Jül-4334

Jülich Supercomputing Centre (JSC)

Classical and quantum statistical analysis of a constraint satisfaction problem

Markus Peschina

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
FORSCHUNGSZENTRUM

Classical and quantum statistical analysis of a constraint satisfaction problem

Markus Peschina

Berichte des Forschungszentrums Jülich; 4334
ISSN 0944-2952
Jülich Supercomputing Centre (JSC)
Jül-4334

Vollständig frei verfügbar im Internet auf dem Jülicher Open Access Server (JUWEL)
unter http://www.fz-juelich.de/zb/juwel

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek, Verlag
D-52425 Jülich · Bundesrepublik Deutschland
Telefon: 02461 61-6123 · Telefax: 02461 61-6103 · e-mail: zb-publikation@fz-juelich.de

Acknowledgements

I am grateful to many people for help, both direct and indirect, in writing this diploma thesis. First of all, I want to thank my advisor from the research centre in Juelich, PD Dr. Thomas Neuhaus, for his guidance and many fruitful discussions. He was never tired of explaining me all the details of the methods that I have used.

I also want to thank Prof. Dr. Kristel Michielsen and Prof. Dr. Hans De Raedt for patiently answering my questions. Both were always available and helped me with their own programs as well as organizational issues.

Of course I thank my parents and my family for their continuous support. In particular I thank Claudia Nowoczyn for her help. I feel grateful for the encouragement of Stephanie Mangold during my time in the research centre in Juelich.

I thank the reasearch group Quantum Information Processing for the interesting conversations during lunch.

Finally and especially I have to thank Prof. Hartmut Wittig, Prof. Dr. Dr. Thomas Lippert and PD Dr. Johannes Schneider, since without their aid the present work would not have been possible.

Contents

Chapter 1

Adiabatic quantum computing

The use of quantum computers promises higher computational power than the use of a classical computer. Typical examples of quantum algorithms are Shor's algorithm [1] and Grover's quantum search algorithm [2]. Shor's algorithm, developed in 1994, is able to give an exponential speedup in finding prime factors of large numbers. Grover's algorithm is able to find an element of an unstructured database of n elements in a time of the order $\mathcal{O}(\sqrt{n})$, which is faster than the linear time $\mathcal{O}(n)$ that a classical computer would consume in performing this task. Both algorithms have been implemented physically on a quantum system [3–5]. For an overview of which physical quantum systems are most promising to construct a quantum computer, we refer to reference [6] for further reading. In what follows we explain why, in principle, a quantum computer can do calculations faster than a classical one. We use the so-called network model [7], which requires the definition of a qubit. Let $|\psi\rangle$ be a state vector of a two-level quantum system. We write

$$|\psi\rangle = a|0\rangle + b|1\rangle, \tag{1.1}$$

where $|0\rangle$ and $|1\rangle$ are the two states spanning the corresponding Hilbert space $\mathcal{H}$. We treat $|\psi\rangle$ as a qubit, e.g. the smallest quantity of information that can be processed by a quantum computer. A measurement of $|\psi\rangle$ provides us with the outcome of a

calculation, which in this case corresponds to either $|0\rangle$ or $|1\rangle$. So, according to the rules of quantum mechanics every quantum algorithm needs to be probabilistic. The task is to evaluate a function $f(x) \rightarrow y$ where $x \in [0,1]^n$, $y \in [0,1]$ and n is the size of the system, e.g. the number of qubits, so that the state of the system is given by $|x\rangle = |\psi_1\rangle \otimes \ldots \otimes |\psi_n\rangle \in \mathcal{H} \otimes \ldots \otimes \mathcal{H}$. The operation of a unitary operator $\hat{U}_{f(x)}$ reads

$$\hat{U}_{f(x)}|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle, \tag{1.2}$$

where $\oplus$ is the addition modulo 2 operator. When we prepare the system in an equal superposition of all 2^n possible basis states of the Hilbert space $\mathcal{H} \otimes \ldots \otimes \mathcal{H}$, we are able to perform 2^n evaluations of the function $f(x)$ by applying the operator $\hat{U}_{f(x)}$ only once:

$$\hat{U}_{f(x)}|x\rangle|0\rangle = \sum_{i=0}^{i=2^n-1} |i\rangle|f(i)\rangle, \tag{1.3}$$

where i is the binary representation of integer numbers up to $2^n - 1$. This approach is called the network model because operations on qubits are performed with unitary operators and one can imagine a sequence of unitary operators acting on qubits as a network of gates acting on classical bits. It is difficult to develop algorithms within this model as some of the basic features of bits cannot be ported to qubits. For example, we are not able to copy an arbitrary state of a qubit to another one [8]. When we perform a measurement of the state $|x\rangle = \sum_{i=0}^{i=2^n-1} |i\rangle$ before the application of $\hat{U}_{f(x)}$, we project it to one possible basis state and as a result we will no longer obtain a superposition of all possible values of $f(x)$.

In 1998 Kadowaki and Nishimori proposed a technique called quantum annealing [9] to get a faster convergence than simulated annealing in optimization problems. In quantum annealing quantum fluctuations are used in addition to thermal fluctuations to find the optimal state of the system. Quantum mechanically, disorder is introduced via a Hamiltonian that does not commute with the Hamiltonian of the optimization problem and whose ground state is a superposition of all eigenstates of the problem

Hamiltonian. The quantum annealing procedure slowly removes the disorder so that the system will settle in one of its low energy states. The quantum adiabatic (QA) algorithm [10], introduced by Fahri et al. in 2001, is a similar technique to solve combinatorial optimization problems. The basic idea is to slightly modify the Hamiltonian of the system from a simple one, called the *driver* Hamiltonian H_D, to the one corresponding to the *problem* Hamiltonian H_P. The problem Hamiltonian H_P encodes the solution of a given problem in its ground state. The algorithm starts from the ground state of the driver Hamiltonian H_D. The quantum adiabatic theorem [11] ensures that the system stays in the instantaneous ground state when we modify its Hamiltonian H slowly enough compared with the transition times of the spectrum of H. This is the main difference compared to quantum annealing, for which the system is not in its instantaneous ground state during the whole algorithm. We write the QA Hamiltonian as

$$H(t) = (1 - \lambda(t))H_D + \lambda(t)H_P, \tag{1.4}$$

where $\lambda(t) \in [0, 1]$ denotes the adiabatic control parameter. To hold the system in the instantaneous ground state, $\lambda(t)$ needs to be a slow varying function. In chapter 5 we demonstrate what this exactly means. According to the adiabatic theorem, at the end of the QA algorithm the ground state of the problem Hamiltonian H_P is reached with high probability and a measurement can be performed. This ground state encodes the solution of the problem. The advantage of the QA algorithm is its simplicity. We are potentially able to compute many problems with an adiabatic quantum computer since we only need to translate the problem to a Hamiltonian operator. We do not need to consider qubits and their networks. Furthermore, a universal QA algorithm is equivalent to the gate model, as one can map both models onto each other with polynomial effort [12]. The term *universal* means that we do not necessarily have to consider optimization problems, which have diagonal Hamiltonian operators. A disadvantage of adiabatic quantum computing is that it is not always clear how to implement the specific Hamiltonian on a physical device.

The present work is inspired by the work of A.P. Young on the Exact Cover problem [13, 14]. The scaling of the mass gap of the QA algorithm for the Exact cover problem first seemed to be polynomial up to $N = 128$ [13], but more recent work [14] makes this claim obsolete because of the observation of a first order quantum phase transition in some realizations. Tunneling effects at first order phase transitions correspond to exponentially small mass gaps so that the whole algorithm would scale exponentially, whereas second order quantum phase transitions are believed to yield polynomial scaling of the mass gap [15]. The dependence of the runtime of the QA algorithm on the mass gap will be established in chapter 5.

The structure of the present work is as follows. In chapter 2 we give an introduction to the 3-SAT problem from a computer science as well as from a physical point of view. We also discuss the theory of complexity and the P=NP problem [16]. In chapter 3 we explain how we generate special realizations for the 3-SAT problem. In chapter 4 we describe our methods and results, concerning the classical 3-SAT problem. The main idea is first to try to understand the physical properties of the classical system before taking into account any quantum effects. For this task we employ tools of statistical physics and examine thermodynamic properties of the 3-SAT problem as a physical system. We also try to understand the hardness of the problem for our special realizations. Afterwards we continue with the quantized version of the problem. In chapter 5 we first establish the dependence of the runtime on the mass gap and introduce the three simulation methods we use followed by a presentation of the results. In chapter 6 we present our conclusions and we describe some possible topics for future research.

Chapter 2

Satisfiability Problems

The SATisfiability problem can be seen as the root problem of complexity theory [17]. It was the first problem proven to be NP-complete by Cook in 1971 [18]. This opened a way for the identification of many other problems to be NP-complete and to find a reduction in polynomial time of these problems to the SAT problem. Unless $P = NP$ all of them have shown exponential worst-case complexity. There is a wide range of practical applications for SAT problems: From crosstalk noise prediction in integrated circuits [19] to artificial intelligence planning [20] and software model checking [21]. In general there are two different ways of solving all these problems: backtracking [22] and local search algorithms [23]. Optimizing both methods is a field of extensive research and so both are already very advanced right now. By taking advantage of several heuristics they can handle of the order of 100000 variables for real world problems. In the present work we use an algorithm named zChaff to compare our methods to existing solvers and to check the satisfiability of our instances. The program zChaff is provided by Princeton University [24].

The early work of physicists on the topic of SAT covers, for example, the ground state entropy [25] and the SAT/UNSAT phase transition [26]. There are also many theoretical

papers, written by physicists, concerning the random SAT problem [27–29]. By use of methods from statistical physics, they calculate many properties and physical quantities of the system. Throughout the present work we restrict ourselves to the 3-SAT problem, which is a special case of the SAT problem. This chapter is organized as follows: In section 2.1 we give the definition of the 3-SAT problem that is often used in computer science. Its translation to physical problems is discussed in section 2.2. In section 2.3 we explain the difference between so-called P, NP and NP-complete problems and discuss the complexity of various algorithms to solve 3-SAT problems.

2.1 Definition of the 3-SAT problem

We consider N boolean variables x_i, each taking only the values

$$x_i = 0 \ (\equiv \text{FALSE}) \qquad \text{or} \qquad x_i = 1 \ (\equiv \text{TRUE}), \qquad i = 1 \dots N. \tag{2.1}$$

It is convenient to think of them as a vector

$$\vec{x} = (x_1, x_2, \dots, x_N)^T \quad \in \quad \{0;1\}^N. \tag{2.2}$$

For the SAT problem three logical operators are used

NOT:	$\overline{x_i}$;	$\overline{1} = 0, \overline{0} = 1$
AND:	$\wedge$;	$x_i \wedge x_j = 1$ if and only if $x_i = 1$ and $x_j = 1$
OR:	$\vee$;	$x_i \vee x_j = 1$ if at least one of x_i, x_j is true

to connect the variables. A *literal* is either a variable x_i or its negation $\overline{x_i}$. We now form a clause

$$C_i = (\overline{x_j} \vee x_k \vee x_l), \tag{2.3}$$

which is simply the logical OR of 3 literals (or an arbitrary number in the case of general SAT problems). A formula $\mathcal{F}$ is the logical AND of M clauses

$$\mathcal{F} = C_1 \wedge C_2 \wedge \cdots \wedge C_M. \tag{2.4}$$

We call a formula *satisfiable* if there is at least one *assignment* for the values of its variables so that the formula becomes true. Now we can define the 3-SAT problem as follows

Definition 1 (3-SAT)**.** *For a given formula $\mathcal{F}$ with 3 literals in each clause, is there a satisfying assignment?*

Throughout this work we call a specific formula with given clauses a 3-SAT *instance*.

2.2 Physical description of 3-SAT

We are able to map the 3-SAT problem on a physical system. The idea is to use a Hamilton operator H to encode the satisfying assignments into ground states. In that sense we define the physical energy as the number of unsatisfied clauses and the Hamiltonian H counts their number. Boolean variables $x_i = 0, 1$ are mapped to Ising spins $s_i = 2x_i - 1 = \pm 1$. We define the sign matrix as

$$\varepsilon_{ij} \in \{-1; 1\}^{M \times 3} \qquad i = 1, \ldots, M \quad j = 1, 2, 3 \tag{2.5}$$

where i denotes the number of the clause and j the position of the variables in that clause. If a variable is negated, then $\varepsilon_{ij} = -1$ else $\varepsilon_{ij} = 1$. We define the clause matrix as

$$S_{ij} = s_k \qquad i = 1, \ldots, M \quad j = 1, 2, 3 \tag{2.6}$$

if variable x_k is at position j in clause C_i. The variable exchange symmetry of one clause, see Eq. (2.3), is not broken, but we have to adjust both matrices ε and S simultaneously.

Now we are able to form a component of H for one particular clause C_i. Let l_k denote a literal, taking the value x_k or $\bar{x}_k$

$$\begin{aligned} C_i = (l_1 \vee l_2 \vee l_3) \hat{=} & \left(\frac{\varepsilon_{i1}s_1 + 1}{2}\right)\left(\frac{\varepsilon_{i2}s_2 + 1}{2}\right)\left(\frac{\varepsilon_{i3}s_3 + 1}{2}\right) \\ = & \frac{1}{8}[\varepsilon_{i1}\varepsilon_{i2}\varepsilon_{i3}s_1s_2s_3 + \varepsilon_{i2}\varepsilon_{i3}s_2s_3 + \varepsilon_{i1}\varepsilon_{i3}s_1s_3 \\ & + \varepsilon_{i1}\varepsilon_{i2}s_1s_2 + \varepsilon_{i1}s_1 + \varepsilon_{i2}s_2 + 1] \\ = & \, h(\varepsilon_{i1}s_1,\, \varepsilon_{i2}s_2,\, \varepsilon_{i3}s_3). \end{aligned} \tag{2.7}$$

There are only eight possibilities for the variables $\varepsilon_{ij}s_j$ of the function $h(\varepsilon_{i1}s_1,\, \varepsilon_{i2}s_2,\, \varepsilon_{i3}s_3)$ with a fixed sign matrix. Only one, namely $\varepsilon_{ij}s_j = -1 \quad \forall j$, gives $h = 1$. So the ground state of one clause is 7-fold degenerated. The entire Hamiltonian corresponding to a given instance is then written as

$$H = \sum_{i=1}^{M} h(\varepsilon_{i1}S_{i1},\, \varepsilon_{i2}S_{i2},\, \varepsilon_{i3}S_{i3}), \tag{2.8}$$

and we can reformulate the problem as

Definition 2 (3-SAT)**.** *For a given Hamilton operator H, is there a ground state with $E = 0$?*

If we are only interested in finding a solution, this formulation is equal to Def. (1). Note that we call the physical 3-SAT implementation of a given instance a *realization*.

2.3 Algorithm Complexity

The 3-SAT problem belongs to the class of NP-complete problems [17], and moreover it was the first problem to be classified as such. With $\mathcal{O}(n)$ we denote an upper bound to the resources that a specific algorithm will consume for a given input of length n. Throughout this work we focus on the resource *time*, which means that we are interested

in how much time an algorithm needs or, to get rid of the unit, how many steps it needs to perform to find a solution. All problems with a complexity $\mathcal{O}(n^k)$ and with an arbitrary, but finite, k are solvable in polynomial time and therefore belong to the class of P (polynomial) problems. In complexity theory the class of P problems contains all decision problems that are solvable in polynomial time by a deterministic Turing machine, which can be seen as a classical computer. The other problem class of interest is called NP (Nondeterministic Polynomial), containing all decision problems that are solvable in polynomial time by a nondeterministic Turing machine. As we do not want to cover the topic of Turing machines in this work, we use another definition for NP problems: All decision problems whose solutions can be verified in polynomial time belong to the class NP. Trivially P is a subset of NP. Many problems in NP have an exponential runtime $\mathcal{O}(\mathrm{e}^n)$, and no polynomial algorithm is known to solve this type of problems. However, NP problems are not proven to have an exponential runtime and an open question is whether P=NP or not. Another problem class is called NP-complete, which is a subclass of NP with some special characteristics [17]. Each problem that is NP-complete can be mapped onto another NP-complete problem with an effort that is only polynomial in the size of the problem. Since the SAT problem is NP-complete and needs an exponential runtime we can deduce that all NP-complete problems are exponentially hard. Even more, problems that are NP-complete can be seen as the hardest problems of the class NP. Now it is clear why the P=NP question is important, because if so, we can be sure that all the hard problems that belong to the class NP-complete can be solved efficiently.

