Pro gradu -tutkielma
Teoreettinen fysiikka

Spin Glasses, Boolean Satisfiability, and Survey Propagation

Daniel Landau
2013

Ohjaaja: Juha Honkonen
Tarkastajat: Juha Honkonen
Kai Nordlund

HELSINGIN YLIOPISTO
FYSIIKAN LAITOS

PL 64 (Gustaf Hällströmin katu 2)
00014 Helsingin yliopisto

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Physics

Tekijä — Författare — Author
Daniel Landau

Työn nimi — Arbetets titel — Title

Spin glasses, Boolean Satisfiability, and Survey Propagation

Oppiaine — Läroämne — Subject
Theoretical physics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master of Science Thesis	08/2013	51

Tiivistelmä — Referat — Abstract

In recent years statistical physics and computational complexity have found mutually interesting subjects of research. The theory of spin glasses from statistical physics has been succesfully applied to the boolean satisfiability problem, which is the canonical topic of computational complexity.

The study of spin glasses originated from experimental studies of the magnetic properties of impure metallic alloys, but soon the study of the theoretical models outshone the interest in the experimental systems. The model studied in this thesis is that of Ising spins with random interactions. In this thesis we discuss two analytical derivations on spin glasses: the famous replica trick on the Sherrington-Kirkpatrick model and the cavity method on a Bethe lattice spin glass.

Computational complexity theory is a branch of theoretical computer science that studies how the running time of algorithms scales with the size of the input. Two important classes of algorithms or problems are P and NP, or colloquially easy and hard problems. The first problem to be proven to belong to the class of NP-complete problems is that of boolean satisfiability, i.e., the study of whether there is an assignment of variables for a random boolean formula so that the formula is satisfied. The boolean satisfiability problem can be tackled with spin glass theory; the cavity method can be applied to it.

Boolean satisfiability exhibits a phase transition. As one increases the ratio of constraints to variables the probability of a random formula being satisfiable drops from unity to zero. This transition of random formulas from satisfiable to unsatisfiable is continuous for small formulas. It grows sharper with increasing problem size and becomes discrete at the limit of an infinite number of variables. The cavity method gives a value for the location of the phase transition that is in agreement with the numerical value.

The cavity method is an analytical tool for studying average values over a distribution, but it introduces so called surveys that can also be calculated numerically for a single instance. These surveys inspire the survey propagation algorithm that is implemented as a numerical program to efficiently solve large instances of random boolean satisfiability problems.

In this thesis I present a parallel version of survey propagation that achieves a speedup by a factor of 3 with 4 processors. With the improved version we are able to gain further knowledge on the detailed workings of survey propagation. It is found, firstly, that the number of iterations needed for one convergence of survey propagation depends on the number of variables, seemingly as $\ln N$. Secondly, it is found that the constraint to variable ratio for which survey propagation succeeds is dependent on the number of variables.

Avainsanat — Nyckelord — Keywords
Spin glasses, survey propagation, satisfiability

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpula Campus Library, Gustaf Hällströmin katu 2 (PO Box 64), 00014 University of Helsinki

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

This thesis studies the connection between statistical physics and computational complexity theory, a branch of theoretical computer science. Physics has always benefited from the practical side of computer science in terms of algorithms and advances in numerical simulations and data analysis. Theoretical computer science is less concerned with coding and more similar to mathematics, so the link to physics is not obvious. The connection between the theoretical computer science and statistical physics has recently become deeper, as theoretical analyses and insights have unified models from both sides.

In statistical physics the subject of interest are collections of microscopic objects that follow certain rules. The objective is to find macroscopic properties that abstract away the microscopic details. The most familiar case is point masses interacting with Newtonian forces where statistical physics recreates the usual thermodynamics (see e.g. [1]). In this thesis we are more interested in the famous Ising spins (i.e. spins that can point only up or down, or equivalently have only values in ± 1) interacting in a way that tends to align two spins to point in either the same or opposite directions. In the Ising model (see [2] for a review) the spins are placed on an orthogonal grid and all interactions tend to align neighboring spins in the same direction. This situation is called a ferromagnet. Meanwhile, a model where all interactions try to get neighboring spins to point in opposite directions, is called an antiferromagnet. In this thesis, we are interested in a disordered version where both kinds of interactions occur randomly between spins. This model is called a spin glass. An illustration of these different spin systems is in Figure 1.

The interest in spin glasses started with the discovery that some materials

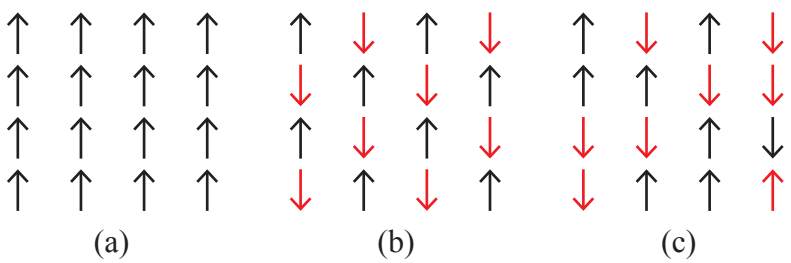

Figure 1: A ferromagnet (a), an antiferromagnet (b), and a spin glass (c). Both the antiferromagnet and the spin glass have total magnetization 0, i.e., the same number of spins pointing up and down, but the structure is different.

(mostly impure metallic alloys, such as CuMn [3]) behaved in an unexpected way at low temperatures. It was noticed that these materials exhibit a cusp in the susceptibility as a function of temperature at low temperatures [3, 4]. In other words, they have a maximum response to an external magnetic field at a finite temperature.

The earliest theoretical work on spin glasses used a model (known as RKKY from the names of the principal authors [5, 6, 7]) where the spins are three-dimensional vectors and interact with randomly placed impurities with a distance decaying force. This is a close representation of the physical system. However, it turns out that a more abstract model captures more of the essential properties. For decades now, theoretical work has mostly focused on studying a generalized Ising spin system where there are no explicit impurities. Instead, there are just spins that interact randomly.

It is remarkable that the complex physics of spin glasses is induced by this very simple and abstract model. The wanted properties did not follow immediately from adopting the model. On the contrary, powerful methods had to be invented to analyze the Ising spin glass system. In this thesis two of these are introduced: the replica trick, which is a mathematical tool for getting from microscopic rules to macroscopic properties; and the cavity method, which is a more physically motivated way to obtain similar results. Both these tools are general. They apply also outside the field of spin glass study and even outside physics, to the problems of computational complexity theory.

Computational complexity theory is also related to deriving macroscopic properties for collections of microscopic objects. In contrast to physics, the main interest is in the process itself as opposed to the original microscopic objects or the final macroscopic result. Computational complexity studies algorithms that take as input some number n of objects and in the basic form produce a single answer, a yes or a no. The question then is: what is the time complexity of the algorithm, that is, how does the time required to find the solution grow as the input grows [8]. A famous example of a complex algorithm is that of the traveling salesman. The problem goes: given a set of cities, can the salesman choose a route to go through each city once in a way that the total distance traveled is shorter than a given distance. The problem in the basic form is again answered by a single yes or no. Here the size of the input is the number of cities. An obvious extension, and a harder problem, is to ask what is the shortest route. In the computer science literature the basic form is called a decision problem, while the more general form is called an

optimization problem.

A famous result is that there is a connection between certain decision problems where finding an efficient solution for one problem leads to finding an equally efficient solution to many other problems. This result defines the class of algorithms called NP-complete. It is due to Cook [9] for a few problems and immediately followed by an extension to about 20 more problems by Karp [10] .

The problems in NP are hard, meaning that their time complexity as the function of the input size grows rapidly. The traveling salesman problem is an example of this. In contrast problems in the class P are easy, meaning that their time complexity grows slowly. It is not mathematically proven that these two classes, P and NP, are really distinct (although it is generally believed that they are [11]). NP-complete refers to the group of problems that are convertible to each other and that all belong to NP. The existence of this convertibility means that research into any of these algorithms, however abstract or useless on their own, will generate useful information on all of the problems in the class, also the useful ones. Cook and Carp did not study the traveling salesman problem. Instead, they studied boolean satisfiability which is the canonical NP-complete problem.

Boolean satisfiability deals with formulas of boolean logic [12], i.e., a sequence of boolean variables (that take values of either TRUE or FALSE) combined with the boolean operators AND, OR and NOT. An example of a boolean formula is

$$x \text{ AND (NOT } y\text{) AND ((NOT } x\text{) OR } z\text{).}$$

The question then is whether we can choose values for all the variables so that the whole formula has the value TRUE. For a small formula such as the example the answer is easily found, for the given example a configuration of $x =$ TRUE, $y =$ FALSE, and $z =$ TRUE makes the whole formula true. When the number of variables involved and the number of operators used on them both grow, finding the answer becomes harder. The satisfiability of a boolean formula has been the subject of not only theoretical work but also of numerical studies [13, 14]. Again this is mostly because of the convertibility of other problems to boolean satisfiability.

While computer scientists are more concerned with decision problems, the connection to physics comes through optimization problems. In equilibrium physics one tries to find the state of minimum energy (ground state). By imposing an energy function (Hamiltonian) on a computational complex-

ity problem, we can discuss it using terms from physics. For the traveling salesman problem the energy is simply the length of a given route while for boolean satisfiability we might work with some measure of how many constraints (the boolean operators AND, OR, and NOT) are violated. By using this insight combined with the Boltzmann probability measure $P(E) = \exp(-E/T)$ and an artificial temperature, the authors of [15] introduced the application of the simulated annealing algorithm to optimization problems, such as the above mentioned boolean satisfiability or the traveling salesman problem. Simulated annealing is a numerical algorithm for finding approximate estimates for a state of minimum generalized energy, such as the shortest route for a salesman or the minimum number of violated constraints for boolean satisfiability.

Numerical studies on boolean satisfiability have shown an interesting phenomenon. It has been found that the fraction of satisfiable formulas decreases with increasing constraints, when testing for the satisfiability of random formulas generated with a constant number of variables, but with varying numbers of constraints. Moreover, this decrease has been noticed to grow sharper and sharper with an increasing number of variables [13, 14]. A schematic of this is in Figure 2. Computer scientists call this a zero-one law, while for physicists a situation where there is a continuous transformation that turns discrete in the limit of large system size can be called by only one name: a phase transition.

The connection does not end there. While simulated annealing applies to any problem where a suitable definition of energy can be formulated, further insights have been gained by identifying boolean variables with Ising spins. This is a natural thing to do as both are variables with just two possible values. Both the above mentioned techniques of spin glass theory have been successfully applied to boolean satisfiability: the replica trick [16] and the cavity method [17]. Both analyses try to find the critical density of constraints where the phase transition from satisfiable to unsatisfiable occurs. While the former gives estimates that are close to the numerical results, the latter gives a value that is believed to be exact [18].

As an intermediate step in the cavity method applied to boolean satisfiability there appears an object called a survey of magnetic fields. The magnetic field on a spin means the tendency of other spins to try to get the spin to point up or down. In the case of boolean satisfiability it means the tendency of all the constraints to try to get one variable to take either the value

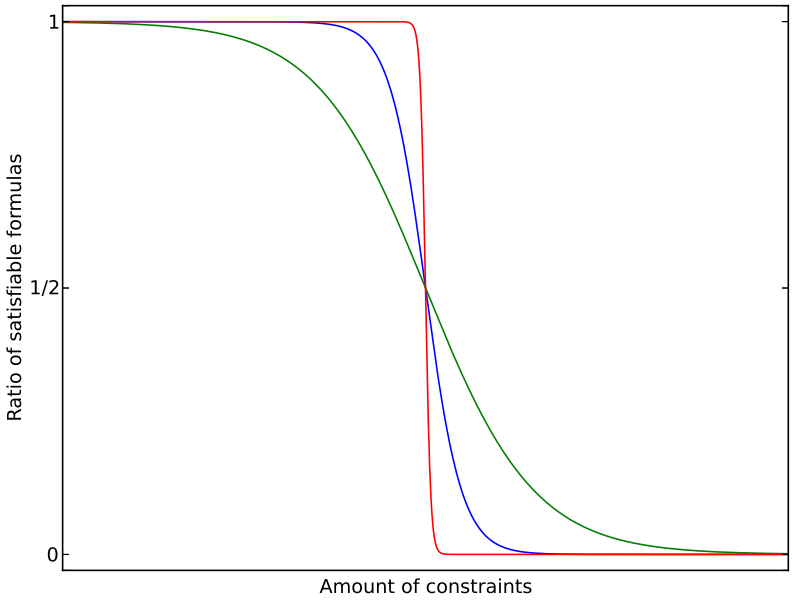

Figure 2: A schematic of the ratio of satisfiable clauses as a function of the amount of constraints. The curves represent tests with different problem sizes so that green is the smallest, blue is larger and red is largest.

TRUE or FALSE. This is merely a substep for finding an analytical value for the location of the phase transition, but it turns out that these surveys can also be calculated numerically for a specific boolean formula [19]. This process is called survey propagation and with it it is possible to find satisfying assignments for very large boolean formulas close to the phase transition. It is known from experiments that the hardest formulas occur near the phase transition [20], which makes survey propagation all the more exciting. The surveys give the probability of each variable taking a certain value and this information can be used to fix the most biased variable. The process is then repeated for the smaller problem with one less variable. While survey propagation can be used close to the phase transition, it is known that it starts to fail before reaching the critical density of constraints [21, 22].

This thesis is organized as follows. In Section 2 the replica trick on spin glasses is described. Section 3 continues by explaining the cavity method on a different kind of spin glass. Section 4 is an overview of the concepts of computational complexity theory used in this thesis while Section 5 introduces the cavity method as applied to boolean satisfiability. The survey propagation algorithm is presented in Section 6 and Section 7 contains my own con-

tributions, namely a shared memory parallelization of the existing survey propagation implementation and further numerical results. Finally, Section 8 offers a summary and conclusions.

