_0 + \sum_{m=1}^{\infty} g_m = 1 \tag{B.57}$$

Using this result and expanding the g_{-1} term on the RHS of Eq. (B.56) yields,

$$\sum_{m=-1}^{\infty} g_m \exp(-mv) = \exp(-cg_{-1}) \sum_{l=0}^{\infty} \frac{(cg_{-1})^l \exp((1-l)v)}{l!}. \tag{B.58}$$

To ensure that this equation holds, we need to have,

$$g_m = \exp(-cg_{-1})\frac{(cg_{-1})^{m+1}}{(m+1)!}. \tag{B.59}$$

Once we know g_{-1} then all of the other g_m are found from it. The equation for g_{-1} is,

$$g_{-1} = \exp(-cg_{-1}). \tag{B.60}$$

We define $W(c) = cg_{-1}$, so we have,

$$c = W\exp(W). \tag{B.61}$$

This is the definition of the Lambert function. We can now find the average density on each site of the lattice. The density is given by (see Eq. (B.43))

$$\begin{aligned} \rho &= \frac{1}{\mu}\int_0^\infty dk g(k\mu)\frac{\exp(k\mu)}{1+\exp(k\mu)}, \\ &= \int_0^\infty dk \sum_{m=-1}^{\infty} g_m\delta(k+m)\frac{\exp(k\mu)}{1+\exp(k\mu)}, \\ &= g_{-1}\frac{\exp(\mu)}{1+\exp(\mu)} + \frac{1}{2}g_0 \to g_{-1} + \frac{1}{2}g_0. \end{aligned} \tag{B.62}$$

where the last expression on the RHS applies as we are interested in the limit $\mu \to \infty$. From Eq. (B.59), we have,

$$g_0 = \exp(-cg_{-1})cg_{-1} = c(g_{-1})^2 = \frac{W^2}{c}. \tag{B.63}$$

The density of the vertex cover is $x_c = 1 - \rho$ which is ,

$$x_c = 1 - \frac{2W + W^2}{2c}. \tag{B.64}$$

There is more information in the solution above, than just the average cover. Since the maximum independent set, or the repulsive lattice gas, on random graphs is highly degenerate, we may define three types of sites: Those that are always uncovered (fraction ρ_u); those that are always covered (the backbone, fraction ρ_b)

and those that are sometimes covered (ρ_e, these contribute to the entropy). That is, for a given random graph, all of the degeneracy occurs due to rearrangements of the atoms on the sites with volume fraction (ρ_e). The quantities ρ_u, ρ_b and ρ_e are related to the solution we have found in the following way:

- The fraction which is always covered, $\rho_b = g_{-1} = W/c$.
- The fraction which is sometimes covered, $\rho_e = g_0 = W^2/c$.
- The fraction which is never covered, $\rho_u = 1 - \rho_b - \rho_e = 1 - (W + W^2)/c$.

BIBLIOGRAPHY

Bibliography

[1] D.J. Amit, H. Gutfreund, and H. Sompolinsky. Spin-glass models of neural networks. *Physical Review A*, 32:1007–1008, 1985.

[2] Wolfgang Barthel and Alexander K. Hartmann. Clustering analysis of the ground-state structure of the vertex-cover problem. *Physical Review E*, 70:066120, 2004.

[3] Sorin Bastea and Philip M. Duxbury. Ground state structure of random magnets. *Physical Review E*, 58:4261, 1998.

[4] M. Bauer and O. Golinelli. Core percolation in random graphs: a critical phenomena analysis. *Euro. Phys. Jour. B*, 24:339–352, 2002.

[5] M. Bauer and O. Golinelli. Random incidence matrices: moments of the spectral density. *Euro. Phys. Jour. B*, 24:339–352, 2002.

[6] Béla Bollobás. The evolution of random graphs. *Transactions of the American Mathematical Society*, 286(1):257–274, November 1984.

[7] J. P. Bouchard and M. Mézard. Self induced quenched disorder: a model for the glass transistion. *Journal of Physique I*, 4:1109, 1994.

[8] Anton Bovier and Pierre Picco, editors. *Mathematical Aspects of Spin Glasses and Neural Networks*, volume 41 of *Progress in Probability*. Birkhäuser, 1998.

[9] A.J. Bray and M.A. Moore. Metastable states in spin glasses. *Journal of Physics C*, 13:L469–L476, 1980.

[10] Ron Breukelaar, Erik D. Demaine, Susan Hohenberger, and David Liben-Nowell. Tetris is hard, even to approximate. *International Journal of Computational Geometry and Applications*, 14(1-2):41–68, 2004.