Up to now finding a solution for the 3-SAT problem takes an exponential amount of time. Let N denote the number of variables and $\mathrm{t}(N)$ the time needed to find one solution, then

$$t(N) \propto \mathrm{e}^{cN} \,. \tag{2.9}$$

The factor c determines the exponential complexity of the algorithm. Checking every

possible assignment of N Boolean variables is the simplest way to find a solution of a given 3-SAT instance. However, this consumes the maximum amount of time to find one solution, namely

$$t(N) \propto 2^N = \mathrm{e}^{\ln(2)N}\,. \tag{2.10}$$

Until now there is no algorithm which is capable of solving the 3-SAT problem in polynomial time. But many approaches exist to lower the constant c of the exponential function. When we use Grover's quantum search algorithm to find a unique solution, we can speed up the calculations, namely

$$t(N) \propto 2^{\frac{1}{2}N} = \mathrm{e}^{\frac{\ln(2)}{2}N}, \tag{2.11}$$

but we cannot raise the complexity class to a polynomial one. Grover's algorithm can only lower the constant from $c_{\text{naive}} = \ln(2)$ to $c_{\text{Grover}} = \frac{\ln(2)}{2}$. To illustrate what exponential hardness means, we take a look at a small example. Consider a normal computer which can compute 10^9 operations per second and a fast one computing 10^{10} operations per second, thus ten times faster. We want to solve an instance of 3-SAT with $N = 80$ variables and to compare the times needed to perform this task (see table (2.1)). We consider the very naive algorithm, Grover's quantum search algorithm and a few probabilistic algorithms [30], denoted by their year of development. We only consider the exponential runtime of the algorithms and cut out any constants or polynomial contributions. From table (2.1) we see that the computer which is ten times faster can in one hour's time only solve problems with a few extra variables compared to the problems the slow computer can solve.

Algorithm	$t_{\text{normal}}[\text{s}]$	$t_{\text{fast}}[\text{s}]$	N^*_{normal}	N^*_{fast}	c
naive	1.21^{15}	1.21^{14}	41	44	0.693
Grover	1100	110	83	89	0.347
1997	11.1^6	1.11^6	62	67	0.462
1999	9.9	1	100	108	0.288
2006	5.3	0.5	103	111	0.280

Table 2.1: Comparison between slow and fast computers running various algorithms. t_{normal} (t_{fast}): time to solve a 3-SAT problem with $N = 80$ variables on a computer with 10^9 (10^{10}) operations per second; N^*_{normal} (N^*_{fast}): number of variables that can be solved with the given algorithm on the normal (fast) computer in one hour; c: time to find one solution is given by Eq. (2.9).

Chapter 3

Instance Generation

There exist many methods to obtain 3-SAT realizations, or in general SAT realizations. Real World problems seem to be easier to solve than so-called tailored ones. In this case tailoring means the artificial construction of a SAT problem according to some specific rules. It is crucial to construct hard but solvable SAT realizations in order to have suitable testcases for new algorithms. So there is a whole community providing standardized 3-SAT packages [31]. There exist many different approaches to construct realizations with specific details or according to specific distributions. One obvious possibility to construct solvable 3-SAT realizations is to use complete solvers as a filter and to keep only the solvable realizations. However, this method renders the realization generation itself an exponential hard problem. There are ideas to reduce other hard problems to SAT, like the factoring problem [32] or graph coloring [33] so that their realizations could be used. Another possibility would be to make use of physical models like spin glasses [34]. In the latter case the realization generation becomes very interesting when physical properties of a class of realizations could be derived analytically [35].

We have special requirements for our realizations. First, we only use realizations with one unique solution and we call this USA (Unique Satisfying Assignment). Second, we

want to know this solution in order to check whether our algorithm can find it. Third, we want realizations that are "hard" to solve, this means that the computer needs an exponential amount of time to find the solution.

Random 3-SAT

The simplest way of producing a 3-SAT realization is to generate a complete randomized one. We first fill the clause with three random chosen variables and then assign a negation to each of them with probability one half. This scheme is repeated for all M clauses to achieve the random 3-SAT realization. The coordination number η counts the number of clauses that a variable belongs to. For methods that fill the clauses with randomized variables, the mean of the coordination number for a whole ensemble is $\langle\eta\rangle = 3r = 3M/N$ with a Gaussian shape (according to the central limit theorem). This is simply because by creating the realization we distribute N variables among $3M$ possible entries in the clauses. The coordination number for the variables with/without a negation η_+/η_- also follows a normal distribution, but with $\langle\eta_+\rangle = \langle\eta_-\rangle = 3r/2$.

Forced 3-SAT

In a forced generation method at least one of the solutions of the realization is known prior to the construction of the realization. So the realization is *forced* to have a particular solution. We can choose an arbitrary distribution of "0" and "1" since we can compensate with the distribution of negations[1]. The probability for the occurrence of a "0" or "1" is set to $p = 0.5$. We first give an example of a simple forced generation

[1]By a local gauge transformation of the 3-SAT realization one solution $\vec{x}$ can be transformed to any other one $\vec{x}^*$. One has to exchange x_i and $\overline{x_i}$ $\forall i$ with $x_i^* = 0$ [35]

method and afterwards explain the method that we finally used and which we will refer to as *forced* method.

Simple forced method

We want to obtain a realization with N variables and M clauses, and we first generate a bit vector of length N, which represents the known solution

$$\vec{x} = (x_1, x_2, \ldots, x_N)^T \quad \in \quad \{0;1\}^N. \tag{3.1}$$

Then we start filling the first position of the first N clauses of a formula $\mathcal{F}$ with M clauses with our solution $\vec{x}$, so that the corresponding 3-SAT realization will evaluate to TRUE under the assignment of $\vec{x}$. This means whenever $x_i = 0$ we have to negate the variable v_i in the matrix. With $\vec{v}$ we denote variables that have not yet been assigned to a value 0 or 1, in contrast to $\vec{x}$, which is a fixed Boolean vector. If $M > N$ we randomly pick variables to fill up the first position of clauses $N+1, \ldots, M$. Since in the present work only realizations in the range $3.5 < r < 8$ are considered, $M > N$ holds for all cases.

$$\begin{aligned}
\mathcal{F} = \; & (\quad v_1 \quad \vee \; . \; \vee \; . \;) \; \wedge \\
& (\quad \overline{v_2} \quad \vee \; . \; \vee \; . \;) \; \wedge \\
& (\quad \vdots \quad \vee \; . \; \vee \; . \;) \; \wedge \\
& (\quad v_N \quad \vee \; . \; \vee \; . \;) \; \wedge \\
& (\; v_{\text{random}} \; \vee \; . \; \vee \; . \;) \; \wedge \\
& (\quad \vdots \quad \vee \; . \; \vee \; . \;) \; \wedge \\
& (\; v_{\text{random}} \; \vee \; . \; \vee \; . \;) \, .
\end{aligned} \tag{3.2}$$

In the next step we fill up the second and third positions of the clauses with randomly chosen variables so that there is no double allocation in one clause. These variables should evaluate to FALSE under the assignment of $\vec{x}$, so for all $x_i = 1$ we have to negate the corresponding variable v_i. Now that we created $\mathcal{F}$, we pass each realization to a

filter with level l because we search for USA realizations. The idea is to first check for each bit assignment with Hamming distance l if it is a solution as well and then to block those realizations. Finally we mix up the variables inside each clause.

The coordination number η reveals that this method is not convenient, because one can easily extract the solution from the histogram of η for a particular realization. Just as for the random realization method, $\langle\eta\rangle = 3r$. Remember that we choose the negations of the variables M times according to the solution and $2M$ times inverse to the solution. So we first need to count the negations of a variable and then decide whether it is FALSE or TRUE in $\vec{x}$. Therefore the distribution of $\langle\eta_+\rangle$ and $\langle\eta_-\rangle$ at very high ($r > 10$) in Fig. 3.1a shows two peaks, one at $\langle\tilde{\eta}_+\rangle = 1/3$ and one at $\langle\tilde{\eta}_-\rangle = 2/3$, whereas $\tilde{\eta} = \eta/3r$ denotes the rescaled coordination number.

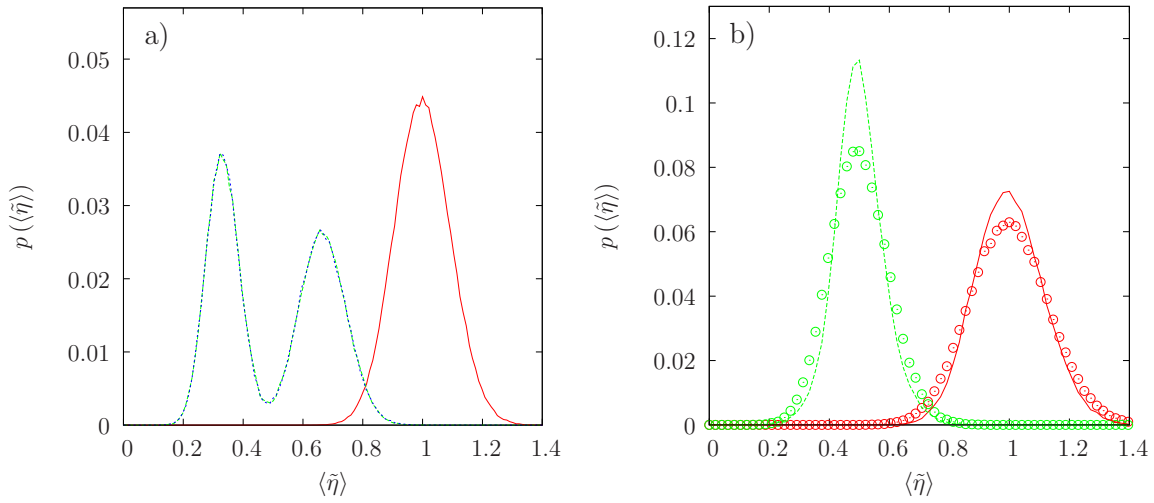

Figure 3.1: Distribution of coordination number $p(\langle\tilde{\eta}\rangle)$; Red: $\langle\tilde{\eta}\rangle$, green: $\langle\tilde{\eta}_+\rangle$ and blue: $\langle\tilde{\eta}_-\rangle$, for

(a) an ensemble of 100000 realizations with N=20 and r=32, generated by the simple forced method;

(b) an ensemble of 5000 realizations with N=20 and r=16, generated by random 3-SAT (open circles) and forced 3-SAT (lines).

Forced method

In what follows we refer to this method as the forced method. All our simulations are done with realizations generated by this method. First we need a solution $\vec{x}$, which is created as described in the previous section. Then we fill the first position in every clause of formula $\mathcal{F}$ as in Eqn. (3.2) and generate two histograms $h_{\text{pos}}(i)$ for variables without negations and $h_{\text{neg}}(i)$ for negated variables. Now the task is to choose $2M$ variables and fill the clause's second and third position so that the solution is not revealed. In the mean our target distribution should be $h_{\text{neg}}(i) = h_{\text{pos}}(i) = 3M/2N = 3r/2$ for all i, as in the random 3-SAT case. It is convenient to prepare a set K which holds variables so that we can take random elements out of it and fill the clauses. This automatically leads us to the target distribution. So for each variable i we calculate

$$n_{\text{pos}}(i) = \frac{3}{2}r - h_{\text{pos}}(i) \qquad \text{and} \qquad n_{\text{neg}}(i) = \frac{3}{2}r - h_{\text{neg}}(i)$$

and add

$$\underbrace{\{v_i, v_i, \ldots, v_i\}}_{n_{\text{pos}}(i)} \qquad \text{and} \qquad \underbrace{\{\overline{v_i}, \overline{v_i}, \ldots, \overline{v_i}\}}_{n_{\text{neg}}(i)}$$

to the set K. Now we randomly draw elements of K and fill the second and third position of the clauses. The negations are set so that the variables evaluate to FALSE under the assignment of $\vec{x}$, to maintain the highest probability that $\vec{x}$ is unique. However, we have no constraint that a variable is unique in each clause, so we have to get rid of variables that occur twice in one clause. At that point we use a Monte Carlo method with a variable exchange as the Monte Carlo move. Since we do not want to change the solution, we can only exchange the variable at the first position in one clause with the variable at the first position of another clause. The same is true for the variables at the second and third positions of the clauses. In the end the realization has to pass a filter as described in section 3 to eleminate solutions that are near our forced one.

The distribution of the coordination number η now looks more like the one for random realizations (Fig. 3.1b). The double peak structure in the distribution of $\langle \eta_+ \rangle$ and $\langle \eta_- \rangle$ has disappeared, and indeed one is no longer able to construct $\vec{x}$ from the histograms of the negations of variables. We want to mention that we can create realizations with a sharp coordination number distribution so that every variable appears in the same number of clauses. We have not studied these realizations intensively, but they seem to be much weaker than the ones with fluctuating coordination number. For $r < 3.5$ we are not able to construct USA realizations efficiently. For example, the probability to obtain a USA for $r = 3.5$ and $N = 40$ is $p_{\mathrm{USA}} \approx 0.005$ and for $N = 50$ $p_{\mathrm{USA}} \approx 0.001$ with our method. This is because there are only a few clauses which naturally lead to many states that satisfy every clause.

Chapter 4

Classical Simulations

We use the term "classical" to describe all simulations that are done without using any concept of quantum theory. In the present work this covers mainly simulations of the Monte Carlo type.

We first tried to apply the simulated annealing method [36] to the 3-SAT problem, but only a few instances could be solved with this method. Simulated annealing is not the right technique to get the right expectation value of physical properties like energy $\langle E \rangle$ or specific heat $\langle C \rangle$, for example. A very promising method for this purpose is an algorithm proposed by Wang and Landau [37], the Wang Landau Algorithm, which can be used to approximate the density of states (DOS). It is a very powerful tool, since using the DOS $g(E_i)$ we can calculate the partition function

$$Z = \sum_{\mathrm{q}} \mathrm{e}^{-\beta E_q} = \sum_i g(E_i)\ \mathrm{e}^{-\beta E_i}, \tag{4.1}$$

where q numbers the configurations and i the energy levels. The partition function Z enables us to compute many thermodynamic quantities, like the mean energy

$$\langle E \rangle = \frac{1}{Z} \sum_i E_i\, g(E_i)\ \mathrm{e}^{-\beta E_i}, \tag{4.2}$$

and the mean specific heat

$$\langle C\rangle = \frac{\partial\langle E\rangle}{\partial T} = \frac{\langle (E-\langle E\rangle)^2\rangle}{kT^2}$$
$$= \frac{1}{\beta^2}\left(\frac{1}{Z}\sum_i {E_i}^2\, g(E_i)\ \mathrm{e}^{-\beta E_i} - \left(\frac{1}{Z}\sum_i E_i\, g(E_i)\ \mathrm{e}^{-\beta E_i}\right)^2\right). \tag{4.3}$$

(4.4)

Throughout this work $\beta = 1/k_B T$ denotes the inverse temperature where the temperature T is measured in units of the Boltzmann constant k_B.

The main goal of this chapter is to obtain the DOS $g(E_i)$ for a large sample of instances (typically 4000) and to evaluate the thermodynamic quantities. Figure 4.1 demonstrates that using the DOS to compute the energy and the specific heat for the 3-SAT problem is much better than using the simulated annealing method. In the case of simulated annealing we started with a system at high temperature (T=30) and let the temperature decrease exponentially to zero. After each temperature decrease we performed 50 000 sweeps and 10 000 measurements with 20 sweeps between each measurement. After 670 simulation steps the system reached the temperature T=0.036 and the system was not yet in the ground state. This becomes perceivable by comparing the simulated annealing simulation results with the ones obtained using the Wang Landau method. From Fig. 4.1a it can be clearly seen that the energy value obtained with the simulated annealing method is much too high. As a consequence, the simulated annealing method is also unable to reproduce the second peak in the specific heat (see Fig. 4.1b).

This chapter is organized as follows: In section 4.1 we give a brief description of the Wang-Landau algorithm and we discuss various possibilities for a parallel implementation of it. In section 4.2 we present our simulation results for the 3-SAT problem. We first describe results for the energy and the specific heat. We then study the density of states and measure the computational complexity of a given realization and a class of realizations in terms of the runtime that is needed to find a solution. To learn more about

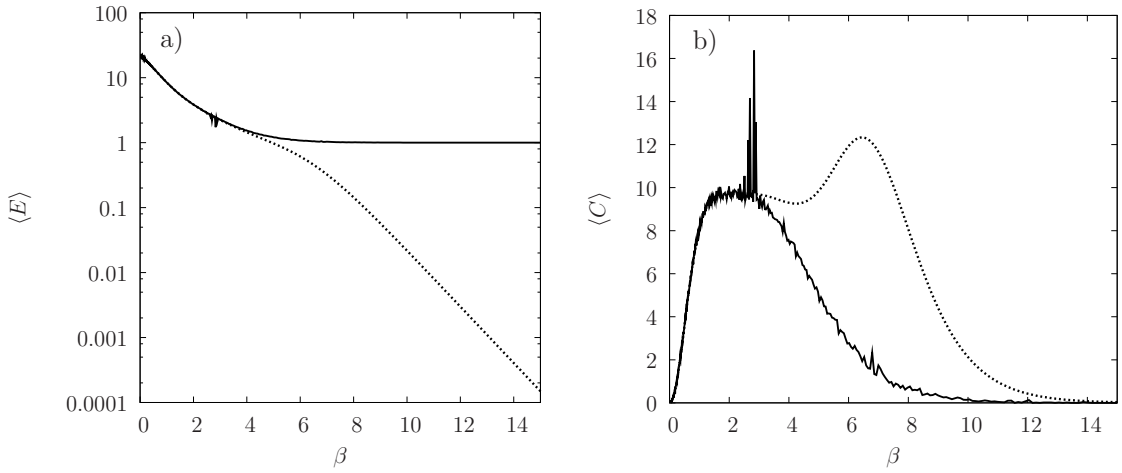

Figure 4.1: Qualitative comparison between the results for the energy (a) and the specific heat (b) as a function of the inverse temperature, obtained with the simulated annealing (continuous lines) and the Wang Landau method (dashed lines) for an unknown ground state of the 3-SAT problem with $N = 40$ and $r = 4.5$.

the runtime of the 3-SAT problem we calculate the microcanonical and the canonical distribution. From the canonical distribution we define a quantity which we call barrier and that can be used to get information about the runtime of the algorithm.