2 Spin Glasses and the Replica Trick

The term spin glass refers to a theoretical concept and several models used in the study of a certain type of magnetic system. While a ferromagnet reaches a state of equilibrium at low temperatures, where all the unit magnets are aligned, a spin glass gets to a state of frozen disorder [4]. What this means is that the components of the system, in this thesis mostly Ising spins, tend to get frozen values so that the thermal average value of the magnetization of all spins is 0, but the average of correlations between spins is non-zero [23] or equivalently the correlation between a spin at time t_0 and some later time t is non-zero [24].

Much research has focused on spin glass models, such as the Sherrington-Kirkpatrick model [23] discussed in this thesis, simply because of their inherent relevance as a theoretical model. Nevertheless, the study of spin glasses originated from experimental phenomena, mainly from the observation that there is a cusp in the magnetic susceptibility as a function of temperature in some materials, such as metals with impurities [3]. The earliest attempts at modeling the phenomenon theoretically consisted of using magnetic moments at random locations and with an aligning tendency that decays with distance [4]. These so-called random site models were an intuitive thing to do, since this was indeed similar to what happens in the experimental studies. Regardless, it has proven much more interesting to study a more abstract model, where the spins themselves are on a uniform lattice, and the disorder is in the interactions, which are taken to be random. One such model is the Sherrington-Kirkpatrick (SK) model. Other models, where the randomness is mainly in the interactions, but the lattice is not uniform, are for example the spin glass on a Bethe lattice [25] and the spin glass formulation of the boolean satisfiability problem from computer science [19].

2.1 Sherrington-Kirkpatrick Model

In the SK model we study a rectangular grid of Ising spins, i.e., spins that can only take a value up or down, or respectively $+1$ or -1. The Hamiltonian of

the system is [26, 27]

$$H\left(\{s_i\}\right) = -\sum_{i=1}^{N}\sum_{j=i+1}^{N} J_{ij}s_i s_j - \sum_i h_i s_i, \tag{1}$$

where N is the number of spins, curly brackets indicate that the Hamiltonian is a function of the set of spins $\{s_i\}$, and the double sum goes over each pair once. The first term describes the interaction between different spins and the second the effect of a possibly site-dependent external magnetic field h_i. The bonding factors or coupling constants J_{ij} are taken to be random with a mean of 0. Depending on the model, the interactions are taken to be short or long range. Usually short range means that $J_{ij} = 0$ for all non-neighboring spins, while long range means that J_{ij} is independent of the distance, i.e., long is the same as infinite range. The SK model is an infinite range one, and the J_{ij} are further constrained to have a Gaussian distribution [23].

Sherrington and Kirkpatrick calculated some properties of the model using the so-called replica trick [23]. Among these was the susceptibility, for which they indeed got a cusp in the temperature profile, as the experiments had shown. As the ultimate focus in this thesis is on boolean satisfiability, the discussion hereafter will be more abstract in nature. The physical results will not be of utmost interest and the focus is on the mathematical tools and concepts of the replica calculation. The following subsections go through the replica derivation explaining the concepts of replica symmetry and breaking thereof. These concepts are also needed in order to understand the cavity method.

2.2 The Replica Derivation

For the purposes of this work there are three results we want to find: the free energy density per spin, an expression for the average magnetization of a spin, and the value of another order parameter, called q, which is usually the square of the magnetization [4]. We are interested not only in studying a certain instance of a SK spin glass, but also of averages over different configurations of the random couplings J_{ij}. The way we will get to the order parameters of interest is through the partition function, as is often the case in statistical physics. It turns out that when doing the average over the distribution of the random couplings, we have to do it on the free energy, instead of on the partition function. This is because averaging over the values of the

spins is fundamentally different from averaging over the values of the random couplings. Averaging the partition function would amount to waiting for both the couplings and the spins to relax, but the relaxation time of the couplings in experimental spin glasses is orders of magnitude larger. Also the theoretical model would lose its glassiness in this kind of an annealed case. Instead, we are interested in quenching the sample, so the J_{ij} are fixed, and then optimizing the disordered system for its minimum energy.

The partition function and the free energy are

$$Z = \sum_{\{s_i\}} \exp\left[-\beta H\left(\{s_i\}\right)\right], \tag{2}$$
$$F = -\frac{1}{\beta} \ln Z,$$

where the sum in the partition function goes through all the possible configurations of the spins. We denote by $P\left(J_{ij}\right)$ the distribution of the J_{ij}. The quantity from which we derive the interesting results is the average of the free energy over the distribution $P\left(J_{ij}\right)$, that is

$$[F]_{av} = -\beta^{-1} \prod_{i<j} \left[\int_{-\infty}^{\infty} dJ_{ij} P\left(J_{ij}\right)\right] \ln Z,$$

where $[\,]_{av}$ means averaging over the couplings. The product goes over the same values of i and j as the double sum in (1) and is understood to be a double product. We have to integrate over all the couplings J_{ij} so all in all there are $N\left(N-1\right)/2$ integrals. We will simplify this notation in the following way

$$[F]_{av} = -\beta^{-1} \int \prod_{i<j} \left[dJ_{ij} P\left(J_{ij}\right)\right] \ln Z,$$

where the single integral sign is taken to mean all the integrations and the limits are implicitly the same as before.

Unfortunately this is not an integral that we know how to calculate. This is where the replica trick steps in. Consider the following series expansion

$$x^n = \sum_{\nu=0}^{\infty} \frac{n^\nu \ln^\nu\left(x\right)}{\nu!}.$$

For n small we can just take the first couple of terms, order differently, and

get [23, 26]

$$\ln(x) = \lim_{n\to 0} \frac{x^n - 1}{n},$$

which is the famous replica trick. For the moment we will focus on finding $[Z^n]_{av}$ and only later substitute it back into the free energy. Recall the definition of Z in (2). Before taking the limit $n \to 0$ we will do some algebra to Z^n assuming that n is an integer.

We can write

$$Z^n = \prod_{a=1}^{n} \left[\sum_{\{s_i^a\}} \exp\left[-\beta H\left(\{s_i^a\}\right)\right] \right],$$

where we have introduced a new index a to go through identical copies (replicas) of the system. It is conventional to simplify this notation so that the n sums are represented by a kind of trace and the product of exponentials is moved to a sum inside the exponent, that is

$$Z^n = \sum_{\{s_i^1\}} \cdots \sum_{\{s_i^n\}} \exp\left[-\beta \sum_{a=1}^{n} H\left(\{s_i^a\}\right)\right] \equiv \mathrm{Tr}_s \exp\left[-\beta \sum_a H\left(\{s_i^a\}\right)\right],$$

where the sum $\sum_a$ is also a shortened version of the previous notation. We now do the average over the distribution of the couplings. We take $P(J_{ij})$ to be a Gaussian with mean 0 and variance J^2/N,

$$P(J_{ij}) = \sqrt{\frac{N}{2\pi J^2}} \exp\left(-\frac{N\left(J_{ij}\right)^2}{2J^2}\right).$$

Doing the average $[Z^n]_{av}$ is just calculating a vast number of similar and separate Gaussian integrals. The results can be collected back as

$$[Z^n]_{av} = \mathrm{Tr}_s \exp\left[\frac{\beta^2 J^2}{2N} \sum_{i<j} \left(\sum_a s_i^a s_j^a\right)^2 + \beta h \sum_i \sum_a s_i^a\right], \tag{3}$$

where we have taken $h_i = h$ for all i [4]. It is of interest for us to write the square of the sum over replicas as a double sum over two replica indices

$$\left(\sum_a s_i^a s_j^a\right)^2 = \sum_a \sum_b s_i^a s_j^a s_i^b s_j^b.$$

We can extract out of this sum the terms that have $a = b$, because $s_i^a s_j^a s_i^a s_j^a = (s_i^a)^2 (s_j^a)^2 = 1$ for all i and j. Also it is sufficient to take a sum that goes through

each pair of replicas just once because of the symmetry and just to take care of the factor of two,

$$\sum_{a,b} s_i^a s_j^a s_i^b s_j^b = n + 2\sum_{a<b} s_i^a s_j^a s_i^b s_j^b.$$

Now we insert this back in to the part of (3) where it appears and we get

$$\frac{\beta^2 J^2}{2N}\sum_{i<j}\left(\sum_a s_i^a s_j^a\right)^2 = \frac{\beta^2 J^2}{2N}\frac{N\left(N-1\right)}{2}n + \frac{\beta^2 J^2}{2N}2\sum_{a<b}\sum_{i<j} s_i^a s_j^a s_i^b s_j^b.$$

We drop the term that is not proportional to N and we add and subtract $i=j$ terms, finding now

$$= \frac{\beta^2 J^2 nN}{4} + \frac{\beta^2 J^2}{2N}\left(\sum_{a<b}\sum_{i,j} s_i^a s_j^a s_i^b s_j^b - \sum_{a<b}\sum_i s_i^a s_i^a s_i^b s_i^b\right),$$

where the last term is again not proportional to N and can be dropped. Finally we change the double sum over i and j to a square of one sum and rearrange the position of the factors N:

$$= \frac{\beta^2 J^2 nN}{4} + \frac{\beta^2 J^2 N}{2}\sum_{a<b}\left(\sum_i \frac{s_i^a s_i^b}{N}\right)^2.$$

Remember that we calculated some Gaussian integrals to get rid of the probability distribution $P\left(J_{ij}\right)$. We are now going to reintroduce Gaussian integrals by applying the well known equation

$$\sqrt{\frac{\lambda}{2\pi}}\int dx \exp\left(-\frac{\lambda x^2}{2} + a\lambda x\right) = \exp\left(\frac{1}{2}\lambda a^2\right)$$

backwards many times to our expression for the average of Z^n. Taking $\lambda = \beta^2 J^2 N$ and $a = \sum_i s_i^a s_i^b/N$ we write

$$\begin{aligned}
\left[Z^n\right]_{av} &= \mathrm{Tr}_s \exp\left(\frac{\beta^2 J^2 nN}{4}\right)\exp\left(\beta h\sum_i\sum_a s_i^a\right)\prod_{a<b}\exp\left\{\beta^2 J^2 N\left(\sum_i s_i^a s_i^b/N\right)^2\right\}^{1/2} \\
&= \mathrm{Tr}_s \exp\left(\frac{\beta^2 J^2 nN}{4}\right)\exp\left(\beta h\sum_i\sum_a s_i^a\right)
\end{aligned}$$

$$\prod_{a<b}\left[\int dQ_{a,b}\sqrt{\frac{\beta^2J^2N}{2\pi}}\exp\left\{-\frac{\beta^2J^2N}{2}\left(Q_{a,b}\right)^2+\beta^2J^2N\left(\sum_i s_i^a s_i^b/N\right)Q_{a,b}\right\}^{1/2}\right]$$
$$=\exp\left(\frac{\beta^2J^2nN}{4}\right)\int\prod_{a<b}\left[dQ_{a,b}\sqrt{\frac{\beta^2J^2N}{2\pi}}\right]$$
$$\exp\left(-\frac{\beta^2J^2N}{4}\sum_{a<b}\left(Q_{a,b}\right)^2\right)\mathrm{Tr}_s\exp\left(\frac{\beta^2J^2}{2}\sum_{a<b}\left(\sum_i s_i^a s_i^b\right)Q_{a,b}+\beta h\sum_i\sum_a s_i^a\right),$$

we have introduced a new n by n matrix $Q_{a,b}$ which is related to the previously mentioned order parameter q [4]. The form we have now is devoid of all terms containing both i and j. Instead, we have terms containing a and b. We have moved from interaction between spins to interaction between different replicas of the same spin. We will do one more transformation before stopping to discuss some of the implications and properties of the result so far. Consider the equation

$$\sum_{\{s_i^1\}}\cdots\sum_{\{s_i^n\}}\exp\left[\sum_i g\left(s_i^a\right)\right]=\exp\left\{N\ln\left[\sum_{\{S^a\}}\exp\left(g\left(S^a\right)\right)\right]\right\}, \tag{4}$$

where g is an arbitrary function and the new variables S^a are a single spin indexed by replicas and the sum goes over all the possible values of them. Let us prove that this equation holds:

$$\exp\left\{N\ln\left[\sum_{\{S^a\}}\exp\left(g\left(S^a\right)\right)\right]\right\}=\exp\left\{\ln\left[\sum_{\{S^a\}}\exp\left(g\left(S^a\right)\right)\right]\right\}^N$$
$$=\sum_{\{S_1^a\}}\cdots\sum_{\{S_N^a\}}\exp\left(Ng\left(S^a\right)\right),$$

where we have the same n times N sums as on the left side of (4) only in different order. We can change the order and write $Ng\left(S^a\right)=\sum_i g\left(s_i^a\right)$ assuming that the fluctuations do not matter in the large N limit [4]. Going back to Z^n we have

$$\left[Z^n\right]_{av}=\int\prod_{a<b}\left[dQ_{a,b}\sqrt{\frac{\beta^2J^2N}{2\pi}}\right]\exp\left(NA\left[Q\right]\right),$$
$$A\left[Q\right]=\frac{\beta^2J^2n}{4}-\frac{\beta^2J^2}{4}\sum_{a<b}\left(Q_{a,b}\right)^2+\ln\left(\sum_{\{S^a\}}\exp\left(\frac{\beta^2J^2}{2}\sum_{a<b}S^aS^bQ_{a,b}+\beta h\sum_a S^a\right)\right). \tag{5}$$

In the limit of large N this integral is solvable with a saddle point method. The saddle point method for integrals of the form $\int dx \exp\left[Nf\left(x\right)\right]$ states that in the limit of large N most of the contribution to the integral will come from the environment around the maximum of $f\left(x\right)$. Thus we can approximate f by the first couple of terms from its Taylor series, i.e.,

$$f\left(x\right) \approx f\left(x_0\right) + \frac{f''\left(x_0\right)}{2}\left(x - x_0\right)^2.$$

The term with the first derivative is missing from the expansion, since the integral is evaluated at an extremum of the function, so the first derivative is zero. All in all the task becomes to find $Q_{a,b}$ so that

$$\frac{\partial A}{\partial Q_{a,b}} = 0$$

for all $Q_{a,b}$. There are two ways to go about this, the replica symmetric way, and the replica symmetry breaking way.