[11] Joseph D. Bryngelson and Peter G. Wolynes. Spin glasses and the statistical mechanics of protein folding. *Proceedings of the National Academy of Sciences of the United States of America*, 84(21):7524–7528, Nov 1987.

[12] J. Chayes. Finite-size scaling in percolation. *Documenta Mathematica*, Extra Volume ICM III:113–122, 1998.

[13] Debashish Chowdhury. *Spin glasses and other frustrated systems.* Princeton university press, 1986.

[14] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM Press.

[15] L. Correale, M. Loeone, A. Pagnani, M. Weigt, and R. Zecchina. Core percolation and onset of complexity in boolean networks. *Physical Review Letters*, 95:018101, 2006.

[16] J.R.L. de Almida and D. J. Thouless. Stability of the sherrington-kirkpatrick solution of a spin glass model. *Journal of Physics A*, 11(5):983–990, 1978.

[17] Erik D. Demaine, Susan Hohenberger, and David Liben-Nowell. Tetris is hard, even to approximate. In *COCOON*, pages 351–363, 2003.

[18] G. Deutscher, R. Zallen, and Joan Adler, editors. *Percolation Structures and Processes: Annals of the Isreal Physical Society.* Adam Hilger and The Isreal Physical Society, 1983.

[19] S.F. Edwards and P.W. Anderson. Theory of spin glasses. *Journal of Physics F*, 5:965–974, May 1975.

[20] P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1961.

[21] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.

[22] K.H. Fischer and J.A. Hertz. *Spin glasses.* Cambridge University Press, 1991.

[23] Michael E. Fisher and Michael N. Barber. Scaling theory for finite-size effects in the critical region. *Physical Review Letters*, 28(23):1516–1519, June 1972.

[24] Aviezri S. Fraenkel. Protein folding, spin glass and computational complexity. In *Proceedings of the 3rd DIMACS Workshop on DNA Based Computers, held at the University of Pennsylvania, June 23 – 25, 1997*, pages 175–191, 1997.

[25] S. Franz, M. Mézard, F. Ricci-Tersenghi, M. Weigt, and R. Zecchina. A ferromagnet with a glass transition. *European Physics Letters*, 55:465, 2001.

[26] Y.T. Fu and P.W. Anderson. Application of statistical mechanics to np-complete problems in combinatorial optimization. *Journal of Physics A*, 19:1605–1620, 1986.

[27] E. Gardner. The space on interactions in neural network models. *Journal of Physics A*, 21:257–270, 1988.

[28] E. Gardner and B. Derrida. Optimal storage properties of neural network models. *Journal of Physics A*, 21:271–284, 1988.

[29] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[30] A. Glaser, A. C. Jones, and P. M. Duxbury. Domain states in the zero-temperature diluted antiferromagnet in an applied field. *Physical Review B (Condensed Matter and Materials Physics)*, 71(17):174423, 2005.

[31] Richard A. Goldstein, Zaida A. Luthey-Schulten, and Peter G. Wolynes. Optimal protein-folding codes from spin-glass theory. *Proceedings of the National Academy of Sciences of the United States of America*, 89(11):4918–4922, June 1992.

[32] Uwe Gropengiesser and Dietrich Stauffer. Limited universality at the percolation threshold in 2 to 6 dimensions. *Physica A*, 210:320–325, 1994.

[33] Alexander K. Hartmann and Martin Weight. Statistical mechanics perspective on the phase transition in vertex covering of finite-connectivity random graphs. *Theo. Comp. Sci.*, 265:199, 2001.

[34] Alexander K. Hartmann and Martin Weigt. Statistical mechanics of the vertex-cover problem. *MATH.GEN.*, 36:11069, 2003.

[35] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. National Academy of Science USA*, 79:2554–2558, 1982.

[36] D.E. Moneton P.W. Stephens R.J. Bergeneau P.M. Horn and G.S. Brown. Synchrotron x-ray study of the commensurate-incommensurate transition of monolayer krypton on graphite. *Phys. Rev. Lett.*, 46(23):1533, 1981.

[37] C. W. Fay IV and P. M. Duxbury. Core percolation by leaf removal on diluted lattices. *in preparation*, 2007.

[38] C. W. Fay IV and P. M. Duxbury. Core percolation by triangle removal on diluted lattices. *in preparation*, 2007.

[39] C. W. Fay IV and P. M. Duxbury. Finding the maximum independent set on diluted regular lattices. *in preparation*, 2007.