4.1 Classical algorithms

In this section we first give short descriptions of the algorithms that we use to obtain the thermodynamic quantities of the 3-SAT problem. We mainly utilize the Wang Landau algorithm [37], an algorithm first proposed in 2001 that can be used to approximate the DOS of a system, followed by a run of the multicanonical algorithm to obtain the DOS. Finally we focus on the possibilities to parallelize the Wang Landau algorithm.

4.1.1 The serial Wang-Landau algorithm

In general, the algorithm is suitable for all problems having discrete energy levels. The goal is to find a probability distribution so that any energy level occurs with equal probability when we perform random Monte Carlo moves (for example single spin flips). Some energy levels are realized by many states and some by only a few. The first ones appear much more often than the latter. Our sought-after distribution has to cancel out this effect. Using ratios of the *real* DOS $g\left(E_i\right)$, where i numbers the energy level, as transition probability between two energy levels numbered by k and l:

$$p\left(E_k \rightarrow E_l\right) = \min\left[\frac{g\left(E_k\right)}{g\left(E_l\right)}, 1\right], \tag{4.5}$$

will produce a random walk and a flat histogram in energy space, but a priori we do not know $g\left(E_i\right)$.

The algorithm works iteratively. We first approximate the DOS by $g(E_i) = 1$ for all energy levels E_i and set in the energy histogram $h(E_i) = 0$ for all energy levels E_i. We then define a multiplicative factor $f = e \approx 2.71$ so that every time the algorithm visits energy level E_i we modify $g(E_i)$ and $h(E_i)$ as follows

$$g(E_i) \rightarrow f g(E_i) \tag{4.6}$$

$$h(E_i) \rightarrow h(E_i) + 1. \tag{4.7}$$

$$\tag{4.8}$$

As a result, energy levels which occur with high probability become more and more unlikely. Since on every move we increment the energy histogram $h(E_i)$ with one, it simply counts the number of visits. When $h(E_i)$ becomes flat we have performed a random walk, and the approximation of $g(E_i)$ is at least as good as the modification factor f. We then take the actual $g(E_i)$ as approximation, replace f by $\sqrt{f}$, reset $h(E_i)$ to zero, and start the next iteration. This will subsequentially lead to a finer approximation of $g(E_i)$. The term *flat* means that the maximum of the histogram

$h(E_i)$ divided by its minimum must be smaller than $\max(h(E_i))/\min(h(E_i)) < 0.9$. This iterative algorithm is called the Wang Landau algorithm. We use the resulting approximation of the DOS $g(E_i)$ as input for a multicanonical simulation [38] which will give us the final estimation for the DOS. Typically we use Jackknife error analysis [39] with 20 Jackknife bins, to estimate the errors of $g(E_i)$ at this point. The first step using the Wang Landau algorithm is called approximation of the DOS and the second step using the multicanonical simulation is called the measurement of the DOS.

4.1.2 The parallel algorithm

The most obvious way to parallelize the Wang Landau algorithm is a domain decomposition of the energy interval in such a way that every processor has its own interval to work on. In the end the different pieces of the DOS have to be normalized and glued together in the right way, as described in [37]. But in case of the 3-SAT problem we only have a small energy interval. Therefore we want every processor to store the whole histogram and DOS. The idea of the Wang Landau algorithm is to use the information of newly discovered states immediately, which means after every Monte Carlo move. This can be done in a shared memory implementation, in which the DOS and the histogram are shared among all processors. In [40] an OpenMP version of this idea is described. We want our code to run on JUROPA, a clustercomputer of the Jülich Supercomputing Centre (JSC), using MPI, so we follow another path: Multiple Monte Carlo Walkers. In this scheme, every processor stores its own histogram and DOS approximation, and at fixed intervals in time they communicate and build one histogram and DOS which contains all the information. When the whole histogram is flat, the factor f is decreased and the next iteration is started. Further, we synchronize the energy interval so that every processor explores the same interval. There are three free parameters in this scheme. The first free parameter is the time schedule for the communication to take place. Since the communication among the processors must be a global one, it is not feasible to

broadcast the data every MC move. We use an interval of $t_{\text{interval}} = 100$ or 1000 sweeps between each message exchange process. The second free parameter is how to build the whole histogram from the histograms stored on each processor. This can be done by summing up all histograms of the p processors

$$h_{\text{mean}}(E_i) = \sum_j^p h_j(E_i), \tag{4.9}$$

or by taking into account only the histogram j that becomes flat first

$$h_{\text{race}}(E_i) = h_j(E_i), \tag{4.10}$$

which forms a race condition among the processors. For technical reasons we store the logarithm of the DOS, so we only have the value $\ln(g(E_i))$. The approximated, logarithmic DOS $\ln(g(E_i))$ at each step can be the simple mean of all $\ln(g_j(E_i))$

$$\ln(g_{\text{mean}}(E_i)) = \frac{1}{p}\sum_j^p \ln(g_j(E_i)), \tag{4.11}$$

or $\ln(g(E_i))$ can be the DOS of the processor on which the histogram j became flat first

$$\ln(g_{\text{race}}(E_i)) = \ln(g_j(E_i)). \tag{4.12}$$

Another possibility is to introduce a weight function $w_j(E_i)$ that measures how much information the processor j has gathered at each energy value E_i

$$\ln(g_{\text{weight}}(E_i)) = \frac{1}{p \cdot \sum_j^p w_j(E_i)} \sum_j^p w_j(E_i) \ln(g_j(E_i)). \tag{4.13}$$

We set $w_j(E_i) = h_j(E_i)$ so that the processor contributes more to the DOS at an energy value that was visited often. It is the latter method using $g_{\text{weight}}(E_i)$ that we employed to produce all our results presented in the next section. In Appendix A, the core of this method implemented in C and MPI can also be found. In what follows we compare as an example our method which uses $g_{\text{weight}}(E_i)$ as DOS and $h_{\text{mean}}(E_i)$ as histogram to a method which uses $h_{\text{race}}(E_i)$ and the DOS $g_{\text{race}}(E_i)$ from the processor that has as the

first a flat histogram. We also try a method which utilizes only the mean values of the histogram and DOS.

In Fig. 4.2a we present the results for the DOS as a function of the flatness of the histogram and the number of sweeps for three versions of the parallel Wang Landau algorithm simulated on 4 processors. We clearly see the convergence of the flatness of the histogram. The general structure of the quantity flatness per iteration is that it starts at zero and approaches 0.9 with the increasing number of MC moves. We want to remark that right after the start all three methods produce the same curve for the flatness (not shown) and that most of the simulation time is spent in the last iterations of the algorithm so that the plot is logarithmic in x-direction. For the curve of the DOS calculated by means of Eq. (4.9) and (4.11) (red dots) not all iterations are shown because the algorithm would approximately need 10^{12} sweeps to converge to the same DOS as obtained with the other methods. Therefore this method is not feasible. Taking the DOS from the processor which has as the first a flat histogram (green dots) is much better, but it seems to be an unstable method because some realizations take a very long simulation time. The method we used to produce all our results utilizes the weighted mean of the DOS and shows the fastest convergence for the 3-SAT problem (blue dots). This observation is consistent with Fig. 4.2b, which presents the speedup S for various versions of the parallel Wang landau algorithm, including one version that works with random realizations of the 3-SAT problem instead of forced ones. The version that calculates the DOS from Eq. (4.11) is not included because it needs much more computation time than the other versions. The speedup $S(p)$ is defined as

$$S(p) = \frac{T(p)}{T(1)}, \tag{4.14}$$

and $T(p)$ is the time needed on p processors. We simulate 100 realizations and calculate the mean value of the time needed to find the solution. The scaling of the method using random realizations compared to the method using forced realizations is worse, but the random realizations are easier to solve. We see that the method calculating the DOS

from Eq. (4.13), which is the method we use to produce all our further results, scales on average only up to 12 processors. For some realizations it scales well up to 64 processors, but for others it does not. In general the speed of the algorithm depends highly on the realizationsm, as the errorbars on the curve for the method using Eq. (4.13) and the difference in curves for the random and forced realizations show.

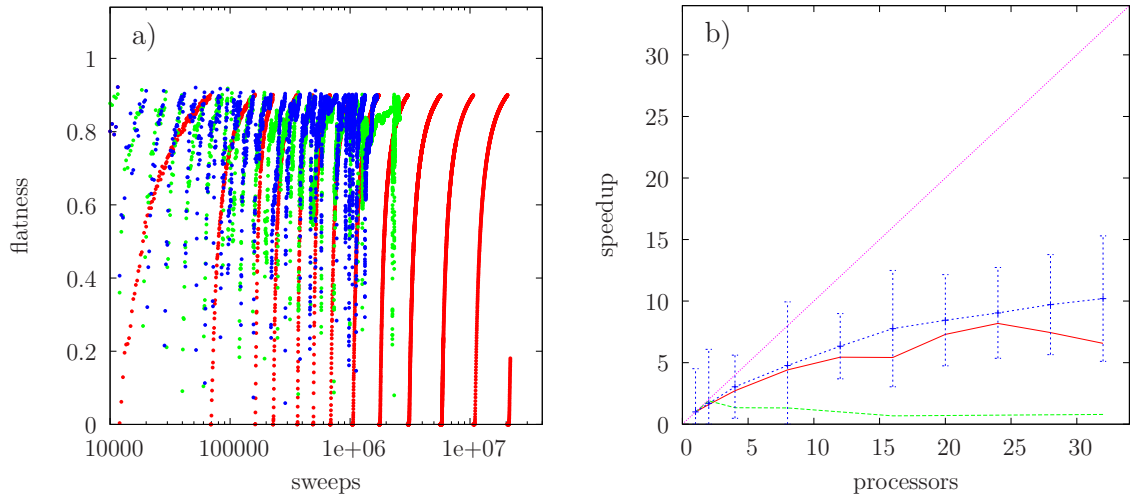

Figure 4.2: *a)* Comparison of three versions of the parallel Wang Landau algorithm for $N = 20$ and $r = 7.5$ simulated on 4 processors. The flatness of the histogram $h(E_i)$ is defined by the ratio $\max(h(E_i))/\min(h(E_i))$. The DOS is calculated from Eq. (4.11) (red dots), Eq. (4.12) (green dots), and Eq. (4.13) (blue dots).

b) Speedup, defined by Eq. (4.14), as a function of the number of processors for three different versions of the Wang Landau algorithm for a 3-SAT problem with $N = 20$ and $r = 7.5$. Red line: DOS calculated using Eq. (4.13) for 100 random realizations; green line: DOS calculated using Eq. (4.12) for 100 realizations obtained by the forced realization method described in chapter 3; blue line: same as green line but DOS calculated from Eq. (4.13), magenta line: Ideal scaling behavior.

4.2 Results

We only present results for realizations that are obtained by the forced method described in chapter 3. This means that the ground state is a forced one. We only use realizations with one unique solution, called unique satisfying assignment (USA).

4.2.1 Energy and specific heat

We have obtained a huge set of DOS data[1] for various ratios $r = M/N$ and system sizes N. We use Eq. (4.2) and (4.3) to calculate from the DOS the energy and the specific heat, respectively. During the measurement of the DOS we utilize Jackknife error analysis with 20 jackknife bins. Since we calculate $\langle E \rangle$ and $\langle C \rangle$ for every bin we can also obtain the errors for $\langle E \rangle$ and $\langle C \rangle$. The hardest realizations for random 3-SAT are found for $r = r_c \approx 4.2$ [41] near the SAT/UNSAT phase transition. One possible explanation for this higher complexity is the rise of a backbone [42], e.g. spins that are frozen in all solutions of the random 3-SAT solutions. As we restrict our analysis to USA realizations there is no backbone, and r values below 4.2 produce even harder realizations than larger r values. We start with large values of r, since for these realizations the DOS is easier to achieve.

Results for energy and specific heat as functions of the inverse temperature for one realization for $r = 7.5$ and $N = 80$ are shown in Fig. 4.3a. The energy drops exponentially to zero with decreasing temperature. This means that the temperature needs to be exponentially low in order to drive the system to the ground state (or to find the sought-after state). This can be understood by means of a a simple physical model:

[1] We used the Wang Landau algorithm and a broad histogram technique to estimate the DOS for $r \in [3.5, 4, \ldots, 8]$ and $N \in [10, 20, \ldots, 140]$.

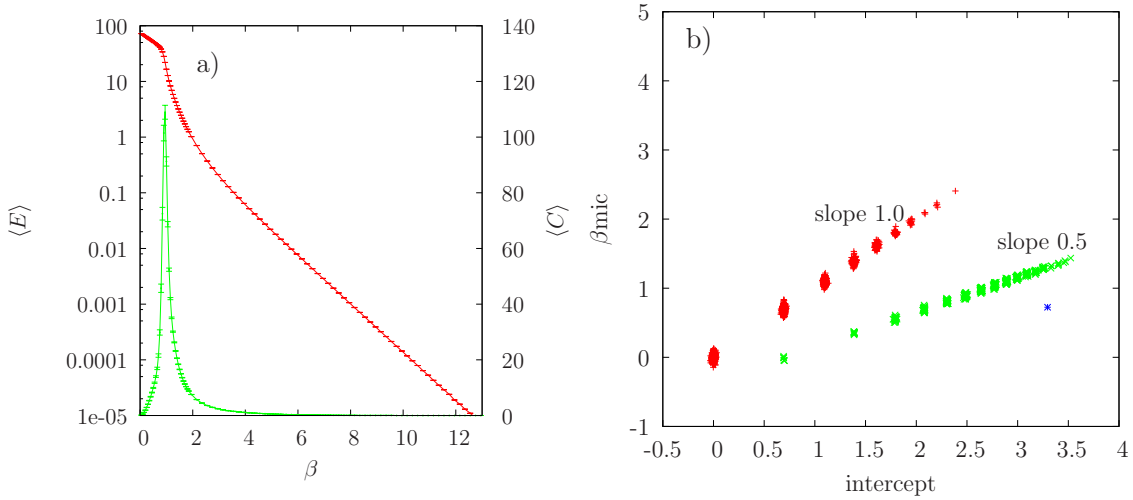

Figure 4.3: *a)* Results for energy (red line) and specific heat (green line) as functions of the inverse temperature for one realization of the 3-SAT problem for $r = 7.5$ and $N = 80$ spins. The energy is plotted in logarithmic scale so that one can see the exponential decay of $E(\beta)$. The energy makes a jump to a lower value at $\beta = 1$, corresponding to the position of the peak in the specific heat. *b)* Scatter plot of intercept b (see text) and β_{mic} for realizations with $E_1 = 1$ (red), $E_1 = 2$ (green) and $E_1 = 3$ (blue) of the 3-SAT problem with $r = 7.5$ and $N = 80$.

Assume that only two states with energy E_0 and E_1 contribute to the partition function

$$Z = g(E_0)\,\mathrm{e}^{-\beta E_0} + g(E_1)\,\mathrm{e}^{-\beta E_1}, \tag{4.15}$$

so that

$$\langle E \rangle = -\frac{\partial \ln(Z)}{\partial \beta} = \frac{E_0 g(E_0)\,\mathrm{e}^{-\beta E_0} + E_1 g(E_1)\,\mathrm{e}^{-\beta E_1}}{Z}. \tag{4.16}$$

We have

$$\begin{aligned} E_0 &= 0, \qquad & g(E_0) &= 1, \\ E_1 &= 1, \qquad & g(E_1) &= \mathrm{e}^{\beta_{\text{mic}}}, \end{aligned} \tag{4.17}$$

where the micro-canonical inverse temperature at energy E_0 is defined as

$$\beta_{\text{mic}} := \left(\frac{\partial S}{\partial E}\right)\Big|_{E_0} = \frac{\ln g(E_1) - \ln g(E_0)}{E_1 - E_0} \tag{4.18}$$

and the entropy $S = k_B \ln(g)$ is measured in units of k_B. So we get

$$\langle E \rangle = \frac{E_1 \frac{g(E_1)}{g(E_0)} \, \mathrm{e}^{-\beta E_1}}{1 + \frac{g(E_1)}{g(E_0)} \, \mathrm{e}^{-\beta E_1}}, \tag{4.19}$$

or

$$\ln\left(\langle E \rangle\right) = \ln\left(E_1\right) + \ln\left(\frac{g(E_1)}{g(E_0)}\right) - \beta E_1 - \ln\left(1 + \frac{g(E_1)}{g(E_0)} \, \mathrm{e}^{-\beta E_1}\right). \tag{4.20}$$

Using Eqn. (4.17), (4.18) and the fact that the last term of Eq. (4.20) vanishes if $\beta > \frac{\ln(g(E_1))}{E_1}$ results in

$$\ln\left(\langle E \rangle\right) = -E_1 \beta + \beta_{\text{mic}}. \tag{4.21}$$

Hence, the energy goes exponentially slowly to zero, with slope $-E_1$ and intercept β_{mic}. The fit of an exponential e^{ax+b} to the energy tail in Fig. 4.3a gives: $a = (-1.00 \pm 0.00)$ and $b = (1.10 \pm 0.00)$. From the DOS we obtain $g(E_0) = (0.98 \pm 0.00)$ and $g(E_1) = (2.94 \pm 0.01)$ and we calculate $\beta_{\text{mic}} = (1.10 \pm 0.00)$. Hence, the parameters b and β_{mic} fit nicely and since $E_1 = 1$ the parameter a also has a correct value.