Replica Symmetric Solution

The replica indices were originally introduced in the derivation as just a way to write the exponentiation of a sum to order n as n separate sums. As it is just a mathematical trick to write an equation in a different way, it is natural to assume that the problem as well as the solution should be symmetric in relation to swapping two replica indices. As it happens, this assumption does not give correct results [23, 4], but we shall first consider it before moving on to breaking of the symmetry. Stated briefly, the replica symmetric assumption amounts to specifying that [4]

$$Q_{a,b} = \begin{cases} q & \text{if } a \neq b \\ 0 & \text{if } a = b \end{cases}, \tag{6}$$

and we are left to find a solution to the single equation

$$\frac{dA}{dq} = 0. \tag{7}$$

The equation (7) at limit $n \to 0$ gives out a self-consistency relation for the order parameter q [26], the solution of which can be substituted back into the formula giving the free energy density. The key in the derivation is carrying out the sums explicitly, using the fact [4] that

$$\sum_{\{S^a\}} \exp\left(A \sum_a S^a\right) \approx 1 + n \ln\left(2 \cosh A\right),$$

when $n \to 0$. The final self-consistency relation for the order parameter q [4],

$$q = \frac{1}{\sqrt{2\pi}} \int dz \exp\left(-\frac{z^2}{2}\right) \tanh^2\left(\beta J \sqrt{q} z + \beta h\right),$$

is in a way the solution for the replica symmetric SK model. It can be proven [28, 29] that this equation gives a unique solution for q, which in turn gives a unique solution for the free energy. As it happens [23], the value differs from the one from numerical simulations by some amount and the value derived for the entropy at $T = 0$ is negative, which is of course a priori wrong. Out of the possible reasons suggested for the failure, it turned out that the replica symmetric assumption (6) is the one that causes the wrong results. Breaking this symmetry leads to the correct results [26] as defined by correspondence with numerical results and a non-negative zero temperature entropy.

Replica Symmetry Breaking Solution

In the previous section we assumed that $Q_{a,b} = q$ for all $a \neq b$. It is a difficult problem to choose $Q_{a,b}$ in any other way, because we intend to take the limit where the $Q_{a,b}$ is a 0×0 matrix. Several ansätze were suggested for the form of the matrix [30, 31], but the one that has endured the test of time is due to Parisi [32, 33, 34]. In it the matrix is divided into regions, where the value is the same within one region and each region has a distinct value q_i. The form of the regions with one division is shown in Figure 3a. Of course, we have to define this separation in a way that allows us to take the $n \to 0$ limit, see [26]. In Figure 3b the matrix is 8×8 and the number of regions is 3, but for the full solution the number of regions will go to infinity. After taking the number of regions to infinity, the matrix is shrunk to 0×0. In this limit the order parameter $Q_{a,b}$ is transformed to a function $P(q)$ on the $[0, 1]$ region [32, 36]. This situation is called full replica symmetry breaking (RSB). For full RSB the function is continuous, and for the replica symmetric case it becomes

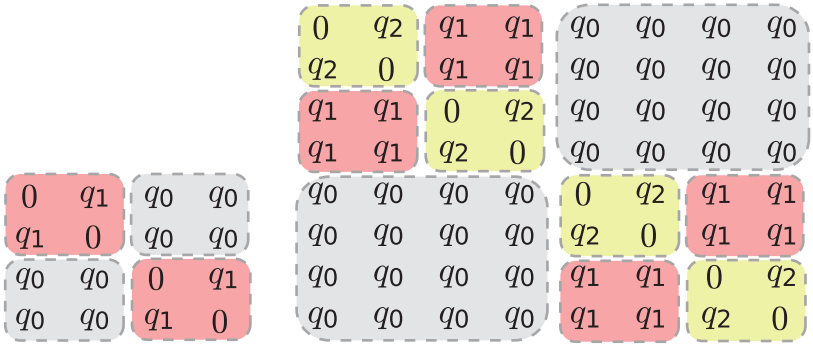

Figure 3: A 4×4 Parisi matrix with one level of replica symmetry breaking (a) and an 8×8 Parisi matrix with two levels (b) [35].

constant [37].

It has been found that in some cases full replica symmetry breaking is needed, while in others one step is enough [38, 39]. While the SK model requires full replica symmetry breaking, for the case of boolean satisfiability only one step of replica symmetry breaking seems to be enough [17]. We will limit our discussion to one step replica symmetry breaking (1-RSB) here also.

We will not go through the full computation even for the 1-RSB case, only part of it. The interested reader is referred to e.g. [26]. The interesting transformation happens in (5) with the term $\sum_{a<b} S^a S^b Q_{a,b}$. We can divide this sum into calculations of the non-diagonal regions with $Q_{a,b} = q_0$ and the diagonal regions with $Q_{a,b} = q_1$. We denote the sum over the elements in the inner regions as $\sum_{\text{blocks}} \sum_{a<b \in \text{block}} S^a S^b$. We can further simplify with $\sum_{a<b} S^a S^b = \frac{1}{2} \left(\sum_a S^a\right)^2$. For the q_0 terms we write the sum as the sum with the whole matrix minus sums with the regions in the diagonal. The q_1 terms we write as sums over the diagonal regions minus the main diagonal, since that is defined as zero. All in all

$$\sum_{a<b} S^a S^b Q_{a,b} = \frac{1}{2} \left[q_0 \left(\sum_a S^a \right)^2 + (q_1 - q_0) \sum_{\text{blocks}} \left(\sum_{a \in \text{block}} S^a \right)^2 - n q_1 \right].$$

This is again a form that only has S^a inside the sums and thus provides for carrying out the sums explicitly. After some calculation (see e.g. [26]) there appears two connected self-consistency equations, for q_0 and q_1, which for our purposes are the solution.

3 Cavity Method on Spin Glasses

The cavity method was first introduced in [40] to derive the same solutions for the Sherrington-Kirkpatrick model as had been previously gotten from the fully replica symmetry breaking solution.

Despite the historical significance of the cavity method on the SK model, we will skip that derivation. We will move straight to the cavity method on a Bethe type lattice as in [25]. This is because we are interested in the boolean satisfiability problem that has more in common with a Bethe lattice than the lattice in the SK model.

The SK model has what is called a dense graph structure because of the infinite range interactions; each spin is connected to all the other spins. The various Bethe style lattices and the boolean satisfiability are on the contrary sparse, meaning that each spin is connected to just a few other spins and the interactions are local.

3.1 The Bethe Lattice

The Bethe lattice is a name attributed to several related types of lattices. The most common type is the inside of a Cayley tree. A Cayley tree is a graph that is built by starting at a root node. One then adds $k+1$ new nodes and connects them to the root node. Then one adds k new nodes neighboring each edge node and continues repeating this process. In the graph, all nodes or spins are connected to exactly $k+1$ other nodes. We call the number of neighbors of a node its connectivity and on a Cayley tree every node has the same connectivity. The Bethe lattice is then the M innermost shells of a Cayley tree of N shells, where usually $M, N \to \infty$ with $M \ll N$. This graph has nice properties, mainly that it does not contain loops and hence the amount of frustration (the fact that the values of spins can not be chosen to minimize the energy cost of all interactions, as some interactions are conflicting) is fixed by the boundary conditions at the edge [41, 25]. Unfortunately the boolean satisfiability problem that is eventually our subject does not produce graph representations that are trees, they are only locally tree-like [19].

The model that we consider is that of a random graph, where each node is connected randomly to exactly $k+1$ other nodes and all such graphs are equally probable. Generating examples of such graphs is not trivial for large graph sizes [42, 43], but for the theoretical analysis, the exact method of generating such a graph is not important. In addition to the shape of the graph

being random, a set of coupling constants is chosen for each interacting pair of spins with some probability distribution. We will go through the outline of the general formalism without taking the case of a certain probability distribution, such as the Gaussian used in the replica formulation. In contrast to the derivation with the replica trick, the average over the distribution of J_{ij} can be taken at the very end, instead of the very beginning [25].

We are interested in the free energy at the ground state. Whereas the derivation with the replica trick wrote the partition function and got the free energy straight from it using various mathematical tricks with varying levels of soundness, the cavity method uses some simple ideas and physically motivated assumptions.

We study a Bethe lattice with a cavity defined as a set of q spins that are connected to just k other spins, instead of $k+1$ as the rest. The values of these cavity spins are fixed. We call this graph a cavity graph. We define three specific operations that we can do to the graph to transform it to another cavity graph or complete it to a Bethe lattice as defined above.

The idea is to study the shift in the ground state energy with these operations and use that information to derive the energy density with the assumption that in the thermodynamic limit of $N \to \infty$ the energy is an extensive quantity, i.e. proportional to N [25]. In other words, the difference between the energy of a Bethe lattice with N spins to one with $N+1$ spins is equal to the energy density of one spin, $e = E(N+1) - E(N)$. This is the key assumption, on which everything else is dependent. We use two of the operations to derive an equation for the energy as an integral over a certain probability distribution and the third to write a self-consistency equation for the probability distribution.

3.2 The Cavity Method

Let us take as an example a Bethe lattice graph with $k+1 = 4$ and with $2(k+1)$ cavity spins, i.e., spins that only have k neighbors. A portion of this graph with all the cavity spins and their neighbors is shown in Figure 4. We will complete this to a non-cavity graph in two different ways, using the operations of site addition and link addition [44]. Adding a site to a cavity graph means adding a new spin, and connecting it to $k+1$ cavity spins. The values of cavity spins are fixed, but the values of normal spins are optimized so as to find the values that give the smallest global energy. By adding two sites,

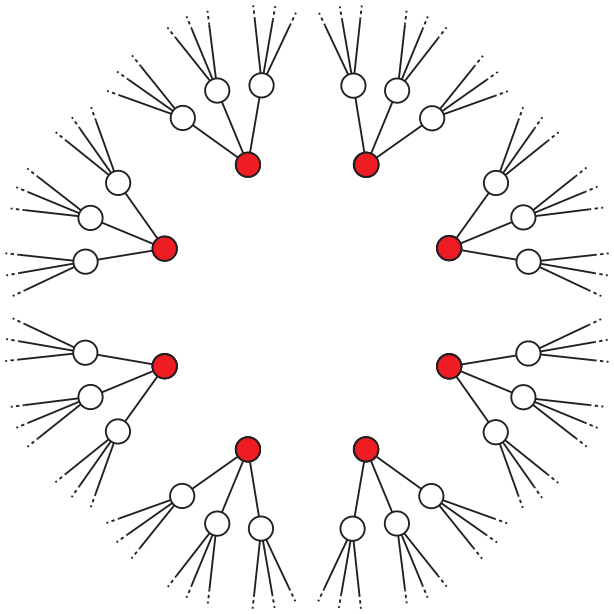

Figure 4: An example of a cavity graph from a Bethe lattice of connectivity $k+1=4$. The $2\left(k+1\right)=8$ central spins marked with red each only have $k=3$ neighbors. Notice that the image of the graph is clipped at the edges, and the actual graph continues outside the image.

we have converted the original cavity graph to a Bethe lattice of size $N+2$, as shown in Figure 5. Adding a link means connecting two cavity spins to each other, making them normal spins and thus requiring us to optimize their values for the smallest global energy. We can convert our cavity graph to a full Bethe lattice by adding $k+1$ links, thus creating a graph of size N with no cavity spins as shown in Figure 6. The assumption then is that one can calculate the ground state energy density e as

$$\begin{aligned} e &= \lim_{N\to\infty} E_N/N = \frac{E_{N+2}-E_N}{2} \\ &= \frac{1}{2}\left[\left(E_{N,2(k+1)}+2\Delta E^{(1)}\right)-\left(E_{N,2(k+1)}+\left(k+1\right)\Delta E^{(2)}\right)\right] \\ &= \Delta E^{(1)}-\frac{\left(k+1\right)}{2}\Delta E^{(2)}, \end{aligned} \tag{8}$$

where E_N is the total energy of an N-spin system, $E_{N,q}$ is the energy of an N-spin system with q cavity spins, $\Delta E^{(1)}$ is the energy of adding one site, and $\Delta E^{(2)}$ is the energy of adding one link.

We now consider the third operation for a cavity graph called iteration [44], where we add a new spin and connect it to k cavity spins. The k spins

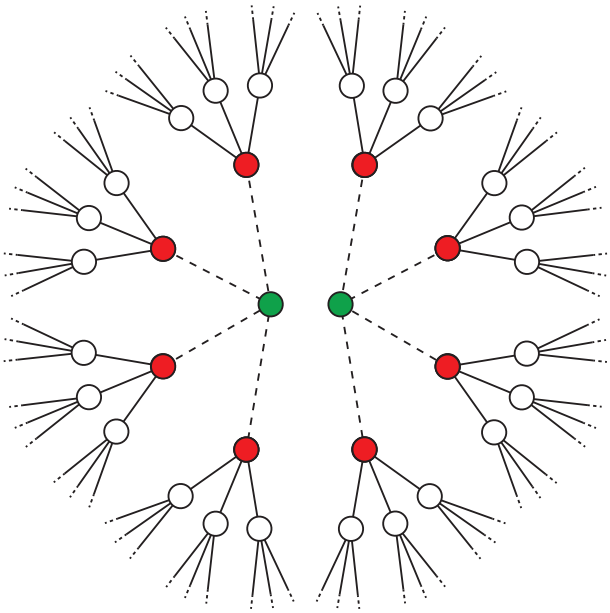

Figure 5: The graph is now a full Bethe lattice. The two added spins are shown in green and the added connections with dashed lines. The new spins as well as the former cavity spins are now optimized for minimum global energy.