[40] C. W. Fay IV, J. W. Liu, and P. M. Duxbury. Maximum independent set on diluted triangular lattices. *Physical Review E*, 73(056112):1–14, 2006.

[41] B. Joos, B. Bergersen, and M.L. Klein. Ground-state properties of xenon on graphite. *Phys. Rev. B*, 28(12):7219, 1983.

[42] R.J. Gooding B. Joos and B. Bergersen. Krypton on graphite: Microstructure at zero temperature. *Phys. Rev. B*, 27(12):7669, 1983.

[43] R. Kariotis, J.A. Venables, M. Hamichi, and A.Q.D. Faisal. Xenon-graphite interaction near the commensurate-incommenserate transition. *J. Phys. C*, 19:5717–5726, 1987.

[44] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[45] Richard M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[46] Richard M. Karp and Micheal Sipser. Maximum matchings in sparse random graphs. In *Proc. 22nd Annual Symp. on Foundations of Computer Science*, pages 364–375, 1981.

[47] Richard Kaye. Mastermind is np-complete. *Mathematical Intelligencer*, 22(2):9–15, 2000.

[48] J. Keller, P. Miltényi, B. Beschoten, G. Güntherodt, U. Nowak, and K. D. Usadel. Domain state model for exchange bias. ii. experiments. *Phys. Rev. B*, 66(1):014431, Jul 2002.

[49] S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264(5163):1297–1301, May 1994.

[50] S. Kirkpatrick and D. Sherrington. Infinite-ranged models of spin-glasses. *Physical Review B*, 17(11):4384–4403, June 1978.

[51] Michele Leone, Federico Ricci-tersenghi, and Riccardo Zecchina. Phase coexistence and finite-size scaling in random combinatorial problems. *Journal of Physics A*, 34:4615, 2001.

[52] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[53] Olivier C. Martin, Rémi Monasson, and Riccardo Zecchina. Statistical mechanics methods and phase transitions in optimization problems. *Theoretical Computer Science*, 265(1–2):3–67, 2001.

[54] M. Mézard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94:197205, 2005.

[55] M. Mézard, J.P. Nadal, and G. Toulouse. Solvable models of working memories. *Journal of Physics*, 1986.

[56] M. Mézard and G. Parisi. A replica analysis of the travelling salesman problem. *Journal of Physics*, 47:1285–1296, 1986.

[57] M. Mézard, G. Parisi, N Sourlas, G. Toulouse, and M. Virasoro. Nature of the spin-glass phase. *Physical Review Letters*, 52(13):1156–1159, March 1984.

[58] M. Mézard, G. Parisi, N. Sourlas, G. Toulouse, and M. Virasoro. Replica symmetry breaking and the nature of the spin glass phase. *Journal of Physique*, 45:843–854, May 1984.

[59] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiablity problems. *Science*, 297:812–815, 2002.

[60] M. Mézard and M.A. Virasoro. The microstructure of ultrametricity. *Journal of Physique*, 46:1293–1307, 1985.

[61] Marc Mézard and Riccardo Zecchina. The random k-satisfiability problem: from an analytic solution to an efficient algorithm. *Physical Review E*, 66:056126, 2002.

[62] Mark Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin Glass Theory and Beyond*. World Scientific, 1987.

[63] Rémi Monasson. Some remarks on hierarchical replica symmetry breaking in finite-connectivity systems. Minerva Workshop on Neural Networks, March 1997.

[64] Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. Determining computational complexity from characteristic 'phase transistions'. *Nature*, 297:812–815, 1999.

[65] P.A. Heiney R.J. Birgeneau G.S. Brown P.M. Horn D.E. Moneton and P.W. Stephens. Freezing transition of monolayer xenon on graphite. *Phys. Rev. Lett.*, 48(2):104, 1982.

[66] C. Moukarzel and P. M. Duxbury. Comparison of rigidity and connectivity percolation in two dimensions. *Phys. Rev. E*, 59(3):2614–2622, March 1999.

[67] C. M. Newman and D. L. Stein. Simplicity of state and overlap structure in finite-volume realistic spin glasses. *Physical Review E*, 57(2):1356–1366, February 1998.

[68] C. M. Newman and D. L. Stein. Meetastable states in spin glasses and disorder ferromagnets. *Physical Review E*, 60(5):5244–5260, November 1999.

[69] Mario Nicodemi, Antonio Coniglio, and Hans J. Hermann. Frustration and slow dynamics of granular packings. *Phys. Rev. E*, 55(4):3962, 1997.