This fitting procedure can be performed more systematically for all realizations. For the example with $N = 80$ and $r = 7.5$ we see a clear correlation between the intercept b and β_{mic} (Fig. 4.3 b). The granularity of the simulation data is due to large r value and due to the integer values of the DOS: The system is bounded by many constraints so that there are only a few states with energy E_1 [2]. For example, $g(E_1 = 1) = 2$ gives $\beta_{\text{mic}} \approx 0.7$ and $g(E_1 = 1) = 3$ gives $\beta_{\text{mic}} \approx 1.1$. If we shift r to smaller values the situation changes. We still have some realizations with a small number of states

[2] mean of 4000 realizations for $N = 80$ is $\overline{g(E_1)} = (3.07 \pm 2.56)$

at energy E_1, but there are more and more realizations with many states at E_1 (see Fig. 4.4a for the 3-SAT problem for $r = 5.5$ and $N = 50$). The histogram shows peaks which fade away into a continuum. If we look at r near or below $r = 4.2$ the granularity disappears completely (results not shown). However, for all values of r the correlation between the intercept and β_{mic} remains. Hence, at low temperatures we only need to consider the ground and first excited state, to reproduce $\langle E \rangle$. This calculation holds for all statistical systems with $E_0 = 0$ and a discrete energy spectrum. In fact, *low temperature* means $\beta > \max_i \left(\frac{\ln g(E_i)}{E_i} \right) > \beta_{\text{mic}}$ so that only the first and second terms contribute to the expectation values $\langle E \rangle$ and $\langle C \rangle$.

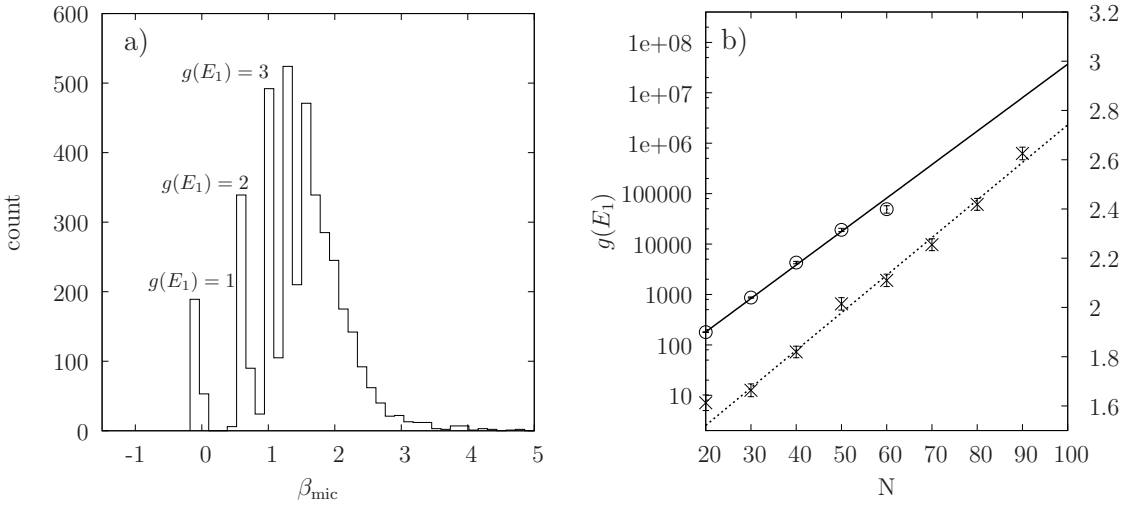

Figure 4.4: *a)* Histogram of β_{mic} for the 3-SAT problem for 4000 realizations with $N = 50$ and $r = 5.5$.

b) Number of first excited states $g(E_1)$ as a function of N for the 3-SAT problem for $r = 3.5$ (open circles) and 7.5 (crosses). The left y axis is used for the $r = 3.5$ case, the right one for $r = 7.5$. The lines are fits to the data (see text).

Before we discuss the specific heat we elaborate more on β_{mic}. In most cases, β_{mic} gives us the lowest temperature for which the system is not yet frozen. From Eq. (1.16) and Eq. (1.17) it follows that β_{mic} can be expressed as the logarithm of the number

of first excited states, $\beta_{\text{mic}} = \ln g(E_1)$. Thinking of an ferromagnetic Ising model, the corresponding quantity $\beta_{\text{mic, Ising}}$ would tell us something about the structure of the problem. For the Ising model with N spins we reach a first excited state by flipping a single spin in the ground state for which $g(E_0) = 2$, so we have $g_{\text{Ising}}(E_1) = 2N$. Using Eq. (1.17) we find $\mathrm{e}^{\beta_{\text{mic, Ising}}} = N$. Recently, interesting calculations for the Ising spin glass have been performed [43].

Fig. 4.4b shows $g(E_1) = e^{\beta_{\text{mic}}}$ as a function of N for the 3-SAT problem for $r = 3.5$ and $r = 7.5$. We have averaged $g(E_1)$ over all realizations. In the case of $r = 3.5$ (left y axis with logarithmic scale) we observe an exponential behavior, resulting in an exponential number of first excited states. From a fit we obtain $g_{E_1,3.5}(N) = b_S\,\mathrm{e}^{a_S N}$ with $a_S = (0.15 \pm 0.01)$ and $b_S = (8.50 \pm 0.86)$. Hence, in this case it is not a single spin flip that separates the ground state from the first excited states, but a combination of many spin flips. For $r = 7.5$ (right y axis without logarithmic scale) we observe a linear behavior, which is completely different from the exponential behavior for the smaller r case. From a fit we achieve $g_{E_1,7.5}(N) = c_S N + d_S$ with $c_S = (0.015 \pm 0.001)$ and $d_S = (1.22 \pm 0.09)$. Hence, the behavior of $g(E_1)$ of the USA 3-SAT problem at $r = 7.5$ is similar to the one of the Ising model $g_{\text{Ising}}(E_1)$. Therefore we can conclude that a flip of a single degree of freedom in the mean will lead from the ground state to a first excited state in the USA 3-SAT problem. The same holds for $r = 6.5$ and $r = 8$, but for $r = 4.5$ and $r = 5.5$ our data is not precise enough to draw a conclusion. For example for $r = 5.5$ it is not clear whether we see a rather weak exponential or a polynomial dependence of $g_{E_1,5.5}(N)$.

From Fig. 4.3 we have seen that for $r = 7.5$ (large r) the energy makes one jump and the specific heat has one peak. We call this peak the *first* peak. The transition can be understood as the change from an entropy driven system to a temperature driven one. The position of this transition is defined by the interplay of the DOS and the energy. At high temperatures, $\beta << \max_i \left(\frac{\ln g(E_i)}{E_i}\right)$, the expectation value of the energy is solely

defined by the DOS. At low temperatures we also have to deal with the factor $\mathrm{e}^{-\beta E}$. As a rough approximation for the position of the first peak, we take an energy value on the left from the high temperature region of the DOS and calculate

$$\beta = \frac{\ln\left(g(E)\right)}{E}. \tag{4.22}$$

For example, from Fig. 4.5 we get $\ln g(E) \approx E$, so that $\beta \approx 1$. This fits perfectly to our position of the first peak (see Fig. 4.3a). For small r we obtain larger $\beta \approx 2.5$, which also fits to the data (results not shown).

If we consider small r values (see Fig. 4.1), $\langle E \rangle$ changes significantly. In that case some of the harder realizations show a second peak in the specific heat $\langle C \rangle$ (Fig. 4.1b), corresponding to a second transition in the energy. At the first peak position of $\langle C \rangle$, the energy drops to $\langle E \rangle = 1$, then the energy remains approximately at this value before starting to approach to zero where the specific heat shows a second peak. This second peak appears at low temperatures, and therefore we use our simple model defined by Eq. (4.15) to calculate the specific heat. We find an approximation for $\langle C \rangle$ with an extremum at

$$\beta = \frac{\ln\left(\frac{g(E_1)}{g(E_0)}\right)}{E_1} \overset{E_1=1}{=} \beta_{\mathrm{mic}}. \tag{4.23}$$

For $r = 3.5, 4.5$ we see a clear correlation between the peak position and the micro-canonical inverse temperature (results not shown), so the model fits well for these values of r. For $r = 5.5$ and for intercepts smaller than 6 this correlation breaks down. To understand this, we have to take a closer look at the DOS, which we do in the next section. For even larger r we do not find enough realizations with a second peak in the specific heat to make an analysis and conclusion.

4.2.2 Density of states

What can we learn from the DOS, that is $g(E_i)$, directly? The physical interpretation of the DOS is that the DOS counts the number of states with energy $0, 1, 2, \ldots$. From a mathematical point of view, the DOS counts the number of possible assignments to a bit vector so that $0, 1, 2, \ldots$ clauses are violated. In the previous example we have considered a realization with $N = 80$ that is a realization with 2^{80} possible states. If we want to examine the whole DOS we need to do this on a logarithmic scale. Furthermore, the physical interpretation of $\ln g(E_i)$ is the microcanonic entropy, measured in units of k_B.

Using figures like Fig. 4.5 we find out that we can qualitatively distinguish between hard and easy realizations by looking at the DOS. For large r, hard realizations have only a few first excited states and easy ones have some more. Lowering r shifts the position of the kink in the DOS (arrow) to lower energy values. Thus, the qualitative shape of the DOS changes with r. At the smallest value of r that is considered in this work ($r = 3.5$), the kink disappears and the hard realizations are characterized by a *jump* of the DOS between E_0 and E_1 (Fig. 4.6). For example, the hardest realization of the class ($r = 3.5$, $N = 60$) has one ground state and 374630 first excited states. So for large r, hard realizations have only a few first excited states and easy ones have some more. But for small r hard realizations have exponentially many first excited states and easy ones have fewer. Now we can understand why the correlation between the intercept and the position of the second peak in the specific heat breaks down for increasing values of r. For intermediate $r = 5.5$ there are some realizations with a kink, some with a jump at E_0 and some with a jump at $E > 0$. We cannot cover the latter case with the simple model described by Eq. (4.15), since it only takes the first and second energy values into account. Furthermore, in such realizations the microcanonical inverse temperature β_{mic} is not relevant for the system, since it is also calculated from the first excited state

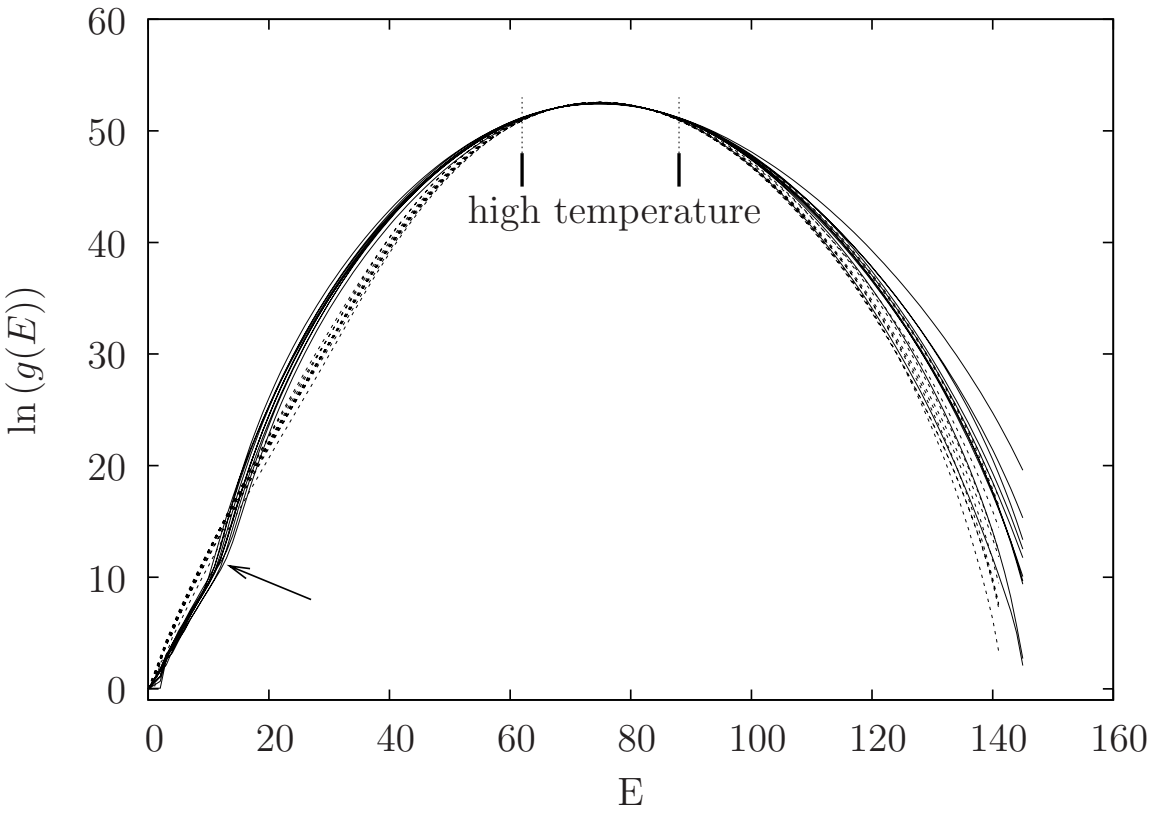

Figure 4.5: DOS of the 10 hardest (solid lines) and 10 weakest (dashed lines) realizations out of 4000 for the forced 3-SAT problem with $r = 7.5$ and $N = 80$. The arrow denotes the position of the kink in the DOS.

only. In general, the jump at E_0 is responsible for the second peak in $\langle C \rangle$.

The probability p to find the system at a specific energy E is

$$p(E) = \frac{1}{Z} g(E)\, \mathrm{e}^{-\beta E}\,. \tag{4.24}$$

At high temperatures $\beta \to 0$, $p(E)$ depends only on the DOS. So the system will with high probability be in a state corresponding to the maximum of $g(E)$. The system would also be in this region with a randomly chosen state, because there are exponentially many of these states. This fact is already known [44]. However, in [44] it is stated that the DOS can be approximated by a normal distribution for random 3-SAT problems. A normal

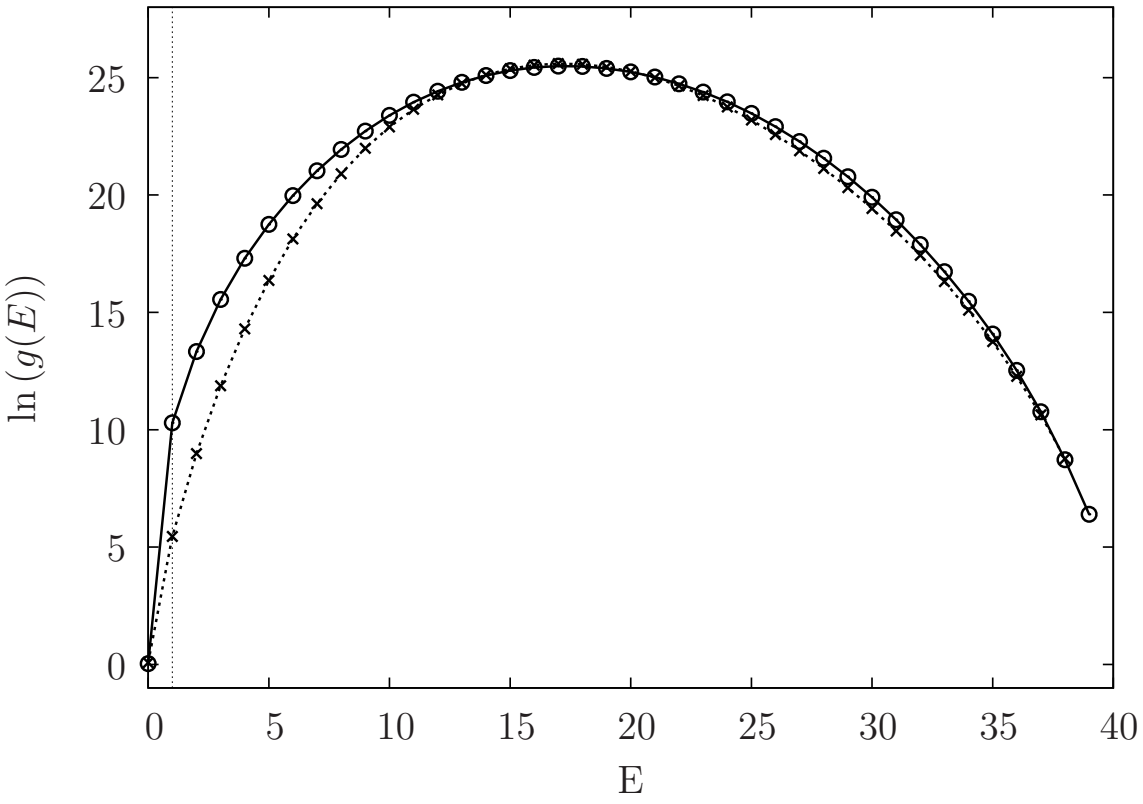

Figure 4.6: DOS of the hardest (open circles) and the weakest (crosses) realization out of 4000 for the forced 3-SAT problem with $r = 3.5$ and $N = 40$. The dashed black line denotes $E = 1$.

distribution in logarithmic space is represented by a parabola. Consequently all effects at low energies cancel out. Every method which finds the ground state by local spin flips must pass through these low energy states. For our realizations the approximation of the DOS by a normal distribution is only feasible for high temperatures. As an example the corresponding energies are marked in Fig. 4.5. It means that a system prepared at a *high temperature* ($T > 20$) will be in states with energies within the interval $[62; 88]$, and we can approximate the DOS by a normal distribution.