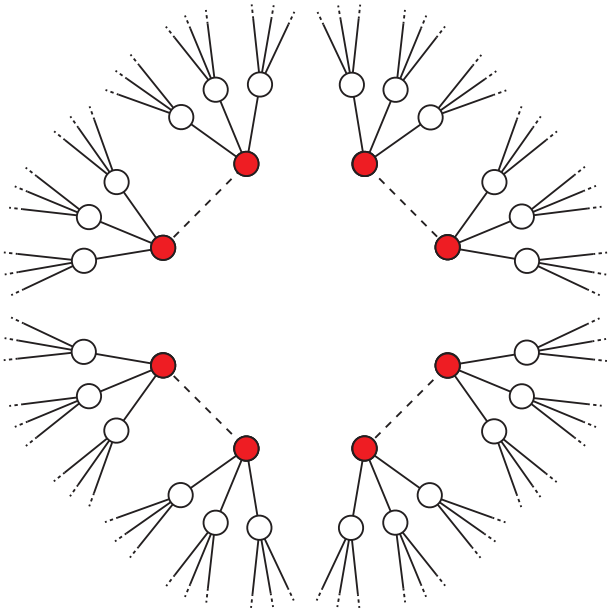

Figure 6: A full Bethe lattice with N spins. The added links are shown with dashed lines. The spins are no longer cavity spins and are thus optimized for minimum global energy.

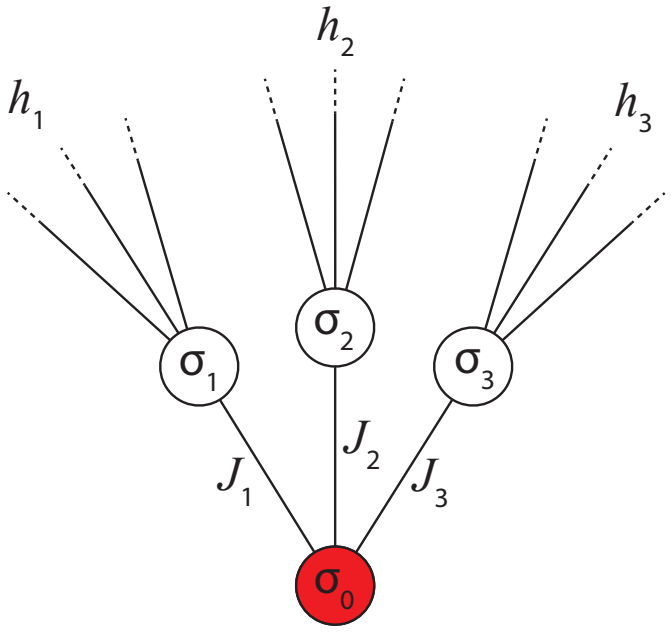

Figure 7: A new cavity spin σ_0 added in the iteration procedure as well as the old cavity spins σ_i, the new couplings J_i and the effective magnetic fields towards σ_i.

previously of the cavity type are now optimized for the smallest energy in relation to the rest of the graph and the new cavity spin with a fixed value. Let us label the k spins turned from cavity to normal as σ_i and the new cavity spin as σ_0. We define h_i as the effective magnetic field from the rest of the graph to σ_i. Finally, we define J_i as the new random coupling constant between σ_i and the new cavity spin σ_0 (Figure 7). The energy of a new link in the iteration procedure is thus a minimum of $-h_i\sigma_i - J_i\sigma_i\sigma_0$ with respect to σ_i. The fact that the spins only take values in ± 1 lets us write the minimum as an absolute value of the other variables and we further want to factor the energy to a part dependent on σ_0 and a part that does not depend on σ_0. This can be done, giving

$$\begin{aligned}\epsilon_i &= \min_{\sigma_i \in \pm 1} \left[-h_i\sigma_i - J_i\sigma_i\sigma_0\right] = -\left|h_i + J_i\sigma_0\right| \\ &= -\frac{1}{2}\left(|h_i + J_i| + |h_i - J_i|\right) - \sigma_0\left(|h_i + J_i| - |h_i - J_i|\right) \\ &\equiv -a\left(J_i, h_i\right) - \sigma_0 u\left(J_i, h_i\right),\end{aligned}$$

where we have introduced new functions a and u.

Replica Symmetric Solution

The replica symmetric (and false) assumption is that the h_i as well as a new h_0 defined as the coefficient of σ_0 in the global energy (i.e. $h_0 = \sum_{i=1}^{k} u\left(J_i, h_i\right)$) are independent random variables from the same distribution $P\left(h\right)$. This allows us to write a self-consistent equation for $P\left(h\right)$ [44],

$$P\left(h\right) = \overline{\int \delta\left(h - \sum_{i=1}^{k} u\left(J_i, h_i\right)\right) \prod_{i=1}^{k} \left[dh_i P\left(h_i\right)\right]},$$

where the bar denotes an average over the distribution of all the coupling constants as well as the distribution over random graphs. A delta function of several variables means the product of delta functions of one variable each. If we can solve the distribution $P\left(h\right)$, we can also solve the ground state energy density of the whole graph, since we can write the energy shift $\Delta E^{(1)}$ and $\Delta E^{(2)}$ in terms of this distribution. For site addition, it is [44]

$$\begin{aligned}\Delta E^{(1)} &= \min_{\sigma_0 \in \pm 1} \overline{\int \sum_{i=1}^{k+1} \epsilon_i\left(h_i, J_i, \sigma_0\right) \prod_{i=1}^{k+1} \left[dh_i P\left(h_i\right)\right]} \\ &= -\overline{\int \left(\sum_{i=1}^{k+1} a\left(J_i, h_i\right) + \left|\sum_{i=1}^{k+1} u\left(J_i, h_i\right)\right|\right) \prod_{i=1}^{k+1} \left[dh_i P\left(h_i\right)\right]}, \end{aligned} \tag{9}$$

where the optimization of the cavity spins σ_i is hidden inside ϵ_i. In the next line it is inside a and u. The optimization of the new spin σ_0 is taken care of by taking the absolute value of the sum over u. Calculating the energy of a link addition similarly entails optimizing the values of the two connected spins, σ_1 and σ_2 [44]:

$$\begin{aligned}\Delta E^{(2)} &= \min_{\sigma_1, \sigma_2 \in \pm 1} \overline{\int \left(-h_1\sigma_1 - h_2\sigma_2 - J\sigma_1\sigma_2\right) dh_1 P\left(h_1\right) dh_2 P\left(h_2\right)} \\ &= -\overline{\int \max_{\sigma_1, \sigma_2 \in \pm 1} \left(h_1\sigma_1 + h_2\sigma_2 + J\sigma_1\sigma_2\right) dh_1 P\left(h_1\right) dh_2 P\left(h_2\right)}. \end{aligned} \tag{10}$$

This is the final answer in the replica symmetric case. It gives the replica symmetric approximation for the ground state energy density.

Replica Symmetry Breaking Solution

In the replica symmetric case, we assumed that there is a single global ground state that we could treat and use to assume a single distribution $P\left(h\right)$ for the

magnetic fields. This assumption is false. Instead, we have a multitude of local ground states separated by large barriers from the global ground state, so that our discussion must use the local ground states. See [25, 44] for detailed discussions on the exact definition of a state. We label the states by an index a and say that each magnetic field h_i^a is taken from a distribution $P_i(h)$. The distributions in different locations vary and are taken from a new distribution of distributions: $\mathcal{Q}[P(h)]$. Another assumption is that the distribution of energies of local ground states is a Poisson distribution, meaning that we have to define a weighting factor μ. The equations for the energy shifts (9) and (10) do not apply as they are. Instead, the following exponential forms give the correct results [44, 38]:

$$\exp\left(-\mu\Delta E^{(1)}\right) = -\int \exp\left(\mu\sum_{i=1}^{k+1} a\left(J_i,h_i\right) + \mu\left|\sum_{i=1}^{k+1} u\left(J_i,h_i\right)\right|\right)\prod_{i=1}^{k+1}\left[dh_i P_i\left(h_i\right)\right],$$

$$\exp\left(-\mu\Delta E^{(2)}\right) = -\int \exp\left[\mu\max_{\sigma_1,\sigma_2\in\pm 1}\left(h_1\sigma_1 + h_2\sigma_2 + J\sigma_1\sigma_2\right)\right] dh_1 P_1\left(h_1\right) dh_2 P_2\left(h_2\right).$$

Now also the functions P_i are indexed. With the energy shifts defined like this, equation (8) applies almost directly. What remains, is to take the average over the distributions of P_i, namely over $\mathcal{Q}[P(h)]$. With this we have a prescription to some specific model of interaction, e.g., J_{ij} from a Gaussian distribution or uniformly from $\pm J$. The latter situation has been solved explicitly in [44] for some analytical results. Using a method of population dynamics it is possible to obtain numerical results with high accuracy [25, 44, 19].

4 Boolean Satisfiability, Factor Graphs and $P = NP$

In order to understand why we are so interested in the satisfiability of boolean clauses, we have to discuss the famous $P = NP$ question. For an excellent and thorough introduction to the subject and theoretical computer science in general, see [8]. For our purposes, this thesis will present here a very brief primer with only a physicists interest in the mathematical details and definitions.

4.1 On the Theory of Computation

When discussing the $P = NP$ question a central concept is the Turing machine. For this and most other discussions it is not really necessary to know the mathematical details of how a Turing machine is defined in terms of the operations it can perform and the language it recognizes. What is important, is that it gives us a proof that we can relatively safely discuss algorithms and their performance in the thermodynamical limit of problem size $N \to \infty$ using the inexact language of pseudocode that one might use for describing an algorithm to be implemented on a computer. We will remain safe in the knowledge that the mathematics is there in case it is needed, but we can use terms familiar to a physicist with some programming proficiency.

The key concept is that of the big O notation of computational time complexity. It describes how the time taken to calculate a function or algorithm depends on the size of its input. Consider the following pseudocode for summing a list of numbers:

```
function sum(aList)
  sum := 0
  for each number aNumber ∈ aList do
    sum := sum + aNumber
  return sum,
```

where `:=` means assigning a value to a variable and the other symbols take their self-evident meaning.

Let us derive the time complexity of this function. There are two rows where the variable `sum` is manipulated and one where it is simply returned out of the function. All of these take a constant amount of time regardless of the size of the list. The row where the variable `sum` is incremented is repeated for each number on the list. For our analysis, there remains only the row where the variable `aNumber` is assigned to each member of the list one by one, and this can also be assumed to take a constant time. All in all the time taken, when we count the time taken for each row is $1+1\cdot n+1\cdot n+1 = 2n+2$. We say that the time complexity of our function sum is $\mathcal{O}(n)$, by which we mean that as n goes to infinity, there exists some constant c so that the function takes less time than cn. Clearly here one such c could be 3, since $3n > 2n + 2$ for all $n > 3$.

This treatment assumed that taking a number from a list is a task that takes a constant time. We always have to be careful to only make such assumptions

if they are correct. It is reasonable to assume a list that can be accessed in constant time, but it is also easy to imagine a list where taking a number takes for example a time comparable to the size of the list.

Consider writing the numbers on rocks and laying them out in a row, this being the list. Now consider you have a stationary adding machine that you have to bring the rocks to. Now also imagine the rocks being so heavy that you can only carry one at a time. Then clearly the above analysis of the function `sum` is wrong, because the first numbers are close by and therefore fast to get to, but the last ones are a distance comparable to n away and take a long time to access. We will not consider such contrived examples further, but it is necessary to keep in mind how long each operation on the pseudocode takes. This is the price we pay for working on a higher level of description instead of straight at the Turing machine level.

Let us take another example, that of finding the largest difference between numbers in a list:

```
function largestDifference(aList)
  largestDifference := (−∞)
  for each number aNumber ∈ aList except the last
    for each number anotherNumber ∈ {numbers
                after aNumber on the original aList}
      currentDifference := |anotherNumber−aNumber|
      if currentDifference > largestDifference
        largestDifference := currentDifference
  return largestDifference
```

Again there are two rows that take constant time, the first and the last. All the three rows that are repeated in the innermost part take constant time. What remains is to find out how many times they are repeated. Clearly the first `for` repeats what follows $n-1$ times, where n is the size of the list. But the next `for` repeats over a different sized list each time. On the first go it goes through $n-1$ numbers, on the next through $n-2$, and so on. Finally it is left with just one number. There is a closed form formula for this kind of a sum $\sum_{i=1}^{n-1} i = n\left(n-1\right)/2$. So all in all the time taken is

$$1 + 5n\left(n-1\right)/2 + 1 = 5n^2/2 - 5n/2 + 2,$$

where we have again assumed that getting a number both from the list and from a specific slice of the list is a constant time operation. The time com-

plexity then is $\mathcal{O}(n^2)$. Notice how all the lower orders of n have no effect, as we are only interested in the large n limit. We do not have to always go even to this level of tedious bookkeeping in order to find out an algorithm's time complexity. We can just immediately see for example for the `largestDifference` function that it goes through the list twice the second time inside the first, so we immediately know that it takes $\mathcal{O}(n^2)$ time. We just have to be very careful that we really know how long each portion of the code takes. Notice how both of these algorithms take an amount of time polynomial in the size of the problem. With this background and before going to $P = NP$, we are now ready to start the discussion of the boolean satisfiability problem.

4.2 Boolean Satisfiability Problem

Consider a set of boolean variables $\{x_i\}$, i.e., a set of variables, each of which can take only the values TRUE or FALSE. A boolean formula is a collection of these variables with repetition combined with the operators AND ($\wedge$), OR ($\vee$), and NOT (denoted by a bar over the negated variable or group of variables). One example of such a formula is

$$(x_1 \wedge x_2) \vee \overline{x_3} \vee \overline{(x_4 \vee x_5)}.$$

Without losing generality [18], we restrict ourselves to formulas of the form

$$\bigwedge_{a=1}^{M} \left(\bigvee_{i=1}^{k} z_i^a \right).$$

We take a conjunction (AND) of so-called clauses, that consist only of a fixed number k of single, possibly negated variables combined with OR. Here z_i^a means the i:th variable of the a:th clause either negated or not, i.e., $z_i^a \in \{x_i, \overline{x_i}\}$. This form is known in the literature as k-conjunctive normal form, or k-CNF for short [18].