[70] M. Nielsen, J. Als-Neilsen, and J. Bohr. Pressure-driven commensurate-incommensurate transistion in low-temperature submonolayer krypton on graphite. *Phys. Rev. Lett.*, 47(8):582, 1983.

[71] U. Nowak, K. D. Usadel, J. Keller, P. Miltényi, B. Beschoten, and G. Güntherodt. Domain state model for exchange bias. i. theory. *Phys. Rev. B*, 66(1):014430, Jul 2002.

[72] W.J. Nuttall, K.P. Fahey, M.J. Young, B. Keimer, R.J. Birgeneau, and H. Suematsu. A sychrotron x-ray diffraction study of the structural phase behavior of multilayer xenon on single-crystal graphite. *J. Phys. Conds. Matter*, 5:8159–8176, 1993.

[73] N. Parga and M.A. Virasoro. The ultrametric organisation of memories in a neural network. *Journal of Physics*, 47:1857–1864, 1986.

[74] G. Parisi. Infinite number of order parameters for spin-glasses. *Physical Review Letters*, 43(23):1754–1756, 1979.

[75] G. Parisi. The order parameter for spin glasses: A function on the interval 0-1. *Journal of Physics A*, 13:1101–1112, 1980.

[76] G. Parisi. Order parameter for spin-glasses. *Physical Review Letters*, 50(24):1946–1948, 1983.

[77] G. Parisi and F. Zamponi. The ideal glass transition of hard spheres. *J.CHEM.PHYS.*, 123:144501, 2005.

[78] Giorgio Parisi. Mean field theory of spin glasses: statics and dynamics. *arXiv:cond-mat*, (0706.0094), 2007.

[79] A P Ramirez. Strongly geometrically frustrated magnets. *Annual Review of Materials Science*, 24(1):453–480, 1994.

[80] S. Suresh Rao and Somendra M. Bhattacharjee. Protein folding and spin glass. *Physica A*, 224:279–286, 1996.

[81] Francesco M Russo. Lattice gas analogue of the sherrington-kirkpatrick model: a paradigm for the glass transition. *Journal of Physics A: Mathematical and General*, 31(35):7249–7264, 1998.

[82] Francesco M. Russo. On lattice gas models for disordered systems. *Physics Letters A*, 239:17–20, February 1998.

[83] D. Sherrington and S. Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792–1796, December 1975.

[84] D. Stauffer and A. Aharony. *Introduction to Percolation Theory*. Taylor & Francis Ltd., revised second edition edition, 1994.

[85] Dietrich Stauffer, Joan Adler, and Amnon Aharony. Universality at the three-dimensional percolation threshold. *Journal of Physics A*, 27:L475–L480, 1994.

[86] Jeff Stuckman and Guo-Qiang Zhang. Mastermind is np-complete. *INFOCOMP Journal of Computer Science*, 5(2):25–28, 2006.

[87] Robert Endre Tarjan and Anthony E. Trojanowski. Finding a maximum indepentdent set. *Siam J. Comput.*, 6(3):537–546, 1977.

[88] D.J. Thouless, P.W. Anderson, and R.G. Palmer. Solution of solvable model of a spin glass. *Philosophical Magazine*, 35(3):593–601, 1977.

[89] G. Toulouse. Theory of the frustration effect in spin glasses: I. *Communications on Physics*, 2:115–119, 1977.

[90] G. Toulouse, S. Dahaene, and J.-P changeux. Spin glass model of learning by selection. *Proc. National Academy of Science USA*, 83:1695–1698, 1986.

[91] Martin Weight and Alexander K. Hartmann. Typical solution time for a vertex-covering algorithm on finite-connectivity random graphs. *Phys. Rev. Lett.*, 86:1658, 2001.

[92] M. Weigt and A. K. Hartmann. Glassy behavior induced by geometrical frustration in a hard-core lattice gas model. *Europhysics Letters*, 62:533, 2003.

[93] Martin Weigt. Dynamics of heuristic optimization algorithms on random graphs. *Euro. Phys. Jour. B*, 28:369–381, 2002.

[94] Martin Weigt and Alexander K. Hartmann. Minimal vertex covers on finite-connectivity random graphs - a hard-sphere lattice-gas picture. *Phys. Rev. E*, 63(056127), 2001.

[95] Gerhard J. Woeginger. Exact algorithms for np-hard problems: a survey. In *Combinatorial optimization - eureka, you shrink!*, pages 185–207. Springer-Verlag New York, Inc., New York, NY, USA, 2003.

[96] Takayuki Yato. Complexity and completeness of finding another solution and its application to puzzles. Master's thesis, The University of Tokyo, January 2003.