4.2.3 Computational complexity

We measure the computational complexity of a given realization, or a class of realizations, in terms of the runtime that is needed to find a solution. In practice we count the number of single spin flips that are needed by the walker to get from the energy maximum to the energy minimum, during the measurement of the DOS that is in the multicanonical simulation (see section 4.1.1). We call this number τ_α and we ensure that the walker visits the ground state at least 100 times ($\alpha = 1, \ldots, 100$) per realization, each time starting from a state with maximal energy (see Fig. 4.7a). Therefore τ_α is a measure for the time that our algorithm consumes, and from this we can determine the complexity of the algorithm. In general, the histogram of τ_α (Fig. 4.7b) shows an exponential tail for a large number of spin flips. This is because repeatedly making the "transition" from the maximum energy to the minimum energy is like solving the problem many times. Previously it has been shown that solving a 3-SAT problem many times, starting from different initial conditions, leads to an exponential distribution of the runtime [45]. The histogram of τ_α is not described by an single exponential function since there exists a minimum number of spin flips that we need to perform to drive the system from the energy maximum to the energy minimum. Because this minimum number of spin flips occurs rarely in the distribution of τ_α and also the exponential tail occurs rarely, we find an maximum in between. The fit of an extremal value distribution (Weibull distribution) to our data yields no stable results. We take the mean value of the histogram of τ_α and denote it by $\tau_{\text{mean}} = 1/L_\alpha \sum_\alpha^{L_\alpha} \tau_\alpha$. The convergence of τ_{mean} for one realization is shown in Fig. 4.7b. This value is not universal since it depends on the algorithm (e.g. the spin updates) that is used. Therefore it also shows no correlation with the runtime of the zChaff algorithm [24].

The number τ_{mean} is defined for a single realization, but normally we consider an ensemble of 4000 realizations for the same N and r. Moreover, the distribution of

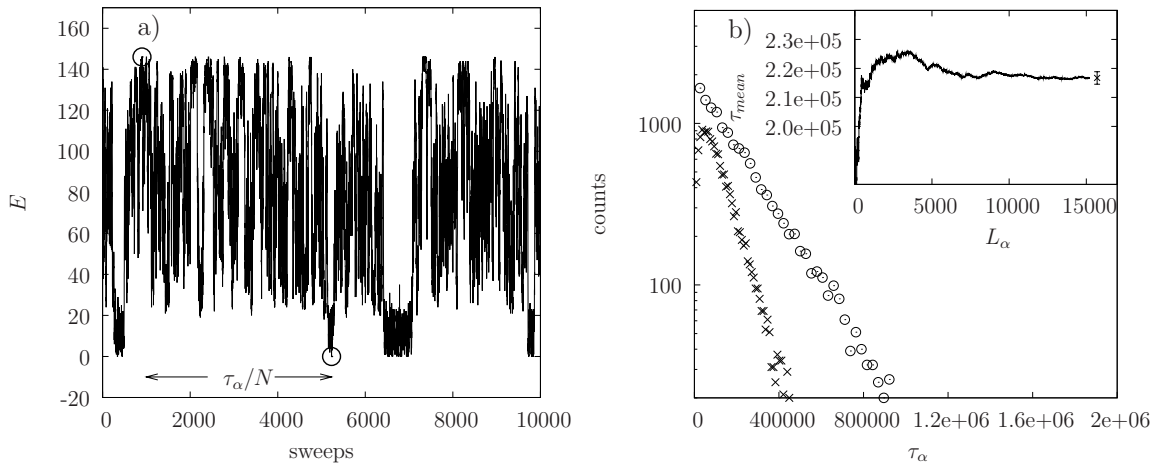

Figure 4.7: *a)* Transition profile of the Wang-Landau walker for the typical realization of an ensemble with $r = 7.5$ and $N = 80$. The arrows indicate the length of one transition time τ_α measured in the number of sweeps (N spin flips).
b) Histogram of the transition times τ_α for the typical realization with $r = 7.5$ and $N = 80$ on a logarithmic scale. Open circles: Transition time from the energy maximum to the energy minimum; crosses: Transition time from the energy minimum to the energy maximum.
Inlay: Convergence of the mean value $\tau_{\text{mean}} = 1/L_\alpha \sum_\alpha^{L_\alpha} \tau_\alpha$ with the number of transition events L_α.

τ_{mean} among an ensemble shows a clear exponential tail. Note that the Wang-Landau algorithm produces a random walk in energy space, and a random walk needs N^2 single moves to make the transition from the maximum energy to the minimum energy. So we have an intrinsic factor of N^2 in τ_{mean}. By defining $\tau = \tau_{\text{mean}}/N^2$ we compensate for this. Since we want to present the typical case complexity we take the median value τ_{median} of all τ of an ensemble with fixed r and N. The median has the advantage that it neglects the exponential rare realizations and that it exhibits a good convergence for our ensembles (not shown). By modifying N we obtain a curve that grows exponentially with N, as expected because of the NP-hardness of the 3-SAT problem. For different

$r \in [3.5; 8]$ we achieve different curves, as shown in Fig. 4.8a. The statistical error of the median is obtained by Jackknife error analysis with 10 bins. The convergence of $\tau_{median}(N)$ to the curve in Fig. 4.8a when we consider only parts of the statistic is from below (not shown), so at most we have too small median values as systematic error. For each value of r we can determine the complexity c_r (as defined in section 2.3) from a fit of the exponential function $f(N) = A_\tau \, \mathrm{e}^{c_r N}$ to our data. As an example, we have drawn this fits in Fig. 4.8a for the curves with fractional r-values. We also want to remark that Fig. 4.8a depicts results obtained with two different simulation methods, $r = 3.5$, 4.5, 5.5, 6.5 and 7.5 are obtained with a standard Metropolis update, and results for $r = 4$, 5, 6, 7 and 8 are obtained by a heatbath algorithm. Therefore the values of the amplitude A_τ for the different algorithms do not match. Nevertheless, the complexity is not altered using a different update algorithm. In Fig. 4.8b the complexities c_r are plotted as a function of r. Some datapoints are constructed with less statistic than others and therefore they have a larger errorbar. The curve is obtained with the constraint of USA realizations and it differs very much from the one without such a constraint (not shown). If we had pure random realizations, there would be a peak in $c(r)$ at $r \approx 4.2$, the point where the SAT/UNSAT phase transition takes place. At this point the probability that a given realization has a ground state with $E_0 = 0$, that is the realization is satisfiable, is $p_{\mathrm{SAT}} = 0.5$. For most of the algorithms that want to find a satisfying assignment this is the hardest region. If we slightly increase r the probability decreases, $p_{\mathrm{SAT}} < 0.5$, and if we decrease r, $p_{\mathrm{SAT}} > 0.5$ increases, until almost all realizations are solvable. Exactly at $p_{\mathrm{SAT}} = 0.5$ it is hard to solve the SAT problem. As already mentioned, the hardness in this region can be explained with the rise of a backbone [42]. However, in the present work we consider only USA realizations, so there is no peak in $c(r)$. If we lower r below the $r = 4.2$ the realizations become harder and harder, as shown in Fig. 4.8b.

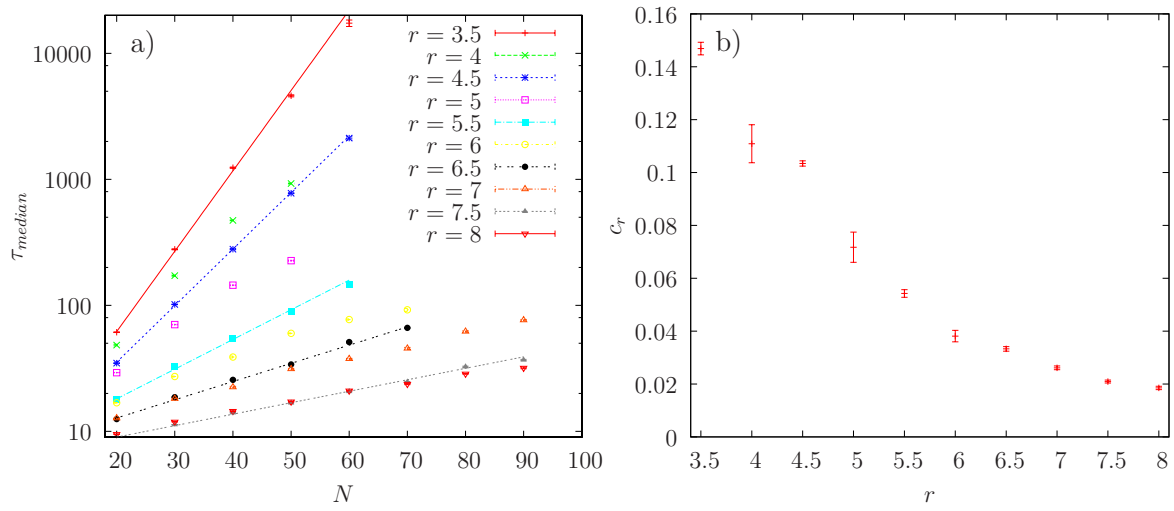

Figure 4.8: *a)* Scaling of τ_{median}, a measure for the computational complexity, as a function of N for various r. The lines are our fits with exponential functions (see text) to the curves with a fractional r value.

b) Complexity c_r as a function of r.

4.2.4 Distribution Functions

Microcanonical distribution

Since the goal is to learn more about the nature of the 3-SAT problem we look at the procedure of local search by tracking a quantity called ground-state overlap $o_{gs}(\vec{s})$ during the simulation:

$$o_{gs}(\vec{s}) = \frac{1}{N}\sum_i s_i s_i^{gs} \quad \in [-1, 1], \tag{4.25}$$

where gs refers to the ground state. The ground-state overlap observable is directly related to the Hamming distance of two states [3]. We construct a histogram of the

[3] Let $h(\vec{s}^{\text{gs}}, \vec{x})$ denote the Hamming distance between the ground state and the state x. The ground-state overlap is then given by

$$o_{gs}(\vec{x}) = \frac{h(\vec{s}^{\text{gs}}, \vec{x}) - N/2}{N/2}. \tag{4.26}$$

energy and the ground-state overlap observable during the Wang-Landau simulation and update it every Monte Carlo move. The integral of the histogram is normalized to one. This results in a probability $p(E, o_{gs})$ that a state with a given energy and overlap with the ground state appears in the simulation. Note that the simulation is designed to produce a flat histogram in the direction of energy, but not in the direction of the ground-state overlap. Therefore we do not have entries at every ground-state overlap value. We call $p(E, o_{gs})$ the microcanonical distribution since the energy at every entry is fixed and it can be written as

$$p(E, o_{gs}) \propto \frac{\sum_{\text{conf}} \delta\left(E - E_{\text{conf}}\right) \delta\left(o_{gs} - o_{\text{conf}}\right)}{N_{\text{micro}}}, \tag{4.27}$$

where "conf" refers to the spin configurations and where the normalization factor

$$N_{\text{micro}} = \sum_{\text{conf}} \delta\left(E - E_{\text{conf}}\right) = g(E), \tag{4.28}$$

takes into account that every energy value has equal probability. Another possibility would be to take

$$N_{\text{micro}} = \sum_{\text{conf}} 1, \tag{4.29}$$

meaning that every state appears with equal probability. We choose $N_{\text{micro}} = g(E)$ as normalization, because the Wang-Landau algorithm produces an approximation to this function.

The ground state $\vec{s}^{gs}$ appears at $p(0, 1)$ and has a high probability since it is unique for this particular energy. We have observed that easy realizations are characterized by a structure that allows the system to smoothly walk from the high temperature phase to the ground state. Hard realizations show a valley in the distribution that separates the ground state from the high temperature phase.

In Fig. 4.9 the microcanonical distribution for $r = 7.5$, $N = 80$ is plotted for an easy and a hard realization. The color blue marks areas of states that occur with a high

probability. The plot for the hard realization shows two separated blue areas whereas the plot for the easy realization shows one single blue domain. We now want to find out what is happening during the annealing process. At high temperatures the system starts at the maximum of the DOS, which is approximately located at the center of the energy interval (see for example Fig. 4.5). Since an arbitrary state has nothing in common with the ground state, the ground-state overlap is zero. Hence, we start in the middle of the 2D plot independently of the character of the realization. In case of the easy realization, the system can smoothly follow the blue domain to the ground state, characterized by $E = 0$ and $o_{gs} = 1$. However, for the hard realization the 2D plot shows a deep valley in the neighborhood of the ground state. Moreover, for lower energies the blue area bends away from the ground state to states with $o_{gs} = 0.5$. So the structure of the realization points away from the ground state, and in this example the algorithm needs to overcome a barrier of $\approx \mathrm{e}^{10}$ in the probability distribution to reach states which are near the ground state.

If we lower r the distribution changes. The separation of the ground state and the bulk of the most probable states in the energy direction get closer until for $r = 3.5$ the separation is $\Delta E = 1$. Moreover the valley, which is along the path to the ground state, becomes narrower. Furthermore, the ground state becomes isolated from the other states, which conforms to the disappearance of the kink in favor of a jump in the DOS. We do not show the plots because they are not very illustrative, since the details are concentrated only between two energy values. All the above mentioned facts lead to an increasing computational effort for the system to find the ground state: There are fewer states near the ground state, since the valley becomes narrow and isolated. In addition the system is driven along the wrong path over a wider range in energy, up to $E = 1$. Normally, a low temperature leads to such a low energy, but then the system is trapped by a probability-gap in the ground-state overlap direction and by temperature in the energy direction. For $r = 7.5$ the system ends up in the wrong path at a higher energy

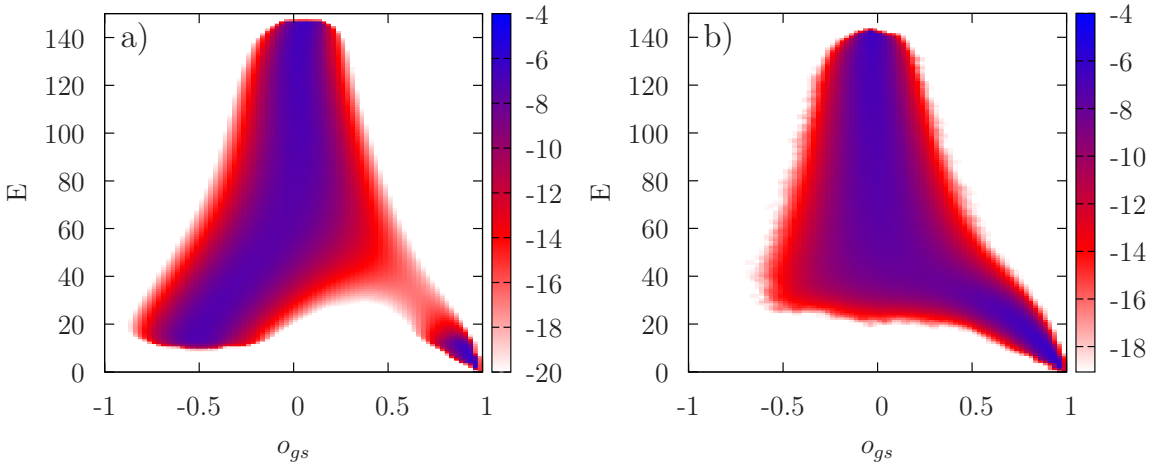

Figure 4.9: Logarithm of the microcanonical distribution $\ln p(E, o_{gs})$ for $r = 7.5$ and $N = 80$.

a) Hardest realization

b) Weakest realization

than $E = 1$ and has yet more possibilities left, so it is easier to propagate backwards.