The satisfiability problem is to find out whether there is an assignment of truth values to the variables $\{x_i\}$ so that the truth value of the whole formula is true. We call such an assignment a configuration, with $C \in \{\text{TRUE, FALSE}\}^N$, where N is the number of variables. This seemingly simple question is of immense importance.

Now let us write an algorithm to solve this problem:

```
function solve(variables, clauses)
```

```
  solved := false
  for each configuration aConfiguration ∈
          {possible configurations of the variables}
    if aConfiguration makes all clauses satisfied
      solved := true
  return solved
```

Clearly in this algorithm some rows take anything but a constant time. The repetition goes through all possible configurations of the variables, that is, the following part is repeated 2^N times, where N is the number of variables. We will not write an algorithm for checking whether a configuration satisfies a formula, but we can reason how long such a function should take. Such an algorithm should go through all the M clauses once and it should take at most an amount of time proportional to k, the number of variables in a clause. Since k is a constant independent of the size of the problem, the time complexity of this check is thus $\mathcal{O}(M)$ and the total time complexity of the solving algorithm is $\mathcal{O}(2^N M)$. We are only interested in studying problems with the ratio of clauses per variable a constant $\alpha = M/N$ independent of the problem size, so we can further simplify the complexity to $\mathcal{O}(N2^N)$. This algorithm takes an amount of time exponential in the problem size.

4.3 $P = NP$

Consider a different algorithm for solving the satisfiability problem:

```
function nondeterministicSolve(variables, clauses)
  solved := false
  choose one random configuration aConfiguration ∈
         {possible configurations of the variables}
  if aConfiguration makes all clauses satisfied
    solved := true
  return solved
```

Clearly this algorithm does not take $\mathcal{O}(N2^N)$ time, but instead just $\mathcal{O}(N)$. Why is this interesting? Because this is precisely the way the class of problems called NP is defined. This statement requires some clarification. The theory of computation using Turing machines is made easier by deriving concepts of higher abstraction with straightforward methods from lower abstractions. One such abstraction is a non-deterministic Turing machine that can take a

guess. The definition for the non-deterministic Turing machine is that it always guesses "correctly". Using this definition the above algorithm solves the boolean satisfiability problem in polynomial time. The letters P and NP stand for polynomial and non-deterministic polynomial. The often heard distinction of polynomial versus exponential time is partly justified because it can be proven that any non-deterministic algorithm can be simulated by a deterministic one in exponential time by just going through all the possible combinations [8].

The problem classes P and NP are just the two innermost classes in the hierarchy of problem classes. Let us consider a boolean formula that is known not to be satisfiable and then ask what is the minimum amount of unsatisfied clauses in the formula. This problem, called an optimization one, is not solvable in polynomial time even on a non-deterministic machine, since there is no simple check for whether a solution is a minimum. The only option is to go through all the options keeping track of the minimum, and this truly takes exponential time.

4.4 Boolean Variables as Spins and the Factor Graph Presentation

The obvious way to write down a specific boolean formula is of course the form already introduced using propositional calculus, i.e., as logical variables $\{x_i\}$ connected by the operators AND, OR, and NOT. With this representation all $x_i \in \{\text{TRUE, FALSE}\}$ and testing whether a formula is satisfied with a configuration of variables $C \in \{\text{TRUE, FALSE}\}^N$, amounts to using the simplification rules of propositional calculus (FALSE$\vee$FALSE $=$ FALSE, TRUE$\vee x =$ TRUE, TRUE $\wedge$ TRUE $=$ TRUE, FALSE $\wedge\, x =$ FALSE) successively to finally get a single TRUE or FALSE.

But there is another way. A boolean variable is an object that can have one of two values, TRUE or FALSE. An Ising spin is also an object that can take one of two values, ± 1. We can describe a boolean formula as a similar structure to the Bethe lattice studied in the previous chapter. While for spin glasses each interaction is pairwise between two spins, for a boolean formula there are interactions between a group of spins. We can visualize this as a graph with two kinds of vertices. One type of vertice is spins as with the Bethe lattice. The other type is called a function or factor in the literature. Each interaction (clause) is represented by this latter type of vertice. In figure

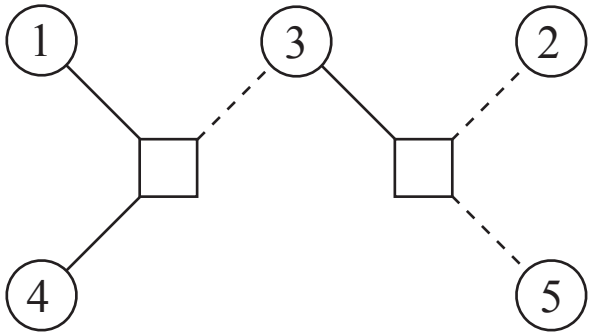

Figure 8: The factor graph presentation of boolean formula. The squares mark clauses and we call them function nodes or factors while the circles mark variables and we called them variable nodes. A variable that appears negated in a clause is connected with a dashed line and a variable that appears plain is connected with a full line.

8 there is an example of such a structure, called a factor graph [45] for the following example formula:

$$(x_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_5)\,.$$

We can now visualize a boolean formula as similar to a graph of connected spins. What about the satisfiability of a formula? A boolean formula is satisfied, if given a configuration of variables with specific values of TRUE or FALSE, it simplifies to TRUE. It is not satisfied if it simplifies to FALSE. What we will do is write a Hamiltonian that takes the value 0 if the formula is satisfied and a value greater than zero if it is not. We have restricted our formulas to a form where we need each of the clauses to take the value TRUE for the whole formula to be TRUE. This suggests that we should take [46]

$$H\left(C\right) = \sum_{a=1}^{M} E_a,$$

where E_a is the energy of each clause. If the clause is TRUE, $E_a = 0$, and $E_a > 0$ if it is FALSE. It does not really matter what is the exact form of E_a [19], so we simply choose one way. The following formula has the property that $E_a = 1$ for a value of FALSE and $E_a = 0$ for a value of TRUE:

$$E_a = \prod_{r=1}^{k} \frac{1 + J_r^a \sigma_{i_r}}{2},$$

where i_r is an index between 1 and N, σ_i is a spin corresponding to the variable x_i with $\sigma_i = 1$ if $x_i =$ TRUE, and $\sigma_i = -1$ if $x_i =$ FALSE. The coupling constant J_i^a is a number in ± 1 so that $J_i^a = -1$ if x_i is negated in the clause a and $J_i^a = 1$ otherwise. With these definitions we are ready to apply the cavity method to the problem of boolean satisfiability.

5 Cavity Method on Boolean Satisfiability

The cavity method is a tool to analyze the properties of a randomly shaped graph taken from a probability distribution with random couplings following their own probability distribution. In Section 3 we studied spin glasses on a Bethe lattice, but here the kind of graph we are interested in is that of boolean satisfiability in its form called 3-SAT. This means that the formulas are organized as clauses of three variables each connected with OR whereas clauses are connected by AND. We produce a random sample of a 3-SAT problem by taking N variables and connecting each distinct triplet of variables with a function (factor) node with the probability $6\alpha/N^2$. For each function node we take coupling constants with equal probability from ± 1. This gives on average a total of $M = \alpha N$ function nodes and a connectivity of variable nodes taken from a Poisson distribution with mean 3α [19].

The cavity method analysis of boolean satisfiability proceeds in much the same way as for the Bethe lattice. We will deal chiefly with the 3-SAT version of boolean satisfiability. We define again a cavity graph. Now it is defined as the original problem with N variables, but where we fix the values of $2r$ variables. Here r is a random number taken from the same Poisson distribution with mean 3α as that used for the original graph. We choose this distribution because we are going to add a new variable connected to these $2r$ cavity variables through r new function nodes. This gives the (almost) correct connectivity for the new variable. There is a small error, as the new SAT problem has function nodes that have been generated in proportion to N^{-2} instead of $(N+1)^{-2}$ making the total average connectivity slightly too high. We will address this issue later on. [19]

Replica Symmetric Solution

To get forward we suppose that the energy of the original cavity graph can be written as an additive function of the fixed values of the cavity spins [19]

$$E\left(\left\{\sigma_i^1, \sigma_i^2\right\}\right) = A - \sum_{i=1}^{r} \left(h_i^1 \sigma_i^1 + h_i^2 \sigma_i^2\right), \tag{11}$$

where we could have as well written the sum to go through $2r$ values. We chose instead to write the two cavity variables to be connected to each new function node separately. Here A is a constant whose value does not matter.

We want to calculate the energy when adding a new spin and the corresponding functions. We add a spin and keep the new spin fixed but optimize the values of the previous cavity spins for smallest global energy. We do this by denoting the new energy as

$$E\left(\sigma_0\right) = A - \sum_{i=1}^{r} \left(\epsilon_i^1 + \epsilon_i^2 + E_i\right),$$

where A is the same constant. The ϵ_i^j are the optimized link energies from the magnetic fields to the former cavity spins. The energy of each new function node is E_i. Without writing the factors explicitly, we assume that these can again be factored as a part dependent on the value of σ_0 and a part independent of σ_0, in other words

$$E\left(\sigma_0\right) = A - \sum_{i=1}^{r} a\left(\left\{J_i^j\right\}, \left\{h_i^j\right\}\right) - \sigma_0 \sum_{i=1}^{r} u\left(\left\{J_i^j\right\}, \left\{h_i^j\right\}\right). \tag{12}$$

As with the Bethe lattice, we identify

$$h_0 = \sum_{i=1}^{r} u\left(\left\{J_i^j\right\}, \left\{h_i^j\right\}\right) \equiv \sum_{i=1}^{r} u_i, \tag{13}$$

which again is used in building a self-consistency equation for the distribution of magnetic fields $P\left(h\right)$. The exact form of the self-consistency relation is dependent on the interactions, so it is obviously different from that of the Bethe lattice spin glass. The magnetic field at the new variable is a sum of the cavity biases u_i it receives. By averaging over the Poisson distribution

marked $f_{3\alpha}(r)$ of the random choice for the number of new functions we get

$$P(h) = \sum_{r=0}^{\infty} f_{3\alpha}(r) \int \delta\left(h - \sum_{i=1}^{r} u_i\right) \prod_{i=1}^{r} \left[du_i Q(u_i)\right], \tag{14}$$

where $Q(u)$ is the distribution of the cavity biases. These are in turn dependent on the input magnetic fields $\left\{h_i^j\right\}$, so we can complete the self-consistency with the second of the two dependent equations [19]

$$Q(u) = \overline{\int \delta\left[u - u\left(\left\{J^j\right\}, \left\{h^j\right\}\right)\right] dh^1 dh^2 P(h^1) P(h^2)}. \tag{15}$$

If we assume that the previous equations for $P(h)$ and $Q(u)$ allow us to solve $P(h)$ we can calculate the shift in energy when we add one spin. This is the same as the ground state energy density with the assumption that the energy is an extensive quantity. By subtracting the energy of the original cavity graph (11) from the energy of the $N+1$ graph (12), we get for the energy shift with a specific value of r[19]

$$\Delta E = \sum_{i=1}^{r} \left[-a_i + \left(h_i^1 \sigma_i^1 + h_i^2 \sigma_i^2\right)\right] - \sigma_0 \sum_{i=1}^{r} u_i.$$

If we now optimize this for the values of both the original cavity spins and the new spin, we get

$$\Delta E = \sum_{i=1}^{r} \left[-a_i + \left|h_i^1\right| + \left|h_i^2\right|\right] - \left|\sum_{i=1}^{r} u_i\right|.$$

Finally, by averaging this over the choice of r, the random couplings J, and over all the random graphs we get

$$\overline{\Delta E} = \sum_{r=0}^{\infty} f_{3\alpha}(r) \overline{\int \Delta E \prod_{i=1}^{r} \left[dh_i^1 dh_i^2 P(h_i^1) P(h_i^2)\right]}.$$

This is almost the final answer for the ground state energy density, but we need to make a correction for having slightly too many function nodes as mentioned before. We delete on average 2α function nodes after adding a new variable. Each deletion shifts the energy by [19]

$$\overline{\Delta E'} = \overline{\int \left(\sum_{i=1}^{3} |h_i| + \min_{\sigma_1,\sigma_2,\sigma_3 \in \pm 1} \left[E(\sigma_1, \sigma_2, \sigma_3) - \sum_{i=1}^{3} h_i \sigma_i\right]\right) \prod_{i=1}^{3} dh_i P(h_i)},$$

giving finally the energy density as

$$e = \overline{\Delta E} - 2\alpha\overline{\Delta E'}.$$

Replica Symmetry Breaking Solution

As with the Bethe lattice, the replica symmetry breaking assumption is that we have a multitude of local ground states instead of a single global one. Similarly, instead of handling directly the distributions $P(h)$ and $Q(u)$ we must look at so called surveys on these distributions, i.e., distributions of distributions: $\mathcal{P}[P(h)]$ and $\mathcal{Q}[Q(u)]$. The simple iteration equation for a magnetic field (13) now gets an extra exponential reweighting term [19]. So the replica symmetry breaking self-consistent pair of equations for cavity biases and magnetic fields (14) and (15) becomes

$$\begin{aligned} Q_i(u) = \int \exp\left[\mu\left(a\left(\left\{J^j\right\},\left\{h^j\right\}\right) - \left|h^1\right| - \left|h^2\right|\right)\right] \\ \times \delta\left[u - u\left(\left\{J_i^j\right\},\left\{h_i^j\right\}\right)\right] dh^1 dh^2 P_i\left(h^1\right) P_i\left(h^2\right), \end{aligned} \tag{16}$$

$$P_0(h) = \int \exp\left[u\left|\sum_{i=1}^{r} u_i\right|\right] \delta\left(h - \sum_{i=1}^{r} u_i\right) \prod_{i=1}^{r} \left[du_i Q(u_i)\right]. \tag{17}$$

Notice that in these equations both the average over the distribution of coupling and the average over the choice of r are missing. This is because we do not expect to find a closed form for the solution of this pair of equations, but instead iterate with this update mechanism on a population dynamics algorithm [19, 47] and collect detailed numerical statistics. The average over the couplings and choices of r is done in the population dynamics algorithm by successively choosing new couplings and new r. The weighting factor μ needs to be taken into account also when calculating the energy.