Canonical distribution

The microcanonical distribution tells us something about the inner structure of the problem, but we want to perform an active search, which physically can be an annealing process of the system. We also want to understand how the valley and the move to low-energy states separated by the valley from the ground state affect such an annealing. With the aid of the microcanonical distribution $p(E, o_{gs})$ the canonical one $p(T, o_{gs})$ can be calculated. The fixed energy needs to be interchanged by fixed temperature with the introduction of a Boltzmann factor:

$$p(T, o_{gs}) = \frac{1}{Z} \sum_{\text{conf}} p(E_{\text{conf}}, o_{gs}) \, \mathrm{e}^{-\beta E_{\text{conf}}} = \sum_i g(E_i) p(E_i, o_{gs,i}) \, \mathrm{e}^{-\beta E_i} \, . \tag{4.30}$$

The normalization is chosen so that every temperature appears with equal probability. We obtain a distribution that gives us hints about the peak positions in the specific heat function. A nice example is depicted in Fig. 4.10a. It shows a hard realization for $r = 4.5$ with two peaks in the specific heat. The first peak corresponds to the transition from the unordered phase, which is characterized by $o_{gs} \approx 0$, to the phase with $E = 1$ states, which have $o_{gs} \approx -0.5$ as we know from the previous section. The second peak denotes the transition to the $E = 0$ frozen system.

In general, for a different realization the plot will look different. There can be one or two peaks in C and a shift in the ground-state overlap or not. The canonical distribution also shows a rich structure for large β. There are islands of nearby ground-state overlap values with jumps in between for some realization dependent ground-state overlaps, which take into account the inner structure of the particular realization (Fig. 4.10b). We do not consider these facts in detail, it may be a topic for future research.

Barrier

Let β_{CMAX} denote the inverse temperature of the peak in the specific heat function, where the energy drops to zero. Fig. 4.10b shows a bimodal distribution of the ground-state overlap at the fixed inverse temperature β_{CMAX}. The system has either $o_{gs} \approx -0.3$ or $o_{gs} = 1$. In between there is a valley in the probability distribution $p(\beta_{\mathrm{CMAX}}, o_{gs})$. To determine the depth of this valley, we can simply use the microcanonical distribution $p(E, o_{gs})$ and calculate the canonical distribution $p(\beta_{\mathrm{CMAX}}, o_{gs})$ from Eq. (4.30). However, this leads to large errors and in case of a deep valley we are not even able to perform the calculations. Naturally this is because we sample the canonical distribution $p(E, o_{gs})$ in the energy direction so that we do not have entries for every ground-state overlap value. To overcome this lack of information, we perform a Wang-Landau simulation in the ground-state overlap direction, to force the system to occupy states right

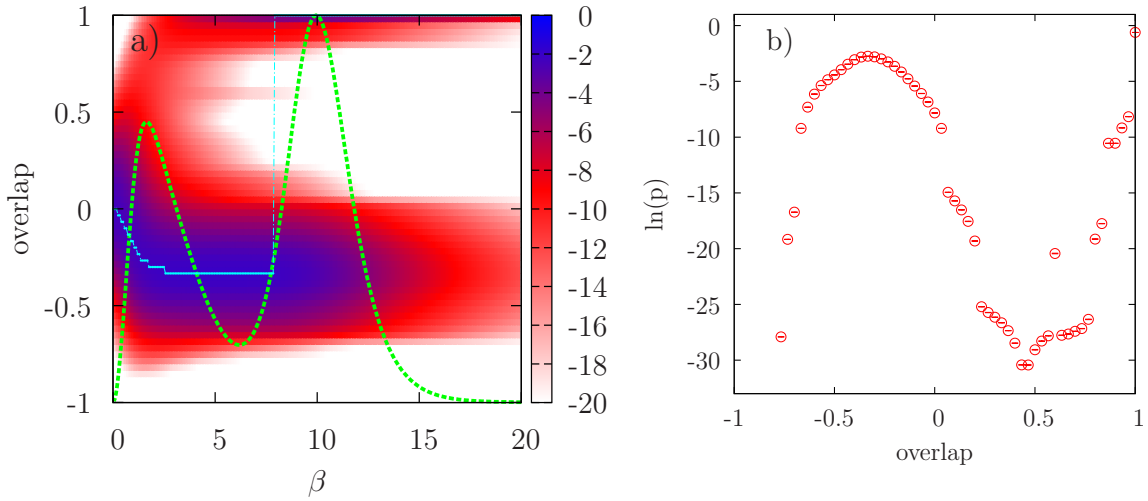

Figure 4.10: *a)* Logarithm of the canonical distribution $p(T, o_{gs})$ for a hard realization for $N = 60$ and $r = 4.5$. The specific heat as a function of β (green curve) has its own y-axis but shares the β-axis. The cyan curve shows the most probable states as a function of β.

b) Logarithm of the canonical distribution at β_{CMAX} for the same realization.

in the valley. We then achieve the final canonical distribution by a multimagnetic simulation [46]. In Fig. 4.10b the result of one of these additional simulations is shown. The canonical distribution is measured for the very high inverse temperature $\beta = 9.95$ at a realization with $r = 4.5$, the regime where most of the realizations show jumps in the distribution $p(\beta_{\mathrm{CMAX}}, o_{gs})$. If we consider $r > 5$ these jumps disappear and $p(\beta_{\mathrm{CMAX}}, o_{gs})$ becomes smooth. The depth of the valley acts like a barrier B_0 that the system has to overcome to find the ground state. Defining $P_{\mathrm{max,\,left}}$, P_{min} and $P_{\mathrm{max,\,right}}$ as the extremal values of $p(\beta_{\mathrm{CMAX}}, o_{gs})$ sorted by increasing ground-state overlap, we can express the barrier as

$$B_0 = \ln\left(\frac{P_{\mathrm{max,\,left}}}{P_{\mathrm{min}}}\right). \tag{4.31}$$

We find a clear correlation between the barrier B_0 and the quantity τ for each realization at $r > 4.5$. An example is plotted in Fig. 4.11a where every red cross denotes one realization. We see a nice exponential increase of τ with B_0. In general, we take 95% of all 4000 realizations for each N and r and perform a binning for each τ- and B_0-value (blue circles in Fig. 4.11a), followed by a fit of the function $f(x) = a' + b' \exp(c'x)$ to the data. The value a' takes into account that τ cannot be zero, but that the barrier can. For very easy realizations the barrier is not well defined, because there is no valley and the canonical distribution at $\beta = \beta_{\mathrm{CMAX}}$ is a monotonic function. Therefore we use a filter to get rid of such realizations. There are certainly regular contributions to the barrier B_0 but hopefully they will be small at values $B_0 >> 1$. For $r \leq 4.5$ the simple correlation breaks down. It seems that there are two different kinds of realizations, as Fig. 4.11b suggests, and for each of them the correlation between B_0 and τ holds. However, we only have a few realizations and only for N-values up to $N = 60$ for $r \leq 4.5$, and this is not enough to determine the two classes.

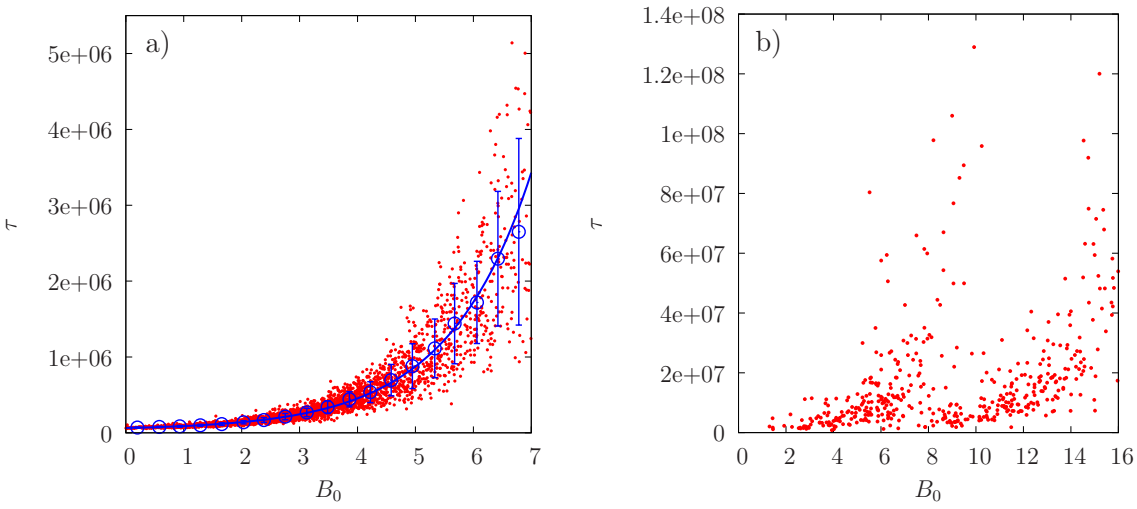

Figure 4.11: Correlation between the barrier B_0 and τ for $N = 80$ and $r = 7.5$ (a) and $N = 50$ and $r = 3.5$ (b). The blue circles denote the bins and the green line denotes a fit (see text).

The barrier B_0 is a static property of the system and thus independent of the algorithm that we use. It nevertheless describes the runtime of the Wang-Landau algorithm. This is a huge advantage over using τ that we used earlier to determine the complexity, since τ is algorithm specific. It would be interesting to see whether the runtime of other local search algorithms could also be described in terms of the barrier. Since the zChaff algorithm [24] is not correlated to τ, it is not correlated to the barrier B_0 as well, but zChaff is not a local search algorithm. Hence, one would have to study other local search algorithms, which is out of the scope of the present work.

Chapter 5

Quantum Simulations

As described in the general introduction, the quantum adiabatic (QA) Hamiltonian

$$H(t) = (1 - \lambda(t))H_D + \lambda(t)H_P, \tag{5.1}$$

needs to be slowly varying so that the system stays in its instantaneous ground state. We consider $\lambda(t) = t/T$ where T controls the rate at which $H(\lambda(t))$ varies and is the runtime of the algorithm. The quantum system evolves according to the time-dependent Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial}{\partial t}|\psi(t)\rangle = H(\lambda(t))|\psi(t)\rangle, \tag{5.2}$$

and we denote the instantaneous eigenstates of this system as

$$H(\lambda(t))\,|l,\lambda\rangle = e_l(\lambda(t))\,|l,\lambda\rangle. \tag{5.3}$$

Using this notation, $|0,0\rangle$ is the initial state and $|0,1\rangle$ is the final state that solves the optimization problem. The adiabatic theorem [11] states that when the energy gap is greater than zero, $|e_0(\lambda) - e_1(\lambda)| > 0$ for $0 < \lambda < 1$, we are sure that at the end of the algorithm at time $t = T$, we will measure the searched state

$$\lim_{T\to\infty} \langle 0,1|\psi(T)\rangle = 1. \tag{5.4}$$

The minimum mass gap is defined as

$$m_{GAP}(\lambda^*) = \min_{0<\lambda<1} \left(e_1(\lambda) - e_0(\lambda)\right), \tag{5.5}$$

where λ^* denotes the position of the minimum. With

$$T >> \frac{\max\limits_{0<\lambda<1} \left(\langle 1,\lambda|\frac{dH(\lambda)}{d\lambda}|0,\lambda\rangle\right)}{m^2_{GAP}(\lambda^*)}, \tag{5.6}$$

we can make the probability of Eq. (5.4) arbitrarily high [47]. The expression in the numerator of Eq. (5.6) is of the order of an eigenvalue of $H(\lambda)$, so only the minimum mass gap controls the runtime of the QA algorithm, $T \propto m^{-2}_{GAP}(\lambda^*)$. We have to remark that recently a discussion about the adiabatic theorem was started with an article that describes inconsistencies of the theorem [48, 49]. Later, the validity of the adiabatic theorem for special assumptions was proven [50], and a reformulation of the theorem was given [51].

To strengthen the mass gap-runtime relation we can also employ the Landau-Zener formula [52,53] to calculate the probability of a non-adiabatic transition, when we make several assumptions. First of all we suppose that λ^*, which denotes the position of the minimum mass gap m_{GAP} in the interval $0 < \lambda^* < 1$, is well defined and unique. Second, we assume that only the ground state and the first excited state contribute to the transition so that we can approximate the quantum system by an effective two-level system described by the Hamiltonian

$$H_{LZ}(t) = \begin{pmatrix} \epsilon_0(t) & \epsilon_{01} \\ \epsilon_{01} & \epsilon_1(t) \end{pmatrix}. \tag{5.7}$$

Here, $\epsilon_0(t)$ denotes the energy of the ground state and $\epsilon_1(t)$ the energy of the first excited state and both are time dependent. The transition region needs to be so small that we can treat $\epsilon_1(t) - \epsilon_0(t) = \alpha t + const$ and $\epsilon_{01} = const$. We solve H_{LZ} for its eigenvalues E_0, E_1 and we obtain an avoided crossing

$$E_1(t) - E_0(t) = \sqrt{\left(\epsilon_1(t) - \epsilon_0(t)\right)^2 + 4\epsilon^2_{01}}, \tag{5.8}$$

which means that at some $t = t^*$ we find $\epsilon_1(t^*) - \epsilon_0(t^*) = 0$ and the mass gap equals $m_{GAP}(t^*) = E_1(t^*) - E_0(t^*) = 2\epsilon_{01}$. The Landau-Zener formula gives us the probability for a diabatic transition in this quantum system

$$P_{LZ} = \mathrm{e}^{-2\pi\epsilon_{01}/|\frac{d}{dt}(\epsilon_1-\epsilon_0)|}. \tag{5.9}$$

Substituting $m_{GAP}(\lambda^*) = 2\epsilon_{01}$ and $\frac{d}{dt} = \frac{d\lambda}{dt}\frac{d}{d\lambda} = \frac{1}{T}\frac{d}{d\lambda}$ in Eq. (5.9) gives

$$P_{LZ} = \mathrm{e}^{-\pi T m^2_{GAP}(\lambda^*)/2|\frac{d}{d\lambda}(\epsilon_1-\epsilon_0)|}. \tag{5.10}$$

Keeping the system in the instantaneous ground state requires $P_{LZ} \to 0$ and therefore

$$T >> \frac{2\hbar D}{\pi m^2_{GAP}(\lambda^*)}, \tag{5.11}$$

with $D = |\frac{d}{d\lambda}(\epsilon_1 - \epsilon_0)|$ denoting the slope difference. We conclude that the runtime T of the QA algorithm in general is determined by the inverse mass gap squared

$$T \propto m^{-2}_{GAP}(\lambda^*). \tag{5.12}$$

It has been shown that one can obtain a polynomial speedup of the runtime with a local QA algorithm applied to the quantum search problem [54]. The idea is that the velocity of the QA control parameter, $\dot{\lambda}(t)$, becomes small in the vicinity of the quantum phase transition so the algorithm spends most of the time at the computationally hard region and only little time at the beginning and end of the algorithm. With this scheme one achieves a runtime $T \propto m^{-1}_{GAP}(\lambda^*)$. It has been shown that the global scheme, with $\dot{\lambda}(t) = const$, is robust against decoherence [55] and that the local scheme is very sensitive to it. Moreover, the computation time in the local scheme should be smaller than the decoherence time [56], just as in the gate model without error correction. Furthermore, for the local scheme we need information about the mass gap function before the algorithm starts, which means it is not of practical use. In the present work we want to find out whether the mass gap scales polynomially or exponentially

with the number of spins N. Therefore, we decide to analyze the scaling behavior of $m_{GAP}^{-1}(\lambda^*) = k(\lambda^*)$, which is the correlation length of the system in case of the Monte Carlo (MC) simulation. In [10] a polynomial scaling of the QA algorithm up to a small number of spins is observed. In [57] the theoretical claim is made that a large number of ground states would lead to an exponentially small mass gap, and in [58] this is verified numerically for systems with a number of spins up to $N = 20$ for $r = 3$. We are interested in much larger systems and in the relation between the classical 3-SAT problem and the quantized version. Therefore we mainly study less hard 3-SAT problems for $r = 7.5$ and 8.

5.1 Quantum algorithms

In general, we use three different methods to get information about the quantized version of the 3-SAT problem. First of all we use Mathematica for numerically exact[1] diagonalization of the QA Hamiltonian for different values of the QA control parameter λ. We obtain the full energy spectrum, for systems limited up to $N = 10$ spins. Another method we use is the Lanczos algorithm [59], which does not give us the full spectrum, but the four lowest eigenvalues only. The program is available in the quantum information processing group at the Jülich Supercomputing Centre as an OpenMP implementation. Using the Lanczos method we simulate systems with up to $N = 20$ spins. We spend most of the computer time on the third method: Quantum Monte Carlo (QMC), using the Suzuki-Trotter approximation [60] to extend the problem by a Trotter-time direction. The disadvantage of QMC is that we only obtain the mass gap $m_{GAP}(\lambda)$ and not a spectrum of eigenvalues. On the other hand, the huge benefit of using QMC is that we are not limited by the size of the Hilbert space, but by the cor-

[1] In this context numerically exact means that we can in principle calculate the results as exact as the machine precission.

relation length $k(\lambda)$ of the system. Hence, we are able to simulate much larger systems compared to what we can simulate with the other algorithms. All three methods give us at least the mass gap and thus the inverse correlation length $k(\lambda)$, $m_{GAP}(\lambda) = k^{-1}(\lambda)$. In what follows we only consider the QMC method.

Quantum Monte Carlo method

We quantize the theory by means of the partition function

$$Z_{QA} = Tr\left[\mathrm{e}^{-\beta\{(1-\lambda)H_D+\lambda H_P\}}\right], \tag{5.13}$$

with the QA Hamiltonian being composed of the problem Hamiltonian H_P and the driver Hamiltonian H_D. Note that for $\lambda = 1$ Z_{QA} describes the classical 3-SAT problem and for $\beta \to \infty$ the problem is solved. We use Trotter-Suzuki's formula [60] and introduce a regular temporal lattice with N_τ time slices, a finite step-size $\Delta\tau$ and periodic boundary conditions in Trotter time. The finite inverse temperature of this system is then given by $\beta = N_\tau \Delta\tau$ and the Boltzmann factor of the quantized theory reads

$$P_B = -\kappa_0 \sum_{\tau=1}^{N_\tau} H_P(s_{1,\tau}, \ldots, s_{N,\tau}) - \kappa_\tau \sum_{i}^{N} \sum_{\tau=1}^{N_\tau} s_{i,\tau} s_{i,\tau+1}, \tag{5.14}$$

where we assume a Wick rotation to imaginary time,

$$\kappa_0 = \lambda \Delta\tau \tag{5.15}$$

$$\kappa_\tau = -\frac{1}{2} \ln\left[\tanh\left((1-\lambda)\,\Delta\tau\right)\right], \tag{5.16}$$

$s_{i,\tau}$ denotes the spin degrees of freedom described in the σ^z-basis, $i = 1, \ldots, N$ and $\tau = 1, \ldots, N_\tau$. In our simulation we use $N_\tau = 128$, 256 and $\Delta\tau = 1$. Hence, we simulate at an inverse temperature $\beta = 128$ or $\beta = 256$, which is much higher than the highest microcanonical inverse temperature that we observe for our 3-SAT realizations[2] and it

[2]The highest microcanonical inverse temperature that we observe is $\beta_{\mathrm{mic}} \approx 12.8$ for systems with $r = 3.5$ and $N = 60$.

should be high enough to drive the system to its ground state in case of $\lambda = 1$. To obtain the mass gap, we calculate the correlation function of a τ dependent operator $A(\tau)$

$$\Gamma(\tau) = \frac{1}{N\,N_\tau} \sum_{i}^{N} \sum_{\tau_0=1}^{N_\tau} \langle A(\tau+\tau_0) A(\tau_0) \rangle, \tag{5.17}$$

where $\langle . \rangle$ denotes the expectation value over different samples. The correlation function $\Gamma(\tau)$ will be a sum of exponentials [61], which for large τ is dominated by the term corresponding to the first excited state. With the periodic continuation of the exponential decay to the N_τ lattice, we obtain

$$\Gamma(\tau) = A_\Gamma \left[\mathrm{e}^{-m_{GAP}\,\tau} + \mathrm{e}^{-m_{GAP}\,(N_\tau - \tau)} \right] + q. \tag{5.18}$$

The constant q can be calculated from

$$q = \frac{1}{N} \sum_{i}^{N} \left(\frac{1}{N_\tau} \sum_{\tau=1}^{N_{\tau_0}} \langle A(\tau_0) \rangle \right). \tag{5.19}$$

We tried several Trotter time τ dependent operators $A(\tau)$, like spin-spin correlation functions and spin-spin correlation functions per time slice, but they do not yield stable results. Incorporating the ground-state overlap $o_{gs}(\vec{s})$ as defined in Eq. (4.25), we set $A(\tau) = o_{gs}(\vec{s}(\tau))$ where $\vec{s}(\tau)$ denotes the vector of spins in time slice τ. This gives us a good observable to calculate the correlation function Eq. (5.17) and to obtain the mass gap from Eq. (5.18). When the gap m_{GAP} becomes small, N_τ needs to become large, so that the correlation function $\Gamma(\tau)$ will fit onto the lattice. This factor is limiting the use of the QMC method. For example, for the $N_\tau = 128$ lattice we can reliably detect correlation lengths up to $k = m_{GAP}^{-1} = 64$ (not 128 because of the periodic boundary condition in the Trotter direction).

We utilize Parallel Tempering (PT) [62, 63], because we observe much smaller errors and smoother mass gap curves with PT than without it. A similar observation is made in [13] for the exact cover model. We also tried a method similar to simulated annealing

in which, instead of cooling the temperature, we tune the parameter λ, but this method yields larger errors. We use $p = 64$ processors and therefore 64 PT partitions in the λ direction. For every realization we first do a simulation with a coarse λ resolution to determine λ^*, the position of the quantum phase transition, followed by a finer partitioning around λ^* to obtain $k(\lambda) = m_{GAP}^{-1}(\lambda)$. We note that in general every time we have done enough PT updates, so that the whole system undergoes a transition in the λ direction from values larger than λ^* to values smaller than λ^*, the expectation values become smooth and the errors become small. To detect whether there are enough PT steps or not, we analyse the assignment of λ-values to the processors. A processor should perform the transition from high $\lambda > \lambda^*$ to low $\lambda < \lambda^*$ at least 5 times, in order to produce good expectation values. Fig. 5.1 shows the assignment of λ-values of 2 processors for each MC sweep, one starting at low and the other at high λ-values. The realization is simulated with $N_\tau = 128$ and shows a correlation length of $k(\lambda^*) \approx 95$. We see that the system undergoes the transition from high to low λ only for the processor that starts at high λ. The other processor remains in the low $\lambda < \lambda^*$ region. We have to perform more sweeps to obtain small errors for the expectation values that we are interested in. Even though the correlation length is larger than the dimension of our lattice, it is a good approximation to the real value. It is an open question and could be a topic of future research why PT yields so many advantages for the simulation of a QA algorithm in the case of the 3-SAT (and Exact Cover) problem.

Comparison of the exact diagonalization, the Lanczos and the quantum Monte Carlo method

For systems with a small number of spins we are able to use all three methods in order to compare them. Fig. 5.2a shows the curve $k(\lambda)$ for a 3-SAT problem with $N = 6$ spins and $r = 4$. The magenta curve depicts the result of the exact diagonalization of $H(\lambda)$. This curve serves as our reference curve. The blue open circles represent the

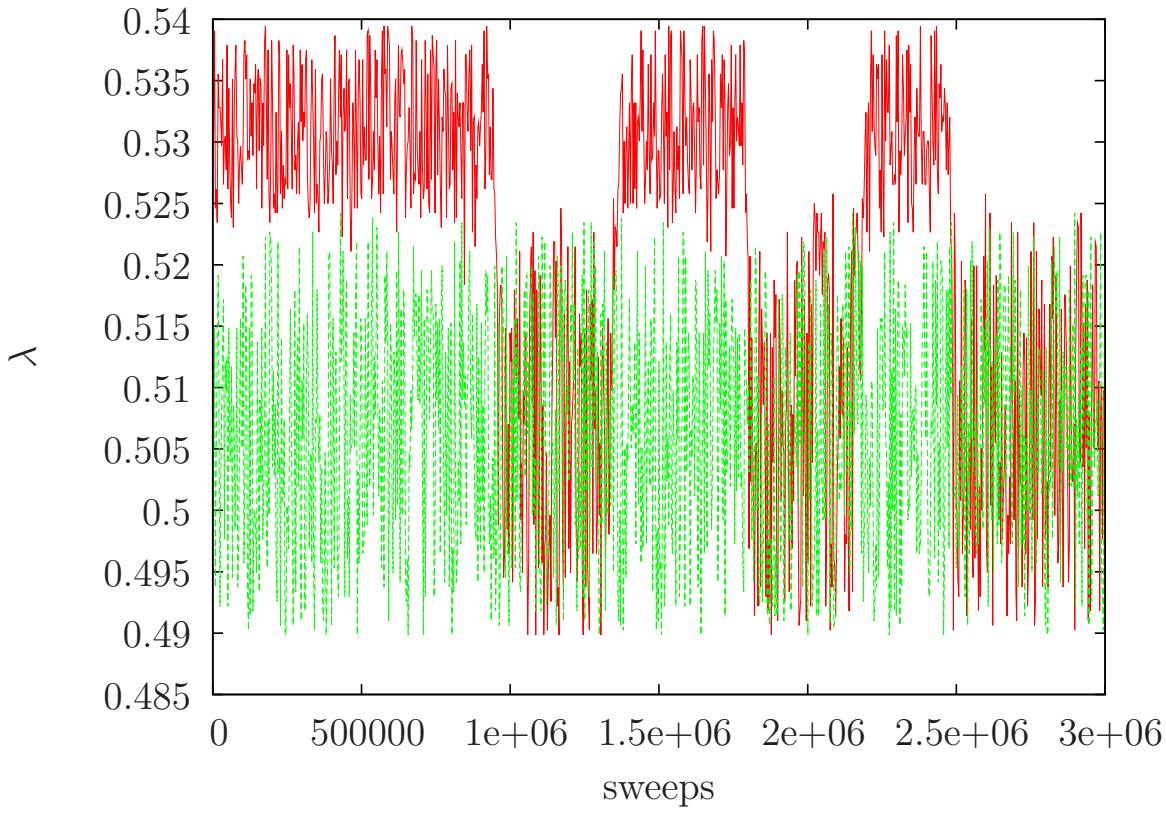

Figure 5.1: Parallel Tempering transitions from high $\lambda > \lambda^*$ to low $\lambda < \lambda^*$ for 2 processors (green, red) for a particular hard realization with $N = 20$ and $r = 7.5$.

results obtained with the Lanczos algorithm running on $p = 32$ processors and lay very nicely on the reference curve. The red dots represent the results obtained with a QMC PT simulation using $\Delta\tau = 1$. As we see it is not appropriate to reproduce the reference curve. This is because the correlation length $k^* = 6$ is of the same order of magnitude as $\Delta\tau = 1$, and then we get an error due to the finite lattice spacing. This error disappears if we repeat the simulation with a smaller lattice spacing $\Delta\tau = 0.25$, as the results represented by the green dots show. Hence, all three methods produce the same curve for the correlation length k as a function of λ, except for large values of λ, where the QMC method differs from the reference curve. In Fig. 5.2b the correlation function Eq. (5.17) and the fit Eq. (5.18) are plotted for the case with $\Delta\tau = 0.25$ and $\lambda = \lambda^* \approx 0.68$.

The fit parameter yields $k(\lambda^*) = (24.0 \pm 0.2)$ for $\tau \in [30, 50]$. Note that because of the small lattice spacing, the correlation length $k(\lambda^*)$ is four times larger than for the case with $\Delta\tau = 1$. In general, one has to find an appropriate τ partition for the fit. When τ is chosen too small, we get contributions of energies larger than E_1. But on the other hand for some realizations we cannot measure the correlation function up to $\tau = L/2$ since the errors become very high and we need to make a fit in between. Taking $\tau \in [k(\lambda^*), k(\lambda^*) + 20]$ for the fit has turned out to be a good rule.

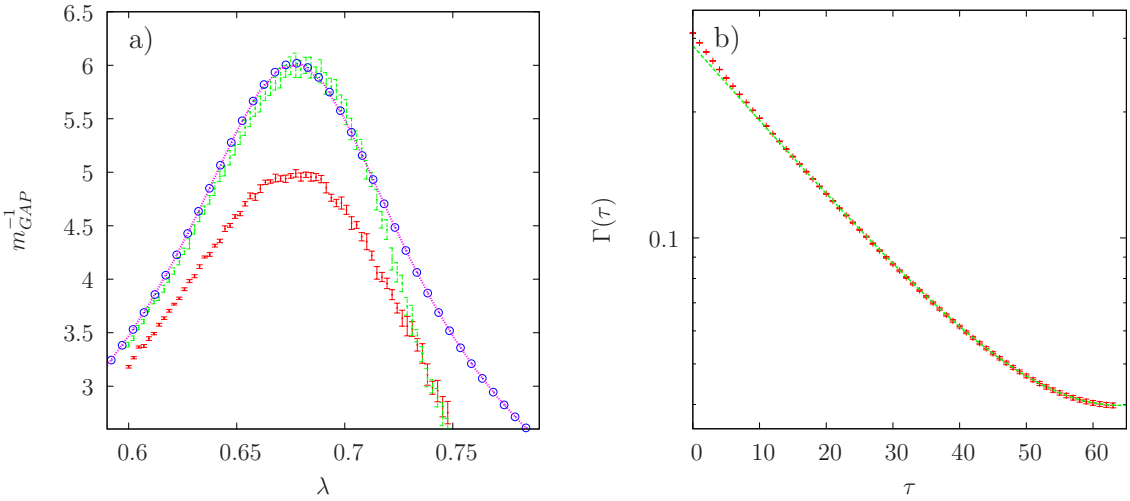

Figure 5.2: *a)* Inverse mass gap as a function of the adiabatic control parameter λ as obtained with different simulation methods for the 3-SAT problem with $N = 6$ and $r = 4$. Red dots: QMC method with $\Delta\tau = 1$; green dots: QMC method with $\Delta\tau = 0.25$; blue open circles: Lanczos method; magenta dashed line: exact diagonalization method. The QMC simulations are performed with $N_\tau = 128$.

b) Correlation function Γ as a function of τ (Eq. (5.17)) for $\Delta\tau = 0.25$ and $\lambda = \lambda^* \approx 0.68$ (red crosses) and the fit Eq. (5.18) (green line).

5.2 Results

5.2.1 Exact diagonalization method

For a small number of spins $N < 8$ we are able to diagonalize the QA Hamiltonian $H(\lambda)$ for many different λ values. We obtain the full spectrum of eigenvalues $e_k(\lambda)$ and eigenstates $\vec{s}_k(\lambda)$ for each value of $\lambda \in [0;1]$, where $k = 0, \ldots, 2^N - 1$ labels the eigenvalues so that $e_0(\lambda) \leq e_1(\lambda) \leq \cdots \leq e_{2^N-1}(\lambda)$. In Fig 5.3a the whole spectrum for the realization with $N = 6$ spins discussed in the previous section is plotted. We find for $\lambda = 0$ the spectrum of the Hamiltonian

$$H(0) = H_D = \sum_i^N \sigma_i^x, \tag{5.20}$$

which counts the number of spins pointing up- or downwards in the x-direction. The ground state $\vec{s}_0(0)$ of H_D is simply the state with all spins pointing downwards in the x-direction and therefore it is unique (USA). The corresponding energy evaluates to $e_0(0) = -N$. Note that this ground state $\vec{s}_0(0)$ viewed from the z-basis is an equal superposition of all 2^N possible states, which includes the sought-after state $\vec{s}_0(1)$. The DOS of the driver Hamiltonian can be written as $g_D(E_k) = \binom{N}{E_k+N}$ with $E_k \in [-N, \ldots, N]$. The spacing of the eigenvalues is $\Delta E = e_1(0) - e_0(0) = 2$, as every time a spin flips from -1 to 1 the energy differs by 2. So the mass gap for $\lambda = 0$ starts from the value $m_{GAP}(0) = 2$. As soon as we let $\lambda > 0$ and because the driver and problem Hamiltonian do not commute, we have to deal with quantum effects and the spectrum becomes complicated. Finally, for $\lambda = 1$ we reach the classical DOS function $g(E_i)$ with all its attributes discussed in section 4.2. The mass gap at $\lambda = 1$ is therefore fully determined by the classical DOS $m_{GAP}(1) = E_1 - E_0$ with $E_0 = e_0(1)$. In between the limits $\lambda = 0$ and $\lambda = 1$ we see that at some $\lambda = \lambda^*$, the mass gap becomes minimal. The code to produce Fig. 5.3a can be found in Appendix B.

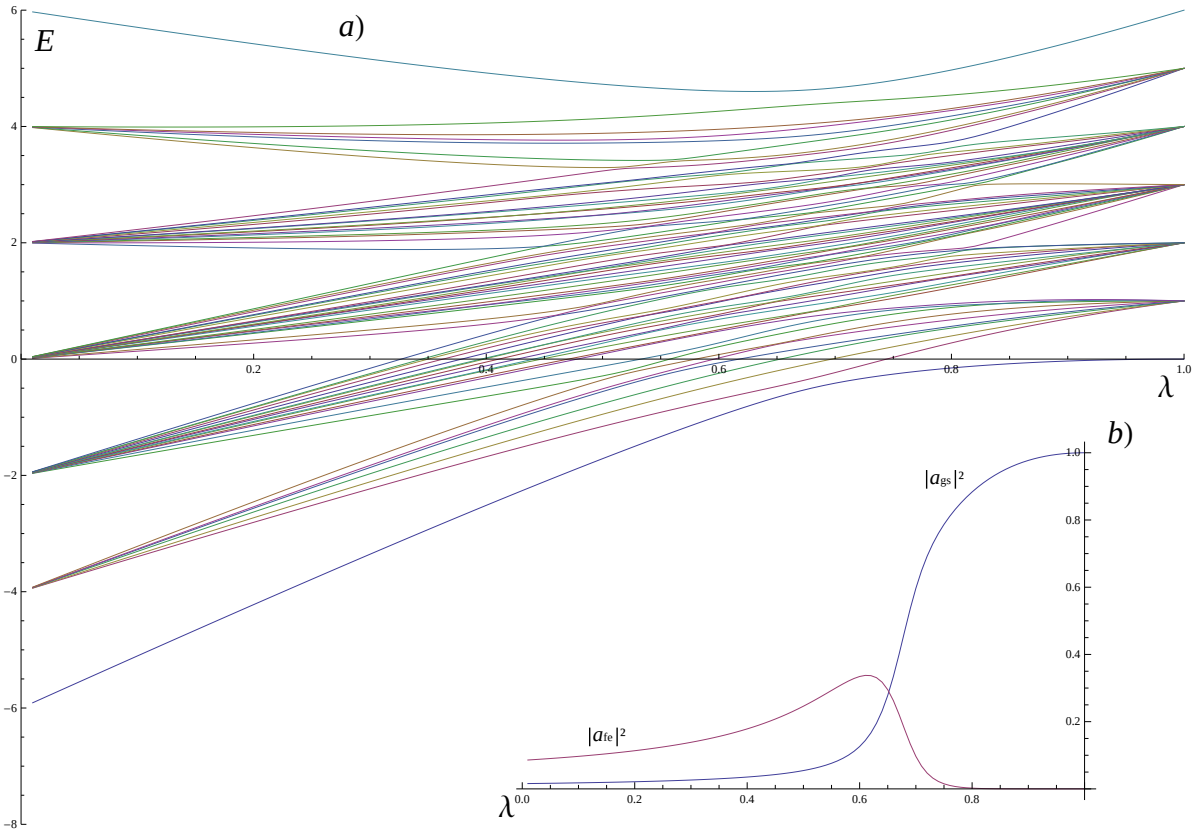

Figure 5.3: *a)* Eigenvalue spectrum for a realization with $N = 6$ and $r = 4$.
b) Probabilities to measure the first excited state $|a_{fe}|^2$ and the ground state $|a_{gs}|^2$ as a function of λ.