The iteration equations (16) and (17) are defined for creating a new survey on magnetic fields, but they could equally well be used to define a new survey on cavity biases. This updating of cavity biases based on neighbors is at the heart of an algorithm called survey propagation for solving a single instance of a satisfiability problem. In it the weighting factor μ is fixed, and the delta functions and other mathematical subtleties are mostly abstracted out for efficient and simple implementation on a computer. The following section describes the algorithm first in the replica symmetric and then in the replica symmetry breaking way. Instead of starting from the equations, an

intuitive understanding of the algorithm is emphasized.

6 Survey Propagation

The analysis of the previous section inspired an efficient algorithm for solving random instances near the satisfiable-unsatisfiable boundary that was mentioned in the introduction. This algorithm, which is derived from analytical results specifically on a random instance, does not fare equally well on some handcrafted hard instances [48, 49], but for truly random instances it is extremely efficient.

Survey propagation (SP) is understood easiest when described in terms of factor graphs. The order of presentation here will follow loosely that of [50]. We will first go through the warning propagation (WP) algorithm which is easier to understand if not very useful in itself. However, WP is an essential substep as it introduces the same key concepts that are used also in survey propagation.

6.1 Warning Propagation

Both the propagation algorithms work by passing messages from node to node. In warning propagation the important messages are cavity fields and warnings. A warning is something a function node sends to a variable node and a cavity field is something that a variable node sends to a function node. A warning can be either 0 or 1; read as "no warning" or "warning", respectively. The meaning of a function node sending a warning (i.e., the value 1) is that it needs the variable to take the correct value in order to be satisfied. Remember that every function node needs to be satisfied for the formula to be satisfied, but it is sufficient for just one variable to satisfy a clause. A cavity field is a number that is either positive, negative or zero. A positive cavity field sent from a variable to a function node means that in the absence of the function node the variable would prefer to be TRUE, a negative cavity field means FALSE, and 0 means that the variable does not have a preference. The meaning of preference here is just what the other function nodes are telling the variable node. The way the function nodes tell a variable something is by sending warnings. This is best clarified through an example.

Suppose we know for a small segment of a graph shown in Figure 9 all the warnings except the one sent from function node a to variable node i,

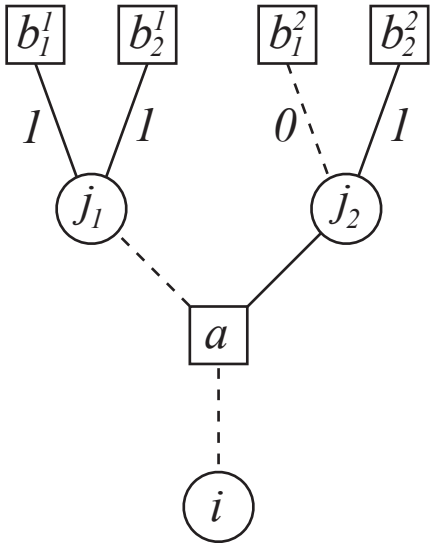

Figure 9: Related to the calculation of the warning from a to i. The information needed is the set of couplings shown as dashed or full lines and the set of warnings shown as numbers in the lines connecting the different b_y^x to j_x.

that is all warnings $b_y^x \to j_x$ are known numbers in $\{0, 1\}$ and their values are shown in the figure. Here and elsewhere in this thesis letters a and b refer to functions and letters i and j refer to variables. Variables that appear negated in a clause are marked by a dashed line. The segment pictured in Figure 9 corresponds to the formula

$$j_1 \wedge j_1 \wedge \overline{j_2} \wedge j_2 \wedge \left(\overline{j_1} \vee j_2 \vee \overline{i}\right) .$$

This particular formula is obviously not satisfiable, since there are conflicting demands on j_2. As this formula does not constitute the whole original boolean satisfiability problem, the whole problem can still be satisfiable. In actuality the clauses b_y^2 could be linked to other variables not shown in this segment. It is worth noticing that the graph is cut above the b_y^x functions and below the variable i, since we do not need to know about them for calculating the warning from a to i.

Let us go through the calculation from left to right using the language of personification as used up to this point. We will soon translate this into exact mathematical formulation. In order to calculate the warning from a to i, we need the cavity fields from the variables j_x to a. For getting those we need the warnings from b_y^x to j_x.

Let us consider first the cavity field from j_1 to a. In the absence of a, j_1

would be TRUE, since both b_y^1 are warning it that they need j_1 to be TRUE. Thus the cavity field of j_1 tells a that j_1 will be TRUE, and a knows that this means that j_1 will be of no help, as a would want a FALSE from j_1. Next a turns to j_2 for help. The function b_1^2 does not send a warning to j_2, which leaves j_2 free to listen just to b_2^2 that does send a warning. The effect is that j_2 sends a positive cavity field (meaning it takes the value TRUE) to a, which immediately satisfies a. Now that a is satisfied, it does not need anything from any other variables, so it sends a warning of 0 to i, meaning i is free to do as other functions demand. Suppose the warnings of b_1^2 and b_2^2 were reversed. Then j_2 would choose FALSE, a will not be satisfied by j_2 and a would correspondingly send a warning of 1 to i.

With the addition of some notation we are now ready to write the calculation of a single warning as a simple equation. Let a warning sent from function a to variable i be $w_{a\to i}$, the coupling between a function (clause) a and a variable i be J_i^a so that $J_i^a = -1$ if i appears negated in a and $J_i^a = 1$ if i appears plain. Let a cavity field from variable i to function a be $\mathrm{cf}_{i\to a}$ and finally let θ be the Heaviside step function. Also denote by $V(a)$ the set of variables connected to a and by $F(i)$ the set of functions connected to i. The new value of the warning sent from a to i is

$$w_{a\to i} = \prod_{j\in V(a)\setminus i} \theta\left(-J_j^a \mathrm{cf}_{j\to a}\right),$$
$$\mathrm{cf}_{j\to a} = \sum_{b\in F(j)\setminus a} J_j^b w_{b\to j}.$$

On first look this seems like a chicken-and-egg problem: in order to calculate any warning we already have to know the value of all other warnings. It turns out that this is exactly the case. The solution is to start from purely random initial warnings and to update all the warnings from the current value of the other warnings. The process is iterated over many times successively in the hope that at some point a fixed point set of warnings is reached. Then all the warnings will have converged to values that do not change anymore when calculated again from the other warnings. This assumption of convergence seems pretty miraculous, but for a special class of factor graphs that are trees, i.e., they do not contain any loops, it can be proven to hold [45]. Unfortunately, it is known from empirical tests that for general factor graphs with loops, this simple algorithm does not converge. The key benefit of the survey propagation algorithm is that it often converges for graphs that are not

trees. The condition for convergence is that the graph is locally tree-like, i.e., the loops are long [50, 19]. Before we go on into SP, we still need to see how the set of fixed point warnings can be used to solve the underlying boolean satisfiability problem, which after all has been our main aim all along.

We will go through a step by step example of solving a small boolean satisfiability problem with warning propagation. The steps are shown in Figure 10. Consider the factor graph and the fixed point warnings shown in Figure 10a. Let us solve this first by heuristic reasoning, before again writing an exact formulation. In the graph, some variables receive non-zero warnings, while others receive only zero warnings. Remember that a warning from a function to a variable means that the function absolutely needs the variable to be satisfied. If a variable receives conflicting warnings from two functions, in other words the variable is commanded to take two opposing values at once, we can immediately say that the formula is not satisfiable. A happier situation arises when one or more functions demand the same value from a variable. Then we can simply fix that variable. After fixing we have to see what happens to the other functions connected to the fixed variable: If they too are satisfied, they can be completely removed; if they are not, their ties with the fixed variable can be severed.

We can do the fixing in arbitrary order, e.g., starting with the variable 1 (Figure 10a). The variable 1 receives a warning from a, telling it to take the value FALSE. This value does not satisfy b, so the coupling from b to 1 is severed, but b is not removed (Figure 10b). Next we fix the value of 2, again to FALSE and remove function b with its connections and the connection of c and 2 (Figure 10c). Variable 3 does not get any warnings, so we do not fix it yet. We fix 4 to TRUE (Figure 10d). Finally we fix 5 and 6 and remove e and f and cut ties accordingly (Figure 10e). Now we are left with a reduced problem, and we can start again by iterating from random warnings to get the fixed point warnings. In this case the graph is remarkably simple and the fixed point warning is trivially found (Figure 10f), so we can fix the value of 3. We now have a full solution to the boolean satisfiability problem corresponding to the factor graph.

To put the solution of a boolean satisfiability problem using the set of fixed point warnings on a mathematical founding, we introduce two values, the local field H_i and contradiction number c_i of variable nodes [50]:

$$H_i = \sum_{b \in V(i)} J_i^b w_{b \to i},$$

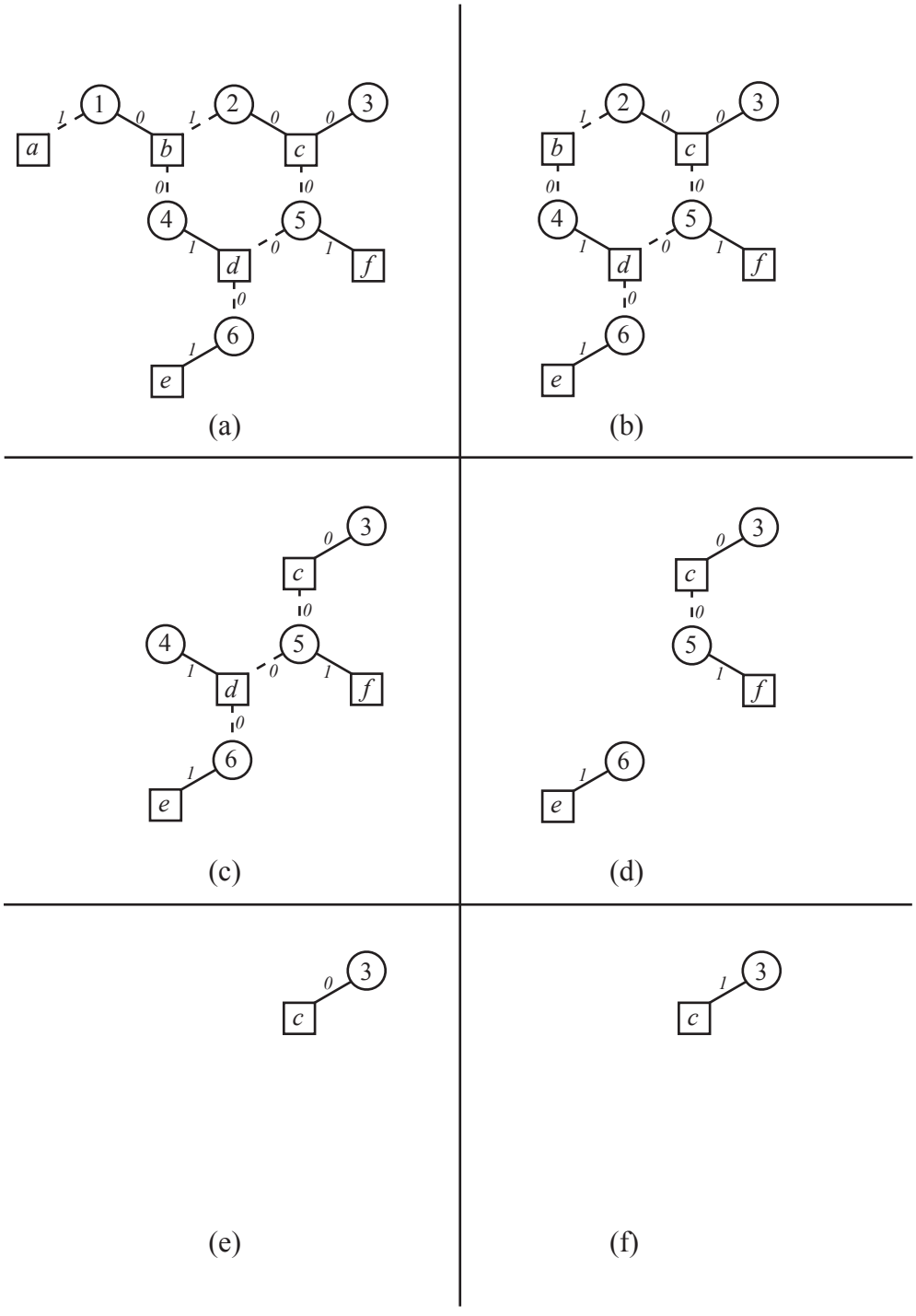

Figure 10: The full solution step by step for the boolean satisfiability formula using the set of fixed point warnings. See the text for detailed explanation.

$$c_i = \begin{cases} 1 & \text{if } \left(\sum_{b\in V_+(i)} w_{b\to i}\right)\left(\sum_{b\in V_-(i)} w_{b\to i}\right) > 0 \\ 0 & \text{otherwise,} \end{cases}$$

where the notation $V_\pm(i)$ means neighboring function nodes of i, where i appears non-negated and negated respectively. If any variable node has a contradiction number of 1, it instantly means that the whole formula is unsatisfiable, because two functions would need two different values from the variable. In the situation where all contradiction numbers are 0, we can use the local fields to fix all the variables that have a nonzero local field. We fix these variables to the sign of the local field, i.e., positive to TRUE and negative to FALSE. After fixing all the variables with non-zero local field and reducing the problem accordingly, we run warning propagation again to find new fixed point warnings. If it happens that at some iteration all the local fields (and warnings) are zero, the problem is underconstrained. Then we can fix a random variable to a random value, and iterate warning propagation again.

6.2 Survey Propagation

Survey propagation uses the same warnings as WP but adds a layer of indirection. Instead of sending warnings, the algorithm uses surveys that are probabilities of sending a warning. The value of a survey, denoted $\eta_{a\to i}$ is real in the interval $[0, 1]$, while in WP the warnings were integers 0 or 1. Similarly to WP, the SP algorithm works by starting from random initial surveys and updating them sequentially, until hopefully a fixed point set of surveys is reached. Whereas WP only converges for trees, SP can converge also for general graphs with loops. Numerical experiments show that this is often the case [50]. The equation for updating a survey from a function a to a variable i, $\eta_{a\to i}$, is [50]

$$\eta_{a\to i} = \prod_{j\in V(a)\setminus i} \left[\frac{\Pi^u_{j\to a}}{\Pi^u_{j\to a} + \Pi^s_{j\to a} + \Pi^0_{j\to a}}\right],$$

where the non-normalized probability functions Π are written as [50]

$$\Pi^u_{j\to a} = \left[\prod_{b\in V^s_a(j)} (1-\eta_{b\to j})\right]\left[1 - \prod_{b\in V^u_a(j)} (1-\eta_{b\to j})\right],$$

$$\Pi^s_{j\to a} = \left[\prod_{b\in V^u_a(j)} (1-\eta_{b\to j})\right]\left[1-\prod_{b\in V^s_a(j)} (1-\eta_{b\to j})\right],$$
$$\Pi^0_{j\to a} = \prod_{b\in V(j)\setminus a} (1-\eta_{b\to j}),$$

where the superscript s is short for satisfying, u for unsatisfying, and 0 for the neutral or indifferent case. We denote as $V^s_a(j)$ ($V^u_a(j)$) the set of function nodes, excluding a, that neighbor j and that want the same (different) value from j as a. These are the function nodes that tend to cause j to satisfy (unsatisfy) a.

The survey $\eta_{b\to j}$ is the probability of b sending a warning to j, i.e., the probability that b needs j to take a certain value in order to be satisfied. Accordingly, $(1-\eta_{b\to j})$ is the probability that b does not care which value j takes.

Let us go through the meaning of the functions Π. The product $\prod_{b\in V^s_a(j)}(1-\eta_{b\to j})$ is the probability that none of the functions b, wanting the same value out of j as a does, care about the value of j. The value in the second square brackets is the probability that one or more of the functions in $V^u_a(j)$ do care about the value of j. In total $\Pi^u_{j\to a}$ is the non-normalized probability that j will take the wrong value from the point of view of a. The other functions Π are needed to normalize the probability.

After this the new survey $\eta_{a\to i}$ is easily understood as the probability that all the other variables j take the wrong value for a. Equivalently, it is the probability that a needs i, i.e., that a sends a warning to i. If one of the products in the equations is over an empty set, the product is 1 by definition. This is because all the products amount to probabilities that none of the members of the set send a warning. The probability that no one sends anything from an empty set is 1.

The fact that SP does often converge to a set of fixed point surveys is quite remarkable. Nothing has been explicitly done to guarantee convergence. Indeed, SP does not converge for problems with the value of the clause to variable ratio α within the satisfiable region but too close to the phase transition [21]. SP usually finds fixed point surveys for randomly generated instances close to the phase transition.

Once we have the set of fixed point surveys, we can use them to solve the underlying satisfiability problem. We define different functions Π, once

again a neutral one but this time a positive and a negative one too [50]:

$$\begin{aligned}
\hat{\Pi}_i^+ &= \left[\prod_{a\in V_-(j)} (1-\eta_{a\to i})\right]\left[1-\prod_{a\in V_+(i)} (1-\eta_{a\to i})\right], \\
\hat{\Pi}_i^- &= \left[\prod_{a\in V_+(i)} (1-\eta_{a\to i})\right]\left[1-\prod_{a\in V_-(i)} (1-\eta_{a\to i})\right], \\
\hat{\Pi}_i^0 &= \prod_{a\in V(i)} (1-\eta_{a\to i})\,.
\end{aligned}$$

Thus $\hat{\Pi}_i^+$ ($\hat{\Pi}_i^-$) is the non-normalized probability that no neighbor wants i to be FALSE (TRUE) and at least one neighbor of i wants i to be TRUE (FALSE). The normalized counterparts to these are the weights

$$\begin{aligned}
W_i^+ &= \frac{\hat{\Pi}_i^+}{\hat{\Pi}_i^+ + \hat{\Pi}_i^- + \hat{\Pi}_i^0}, \\
W_i^- &= \frac{\hat{\Pi}_i^-}{\hat{\Pi}_i^+ + \hat{\Pi}_i^- + \hat{\Pi}_i^0}, \\
W_i^0 &= 1 - W_i^+ - W_i^-,
\end{aligned}$$

where $W^\pm$ are a measure of how biased the variable i is towards either truth value. The neutral W^0 is the bias towards indifference. In the case of all surveys being 0 all the weights are $1/3$ and survey propagation can give no further information. If this situation arises, the formula is most likely under-constrained and solved easily by a local search algorithm such as simulated annealing [50]. In the case of non-zero surveys, the weights will get differing values and we can use them to fix a variable. We choose the variable with the largest difference between the positive and negative weight and fix it to the value of the larger of them. This gives us a strictly smaller problem that according to numerical experiments is still satisfiable if the original problem was satisfiable [50].

7 Parallel Survey Propagation and Numerical Results

In [50] Braunstein et al. have run survey propagation routinely and collected statistics with variables numbering up to 10^5. They have also done isolated

tests with the number of variables in the order of 10^6 to 10^7 variables. In this thesis I present a parallelized version of their program that is run on up to 10^7 variables. With this modification I am able to give a more precise estimate of the values of the clause to variable ratio α for which survey propagation succeeds and to the complexity of the algorithm as a function of the number of variables.

7.1 A Parallel Version of Survey Propagation

In [51] Manolios and Zhang developed a version of SP that can be run on graphics processing units. My version is based on the code provided by Zecchina et al. [50] at http://users.ictp.it/~zecchina/SP/ and it is run on traditional CPUs. With some rudimentary changes and the addition of the OpenMP shared memory parallelization library I get a speedup by a factor of three in the SP part of the program using four processes and running for $N = 10^5, \alpha = 4.24$. Adding the calls to other helper programs increses the running time by only three seconds compared to the running time of about 30 seconds with four processes and 90 seconds with the original serial version.

My parallel version does not solve the problem of scaling to larger problems completely, since it is still reliant on fitting the whole formula into one memory. It does help in studying SP in greater detail on the upper end of the reachable values of N. The authors of [50] collected detailed statistics of SP up to problem size 10^5. In this thesis I present further observations with N up to 10^6 and some notions on SP with $N \approx 10^7$, which in 2013 is feasible with a modern desktop computer, since the memories installed are rudimentally 16GB, or even 32GB.

The biggest portion of the running time of the program is spent iterating over all the edges and updating the surveys. Since the iteration is done in random order and is repeated until convergence is reached, it can be done in parallel [51]. In the code the surveys for each clause are updated together in a function that does not write to any other memory location. There is a simple loop over the set of clauses that I replaced with one that runs the iterations of the loop on different processors.

7.2 On the Limit of Applicability of SP

From numerical studies with exact brute force methods it is known that the critical value for α, below which most random SAT problems are satisfiable

and above which most are unsatisfiable, is $\alpha \approx 4.267$ [19, 52]. I tested the performance of SP with different values of α below the satisfiability threshold. Clearly SP drops in validity well before the problem becomes actually unsatisfiable, as can be seen from the results in Figure 11. Ideally in SP the procedure is to remove only one variable with each set of fixed point surveys. This is unfeasible with large N. Instead, some percentage f of the remaining variables are fixed. I used the value $f = 0.01$% in these tests. The tests with $N = 50000$ were run 20 times for each value of α and the larger ones were run 10 times for each value of α. The results indicate that the validity region of SP is N dependent so that larger N push SP to work closer to the satisfiability limit. In [21] Parisi claims that the validity limit of SP converges at $\alpha \approx 4.253$ for infinite N, but my data do not corroborate this claim. In fact for $N = 750000$ even at $\alpha = 4.254$ there were satisfying assignments for 90% of the formulas.

The reason for this shift is unclear. One possible explanation would be if the maximum fraction to fix depends on N. Then with larger N and a constant fraction f SP would solve problems with larger α and the phenomenon would disappear if only fixing one spin at a time. The data in this thesis do not allow for checking this conjecture.

7.3 On the Computational Complexity of SP

In [50] the authors write that the number of iterations needed to reach convergence to fixed point surveys does not seem to grow with N or possibly grows as $\ln N$. It is not clear from their data either way. A growth as $\ln N$ would be expected, though, as $\ln N$ is the distance where the effect of removing a variable should decay [19]. Before moving on to the numerical results on this question, some words about the theoretical analysis of the complexity are in order.

Since the algorithm is stochastic, it is not possible to give a precise definition for the computational complexity. However, we are also interested in average case complexity. The majority of the time is spent on updating the surveys and so the number of iterations (each updating all the surveys) is an interesting measure. As noted in [50], the complexity of an iteration is $\mathcal{O}(N)$. The authors suggest that the complexity of the whole algorithm is $\mathcal{O}(N^2 \ln N)$ when fixing just one variable with each set of fixed point surveys. Here the assumption is that the number of iterations needed for convergence grows

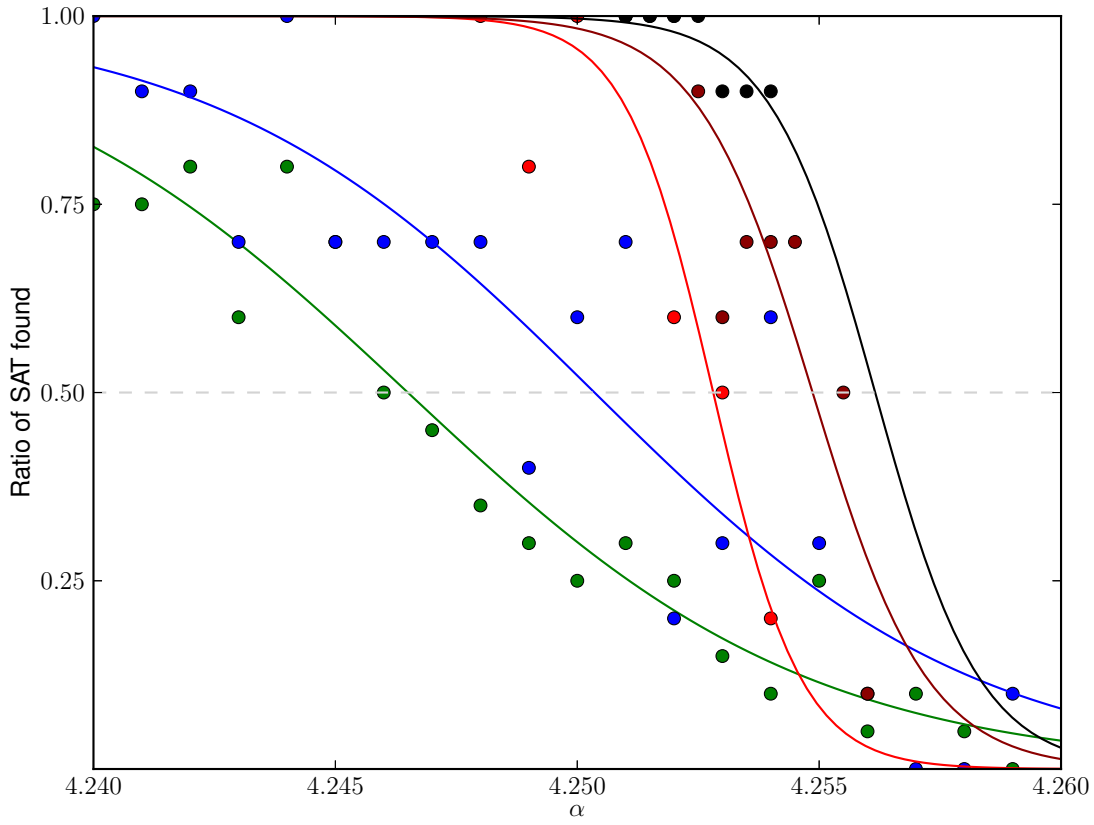

Figure 11: The ratio of succesful SAT assignments found with different values of α. The dots represent data points and the curves are fitted with the following function $\frac{1}{2} - \frac{1}{2}\tanh(a \cdot x + b)$. The curves from left to right are with values of N in 50000, 100000, 200000, 500000, and 750000. The data points that correspond to each curve are plotted with the same color as the curve.

as $\ln N$. The total number of fixed point surveys needed is N. The whole complexity works out to be $\mathcal{O}(N \cdot N \cdot \ln N) = \mathcal{O}\left(N^2 \ln N\right)$.

An order of magnitude improvement is achieved by fixing a fraction of the most biased spins at a time, where Braunstein et al. suggest the complexity to be $\mathcal{O}\left(N \ln^2 N\right)$. This is incorrect. The second logarithm is said to come from the need to sort the biases in order to find the needed fraction of most biased variables. The complexity of sorting a list of N objects is indeed $\mathcal{O}\left(N \ln N\right)$, but the sorting is done after iterating for the fixed point set of surveys. In total the complexity becomes $\mathcal{O}\left(N \ln N + N \ln N\right) = \mathcal{O}\left(N \ln N\right)$. The repetition over fixing a fraction of variables does not show in the complexity, since it is done a number of times independent of N.

The question about whether the number of iterations required to reach fixed point surveys grows as $\ln N$ or is constant is almost a moot point. It does not have any effect on the total computational complexity of SP which will be in any case $\mathcal{O}\left(N \ln N\right)$. On the other hand, for practical purposes the question might be relevant, because the constant factor in the complexity of survey updating is substantially larger than in sorting a simple list of numbers. What is meant by a constant factor is that an algorithm might take time proportional to N or time proportional to, e.g., $10^{10}N$. Both are $\mathcal{O}\left(N\right)$, but the latter takes longer, because of the constant before N.

The numerics show clearly that at least one parameter has a significant impact on how many iterations are needed: the fraction of spins to fix after each convergence, as shown in Figure 12. In fact there seems to be an inverse dependence between the fraction to fix and the total number of iterations. This is something that the Braunstein et al. [50] do not consider when trying to deduce the complexity from data. Instead, they did experiments with the fraction f following a geometric progression where $f \in [4\%..0.125\%]$ were tried in order from largest to smallest. The process was stopped when a correct assignment was found. This is a valid approach when trying to determine whether SP succeeds at all, since SP succeeds more likely with f small and takes a shorter time with f large. However, it is not suitable for finding out the dependence on N of the number of iterations needed. Braunstein et al. averaged over succesful runs with different fractions f and got numbers that did not really hold any information on the complexity. Figure 13 shows my results with a constant f. It shows that there is a weak dependence between the number of variables and it seemingly takes the form of a logarithm as would be expected. Further research is needed to verify this.

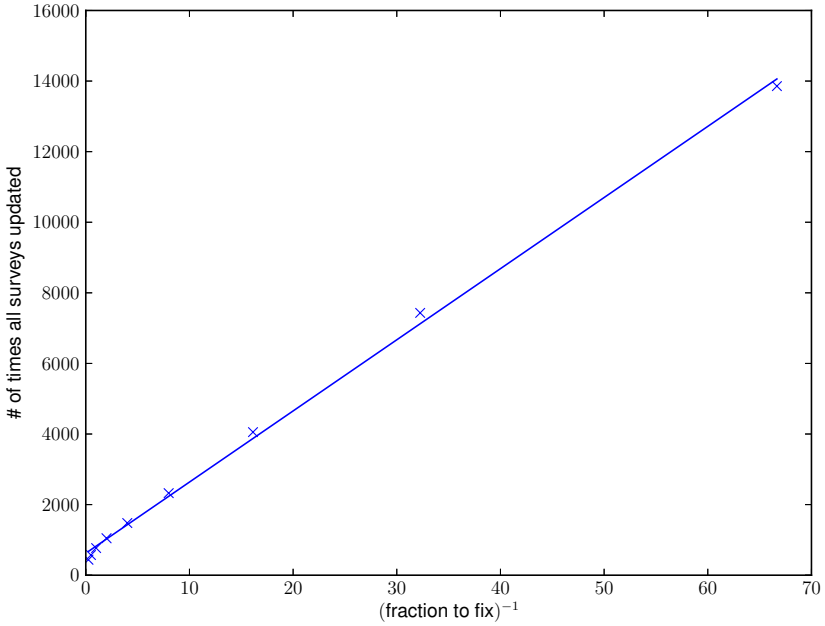

Figure 12: The number of iterations (updates to every survey) needed as a function of the inverse of the ratio of variables to fix per convergence.

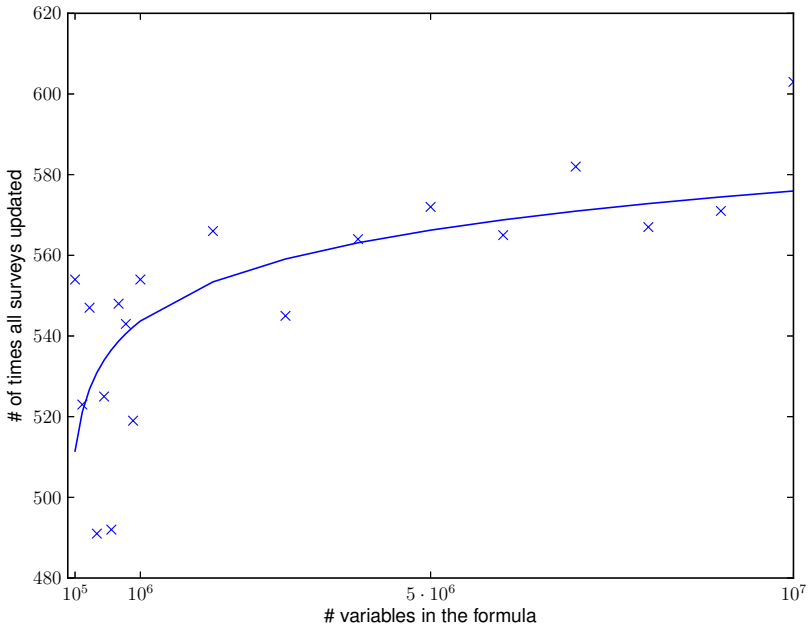

Figure 13: The number of iterations as a function of the number of variables. There is a slight growth with N that seems to resemble a logarithm, but the data are not sufficient to ascertain the fact.

8 Summary and Conclusions

The junction of statistical physics and computational complexity is exciting. In the physics side the main area of interest is the theory of spin glasses; in the computational complexity side boolean satisfiability. The concept of a spin glass, which was first introduced to explain actual physical observations, quickly became a research subject on its own. It has long been studied purely for its assets as an interesting theoretical model. After the major contributions of Edwards and Anderson [24] as well as Sherrington and Kirkpatrick [23], both in 1975, there has been tremendous theoretical progress in the field: the fully replica symmetry breaking solution and physical interpretation by Parisi published in several articles [32, 33, 34, 53] over 1979-1983 that achieved perfect harmony with numerical results; the cavity method, which allowed for deriving much the same results using a different kind of reasoning in 1986 [40]; the application of the cavity method to the Bethe lattice in 2001 [25, 44]; and the expansion of spin glass methods to different fields of science. The progress and results of physicists have prompted also mathematicians to look into the derivations to see whether they could be turned mathematically sound. The efforts of mathematicians have provided some analytical bounds and conditions for rigorously proving some of the results [28, 29]. As these efforts are tangential to the aim of this thesis, they have not been studied here.

In this thesis, a cursory glance to the basic concepts of computational complexity was given. The big O notation of time complexity and the classes of P and NP were introduced, and the reasons why these classes are considered so important were elaborated. The problem of boolean satisfiability was introduced. The work also included an overview on how the similarity of the spin glass on a Bethe lattice to boolean variables on a factor graph allowed for the use of the cavity method [17, 19]. With the cavity method we found the iteration equations for updating surveys on magnetic fields and with these we got a numerical algorithm for solving single instances of satisfiability quickly [19, 50]. The relevant theories are recent: the cavity analysis of boolean satisfiability was introduced in 2002; the survey propagation algorithm was published only in 2005 in a form accessible also without a deep understanding of the cavity formalism. As this is a young field, some of the numerical details are still incomplete.

Using the code provided by Braunstein et al. [50] as a basis, I extended the

program to work in parallel. This made it possible to gain further insight by studying formulas with bigger N. My work provides more detailed information than the previous literature for some properties of SP. Firstly, the data in this thesis give a more detailed picture of the scaling behavior in the number of iterations needed for convergence and support the hypothesis that the scaling should be of order $\ln N$. Secondly, the nature of the phase transition where SP stops working was described in detail, including the curious phenomenon where the location of the transition is dependent on N. This thesis went beyond the previous research in the level of detail in these areas and my parallel version is faster than the original implementation.

References

[1] Franz Mandl. *Statistical Physics*. Manchester Physics Series. Wiley, 1988.

[2] Stephen Brush. History of the Lenz-Ising Model. *Reviews of Modern Physics*, 39(4):883–893, 1967.

[3] Kurt Binder and Peter Young. Spin glasses: Experimental facts, theoretical concepts, and open questions. *Reviews of Modern physics*, 58(4):801, 1986.

[4] Konrad Fischer and John Hertz. *Spin Glasses*. Cambridge University Press, 1993.

[5] Kei Yosida. Magnetic properties of Cu-Mn alloys. *Physical Review*, 106(5):893, 1957.

[6] Tadao Kasuya. A theory of metallic ferro-and antiferromagnetism on Zener's model. *Progress of Theoretical Physics*, 16(1):45–57, 1956.

[7] Melvin Ruderman and Charles Kittel. Indirect exchange coupling of nuclear magnetic moments by conduction electrons. *Physical Review*, 96(1):99, 1954.

[8] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2006.

[9] Stephen Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, pages 151–158. ACM, 1971.

[10] Richard Karp. *Reducibility Among Combinatorial Problems*. Springer, 1972.

[11] Lance Fortnow. The status of the P versus NP problem. *Communications of the ACM*, 52(9):78–86, 2009.

[12] Gareth Janacek and Mark Close. *Mathematics for Computer Scientists*. Bookboon, 2009.

[13] Scott Kirkpatrick and Bart Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264(5163):1297–1301, 1994.

[14] James Crawford and Larry Auton. Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence*, 81(1):31–57, 1996.

[15] Scott Kirkpatrick, Daniel Gelatt, and Mario Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

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

[17] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[18] Allon Percus, Gabriel Istrate, and Cristopher Moore. *Computational Complexity and Statistical Physics*. OUP USA, 2006.

[19] Marc Mézard and Riccardo Zecchina. Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Physical Review E*, 66(5):056126, 2002.

[20] Bob Kanefsky, William Taylor, and Peter Cheeseman. Where the really hard problems are. In *Proceedings of IJCAI*, volume 91, pages 163–169, 1991.

[21] Giorgio Parisi. Some remarks on the survey decimation algorithm for K-satisfiability. *arXiv preprint cs/0301015*, 2003.

[22] Giorgio Parisi. A backtracking survey propagation algorithm for K-satisfiability. *arXiv preprint cond-mat/0308510*, 2003.

[23] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792, 1975.

[24] Samuel Edwards and Phil Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5(5):965, 1975.

[25] Marc Mézard and Giorgio Parisi. The Bethe lattice spin glass revisited. *The European Physical Journal B: Condensed Matter and Complex Systems*, 20(2):217–233, 2001.

[26] Marc Mézard, Giorgio Parisi, and Miguel Virasoro. *Spin Glass Theory and Beyond*. World Scientific, 1987.

[27] Scott Kirkpatrick and David Sherrington. Infinite-ranged models of spin-glasses. *Phys. Rev. B*, 17(11):4384–4403, Jun 1978.

[28] Michel Talagrand. *Spin Glasses: a Challenge for Mathematicians: Cavity and Mean Field Models*. Springer, 2003.

[29] Erwin Bolthausen. *Random Bedia and Spin Glasses: an Introduction into Some Mathematical Results and Problems*. Springer, 2007.

[30] André Blandin, Marc Gabay, and Thomas Garel. On the mean-field theory of spin glasses. *Journal of Physics C: Solid State Physics*, 13(3):403, 1980.

[31] Debashish Chowdhury. *Spin Glasses and Other Frustrated Systems*. World Scientific, 1986.

[32] Giorgio Parisi. The order parameter for spin glasses: a function on the interval 0-1. *Journal of Physics A: Mathematical and General*, 13(3):1101, 1979.

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

[34] Giorgio Parisi. Magnetic properties of spin glasses in a new mean field theory. *Journal of Physics A: Mathematical and General*, 13(5):1887, 1980.

[35] Marc Mézard, Giorgio Parisi, Nicolas Sourlas, G. Toulouse, and Miguel Virasoro. Replica symmetry breaking and the nature of the spin glass phase. *Journal de Physique*, 45(5):843–854, 1984.

[36] Marc Mézard, Giorgio Parisi, Nicolas Sourlas, G. Toulouse, and Miguel Virasoro. Nature of the spin-glass phase. *Physical review letters*, 52(13):1156–1159, 1984.

[37] Giorgio Parisi. A sequence of approximated solutions to the SK model for spin glasses. *Journal of Physics A: Mathematical and General*, 13(4):115–121, 1979.

[38] Jean-Philippe Bouchaud and Marc Mézard. Universality classes for extreme-value statistics. *Journal of Physics A: Mathematical and General*, 30(23):7997, 1997.

[39] Andrea Montanari, Giorgio Parisi, and Federico Ricci-Tersenghi. Instability of one-step replica-symmetry-broken phase in satisfiability problems. *Journal of Physics A: Mathematical and General*, 37(6):2073, 2004.

[40] Marc Mézard, Giorgio Parisi, and Miguel Virasoro. SK model: The replica solution without replicas. *Europhysics Letters*, 1(2):77–82, 1986.

[41] Marc Mézard and Giorgio Parisi. Mean-field theory of randomly frustrated systems with finite connectivity. *Europhysics Letters*, 3(10):1067, 1987.

[42] Nicholas Wormald. Models of random regular graphs. *London Mathematical Society Lecture Note Series*, pages 239–298, 1999.

[43] Brendan McKay and Nicholas Wormald. Uniform generation of random regular graphs of moderate degree. *Journal of Algorithms*, 11(1):52–67, 1990.

[44] Marc Mézard and Giorgio Parisi. The cavity method at zero temperature. *Journal of Statistical Physics*, 111(1-2):1–34, 2003.

[45] Frank Kschischang, Brendan Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.

[46] Alfredo Braunstein and Riccardo Zecchina. Survey propagation as local equilibrium equations. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(06):–06007, 2004.

[47] Andrea Montanari, Federico Ricci-Tersenghi, and Guilhem Semerjian. Clusters of solutions and replica symmetry breaking in random k-satisfiability. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(04):P04004, 2008.

[48] Haixia Jia, Cris Moore, and Bart Selman. From spin glasses to hard satisfiable formulas. In *Theory and Applications of Satisfiability Testing*, pages 199–210, 2005.

[49] Harri Haanpää, Matti Järvisalo, Petteri Kaski, and Ilkka Niemelä. Hard satisfiable clause sets for benchmarking equivalence reasoning techniques. *JSAT*, 2(1-4):27–46, 2006.

[50] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures & Algorithms*, 27(2):201–226, 2005.

[51] Panagiotis Manolios and Yimin Zhang. Implementing survey propagation on graphics processing units. In *Theory and Applications of Satisfiability Testing-SAT 2006*, pages 311–324. Springer, 2006.

[52] Stephan Mertens, Marc Mézard, and Riccardo Zecchina. Threshold values of random K-SAT from the cavity method. *Random Structures & Algorithms*, 28(3):340–373, 2006.

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