The ground state $\vec{s}_0(\lambda^*)$ of $H(\lambda^*)$ is still a superposition of many states in the z-direction

$$\vec{s}_0(\lambda^*) = \sum_{k'}^{2^N-1} b_{0,k'}(\lambda^*)\vec{s}_{k'}, \tag{5.21}$$

where k' denotes the integer representation of the state $\vec{s}_{k'}$ in the z-direction[3]. We use the symbol a to indicate the amplitude of the sought-after state $\vec{s}_0(1)$ in the z-direction, which is the ground state of the problem Hamiltonian. Consequently, we

[3]The index k is different from k' because k counts the eigenstates $e_k(\lambda)$ of the QA Hamiltonian $H(\lambda)$ with $e_0(\lambda) < e_1(\lambda) < \cdots < e_{2^N-1}(\lambda)$

define $a_{gs} = b_{0,k'}$, with k' so that $\vec{s}_{k'} = \vec{s}_0(1)$, to be the amplitude of the sought-after state in the superposition of states representing the ground state of $H(\lambda)$. The QA procedure takes care that the squared amplitude $|a_{gs}|^2$ increases with increasing λ and finally reaches $|a_{gs}|^2 = 1$ for $\lambda = 1$. When we look at the λ dependent amplitude squared, $|a_{gs}(\lambda)|^2$ we observe a rapid increase at $\lambda = \lambda^*$, as well as a rapid decrease of $|a_{fe}(\lambda)|^2$, which is the squared amplitude of the sought-after state in the superposition of states representing the first excited state of $H(\lambda)$ (Fig. 5.3b). Furthermore we can see that the system has to get through the first excited states, as up to some $\lambda \approx \lambda^*$, $|a_{fe}|^2 > |a_{gs}|^2$. In addition, at the maximum of $|a_{fe}(\lambda)|^2$ it is more probable to find the sought-after state to be the first excited state $\vec{s}_1(\lambda)$ of $H(\lambda)$ than the ground state. When we simulate hard realizations with $N = 8$ spins, the effect becomes more profound and the peak in $|a_{fe}(\lambda)|^2$ becomes higher (not shown). This seems to be similar to the classical case, where the system has to pass through a large number of first excited states to finally reach the ground state. It would be interesting for future research to study whether the states that contribute to $\vec{s}_1(\lambda)$ in the z-basis are the same as the classical first excited states. However, the disadvantage of the exact diagonalization method is that we cannot handle much more spins than $N = 8$ and therefore every effect that we observe with this method can probably be due to the finite system size, and of course we are not able to make a prediction for $N \to \infty$.

5.2.2 Lanczos method

The Lanczos method has the disadvantage that it can only simulate small systems with size up to $N = 20$. On the other hand it can determine very small mass gaps m_{GAP}. So we use this method to make simulations for $r = 4.5$, near the phase transition ($r \approx 4.2$) of the random 3-SAT problem. For this range of r-values the realizations are very hard to be solved by classical algorithms. In the quantum systems, this property is translated to the presence of very small mass gaps, the smallest being of the order 10^{-6}. QMC

is not capable of simulating such realizations effectively because we need a very large lattice size N_τ (much larger than the $N_\tau = 128$ or $N_\tau = 256$ that we usually use) and therefore much more statistics to obtain small errors. We are not able to obtain a scaling plot of the mass gap as a function of the number of spins, since we can only simulate up to $N = 20$ spins. Moreover, we also have serious boundary effects, as we have observed in the scaling of the classical complexity. Fig. 5.4a shows the maximum correlation length $\max_{0<\lambda<1}(k(\lambda)) = k(\lambda^*)$ as a function of the barrier B_0 of the classical canonical distribution at $\beta = \beta_{\mathrm{CMAX}}$ as it is defined in section 4.2.4. Note that the barrier B_0 is defined as a logarithmic quantity so that the relation between the correlation length $k(\lambda^*)$ and B_0 is linear on a logarithmic scale. Performing the linear fit $f(x) = ax + b$ to the logarithm of the correlation length yields a slope of $a = (1.55 \pm 0.86)$ with a relative error of 55%. The slope is of the order of 1 so that the barrier, as a static property of the system, directly defines the hardness of the realization [64]. The large error indicates that not only the barrier determines the hardness of the realization but that there must be other effects too. Nevertheless, we do not find a single realization with a large barrier B_0 and a small correlation length $k(\lambda^*)$. The two lines in Fig. 5.4a represent the maximum $k(\lambda^*)$ that we are able to determine with the QMC method on a $N_\tau = 128$ or 256 lattice. Note that doubling the lattice size would not help to solve tho problem. We need a 1000 times larger lattice, which we are not able to simulate with the given ressources.

We are able to compare the Lanczos method and the QMC method for a larger number of realizations for the case with $r = 7.5$ and for $N = 20$ spins. For a correlation length $k(\lambda^*)$ larger than N_τ, we expect the QMC method to produce errors, because then only a fraction of the exponential decrease can be fitted by Eq. (5.18). Nevertheless, it gives an estimate for $k(\lambda^*)$. In Fig. 5.4b a scatter plot is depicted of the correlation length obtained with the Lanczos and the QMC methods. The QMC results are obtained for $N_\tau = 128$ and therefore we find large errors for $k(\lambda^*)$ near 64. Another source of errors

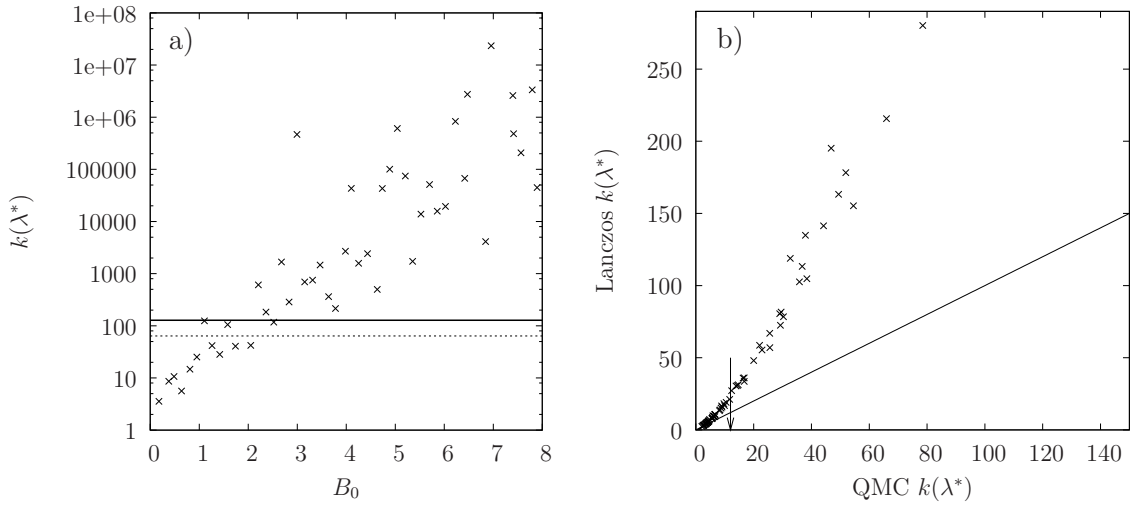

Figure 5.4: *a)* Minimum inverse mass gap $m_{GAP}^{-1}(\lambda^*) = k(\lambda^*)$ as a function of the barrier B_0 of the classical canonical distribution at $\beta = \beta_{\mathrm{CMAX}}$. The mass gap is obtained by using the Lanczos method for realizations with $r = 4.5$ and $N = 20$. The lines represent the maximum $k(\lambda^*)$ that we are able to determine with the QMC method on a $N_\tau = 128$ (dashed) or a $N_\tau = 256$ (solid) lattice. *b)* Scatter plot of the correlation length $k(\lambda^*)$ obtained by the QMC and the Lanczos methods for realizations with $r = 7.5$ and $N = 20$. The arrow indicates the correlation length at the median of the barrier B_0; the line represents the function $f(x) = x$.

is the chosen λ partition. Usually we first make a coarse run with $0.3 < \lambda < 0.8$ and only 100 000 sweeps in order to find the quantum phase transition point λ^*, followed by simulations with λ in a more refined interval around λ^*. However, with increasing correlation length $k(\lambda^*)$ the peak becomes narrower so that the interval may become too coarse to cover the top of the peak and as a result we sample only values at the flank of the peak. For some realizations we refined the λ interval 8 times, in order to resolve the peak, with the Lanczos method. This refining procedure is too expensive for the QMC method, so we constrained ourselves to only one refinement step.

The arrow in Fig. 5.4b marks the correlation length at the median of the barrier B_0, and at this position the difference between the results obtained with the Lanczos and the QMC method is rather small. There is only a factor of ≈ 1.7 difference between the results obtained with the Lanczos method and the one obtained with the QMC method. So, in practice we take the correlation length k_{median} corresponding to the median of the barrier, which is the typical value of B_0. Because of the correlation between $k(\lambda^*)$ and B_0, the value k_{median} is the typical value of the correlation length. Hence, we assume the distribution of the inverse mass gap to be the same as the distribution of the barrier B_0. Therefore it is important that at the median of B_0 the error of the correlation length is small. Furthermore, Fig. 5.4b shows that the QMC results underestimate the correlation length systematically compared with the results obtained with the Lanczos method. We conclude that the correlation length obtained with the QMC method is in general too small.

5.2.3 Quantum Monte Carlo method

Here we describe the results obtained by the QMC method as introduced in section 5.1. Using the definition $A(\tau) = o_{gs}(\vec{s}(\tau))$ we can interprete the quantity called q in Eq. (5.19) as the mean ground-state overlap of the whole system at a given λ value. Note that o_{gs} in the quantum case remains the same as in the classical case, it is the overlap to the ground state of the problem Hamiltonian. For one particular realization the curve $q(\lambda)$ is shown in Fig. 5.5a. The curve $q(\lambda)$ shows a jump for many realizations that are simulated. Because of the refinement of the λ interval there are more data points around the jump, which is located at $\lambda = \lambda^*$. Remember that at λ^* the mass gap of the system is minimal and that at this point the quantum phase transition occurs. Furthermore we can see the first similarity between the quantized system and the classical one: The global minimum of $q(\lambda)$ seems to be correlated to the position in ground-state overlap space of the left maximum of the canonical distribution at $\beta = \beta_{\text{CMAX}}$. The correlation is rather

hard to establish, because we are not able to simulate the whole ensemble with the QMC method. The main problem is that the barrier is only well defined for large values of B_0 but because of the relation between B_0 and $k(\lambda^*)$ the correlation length then becomes too large for the chosen N_τ lattice. We therefore only have a very small window with about ten realizations that fit into the conditions $B_0 > 1.5$ and $20 < k(\lambda^*) < 64$ at $N = 20$. We consider the correlation between $\min_{0<\lambda<1} q(\lambda)$ and the left maximum of the canonical distribution as an indication of the fact that the quantized system has to take the same path as the classical system to find the solution and therefore both systems should spot the same free energy barrier. We use the microcanonical distribution $p(E, o_{gs})$ for a hard realization (see Fig. 4.9a) to illustrate this: The classical system, coming from high energies, follows the most probable path of $p(E, o_{gs})$ to reach the ground state, which leads away from the ground state until a specific value of the ground-state overlap (around $o_{gs} = 0.4$ for hard realizations). Since this value is nearly the same in both the classical and quantum systems, we conclude that the QA algorithm has the same problems of finding the ground state as local search algorithms.

As previously mentioned, the barrier B_0 and the correlation length $k(\lambda^*)$ are correlated. Fig. 5.5b shows an example for $N = 20$ and $r = 7.5$. We also have established the correlation for $N = 30, 40, 50$ and also for $r = 8$. We see a linear relation between B_0 and the logarithmic correlation length $k(\lambda^*)$ for every realization that is simulated. Since the correlation length equals the inverse mass gap, which bounds the runtime of the algorithm Eq. (5.12), we draw the conclusion that the runtime of the QA algorithm for the 3-SAT problem is proportional to the barrier squared B_0^2. This is the next indication, that the quantized system spots the same problems as the classical one. Unfortunately, we are not able to simulate the whole ensemble of realizations, but with the typical value of B_0 we can predict the typical value of the correlation length $k_{\text{median}}(\lambda^*)$. So we restrict ourselves to realizations around the predicted median of $k_{\text{median}}(\lambda^*)$. Since the linear relation between B_0 and $k(\lambda^*)$ is not perfect, we choose various realizations in the

vicinity of the median of B_0 for each particular r and N and compute the mean value and the variance of $k(\lambda^*)$ for them. We obtain the typical scaling of the QA algorithm with N for our USA realizations. In Fig. 5.6 this scaling is plotted on a logarithmic scale and we see a perfect exponential relation $k_{\text{median}}(N) = A_{QA}\, \mathrm{e}^{c_{QA}N}$ with fit parameters $A_{QA} = (1.95 \pm 0.04)$ and $c_{QA} = (0.061 \pm 0.001)$. We can exclude a polynomial scaling because we are not able to fit a single polynomial function to our data. When we compare the QA complexity c_{QA} for $r = 8$ to the classical one obtained with the Wang-Landau algorithm $c_{r=8} = (0.019 \pm 0.001)$ we find that the QA algorithm has a three times larger complexity than the classical one. Please note that this calculation only considers the complexity, not the actual runtime of our algorithms.

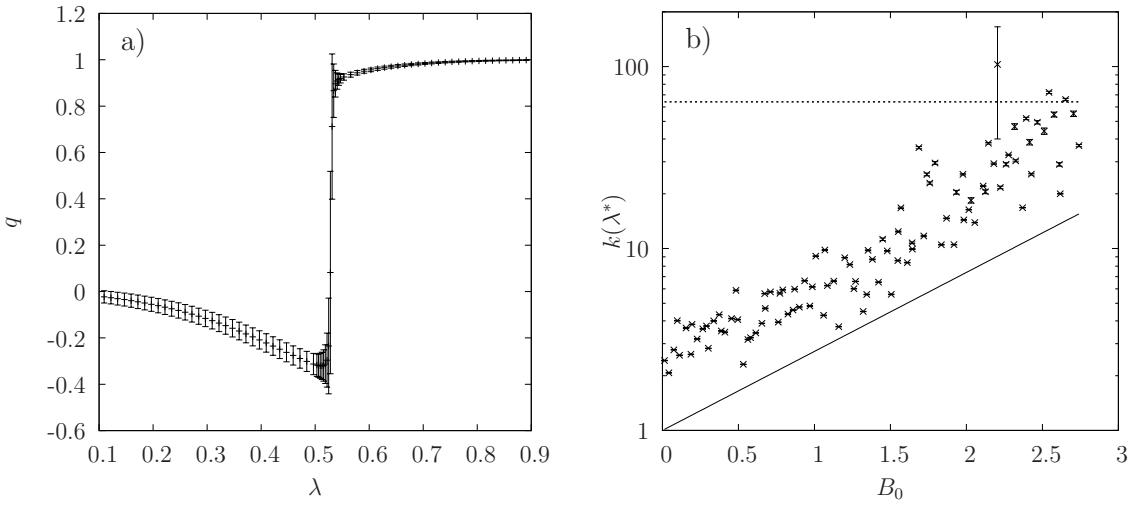

Figure 5.5: *a)* Function $q(\lambda)$ for one realization of the 3-SAT problem with $N = 20$ and $r = 7.5$.

b) Correlation length $k(\lambda^*)$ on a logarithmic scale as a function of the barrier B_0. The constant $L_\tau/2 = 64$ is marked by the dashed line and the function $f(x) = \mathrm{e}^x$ by the solid one. ($N = 20$, $r = 7.5$)

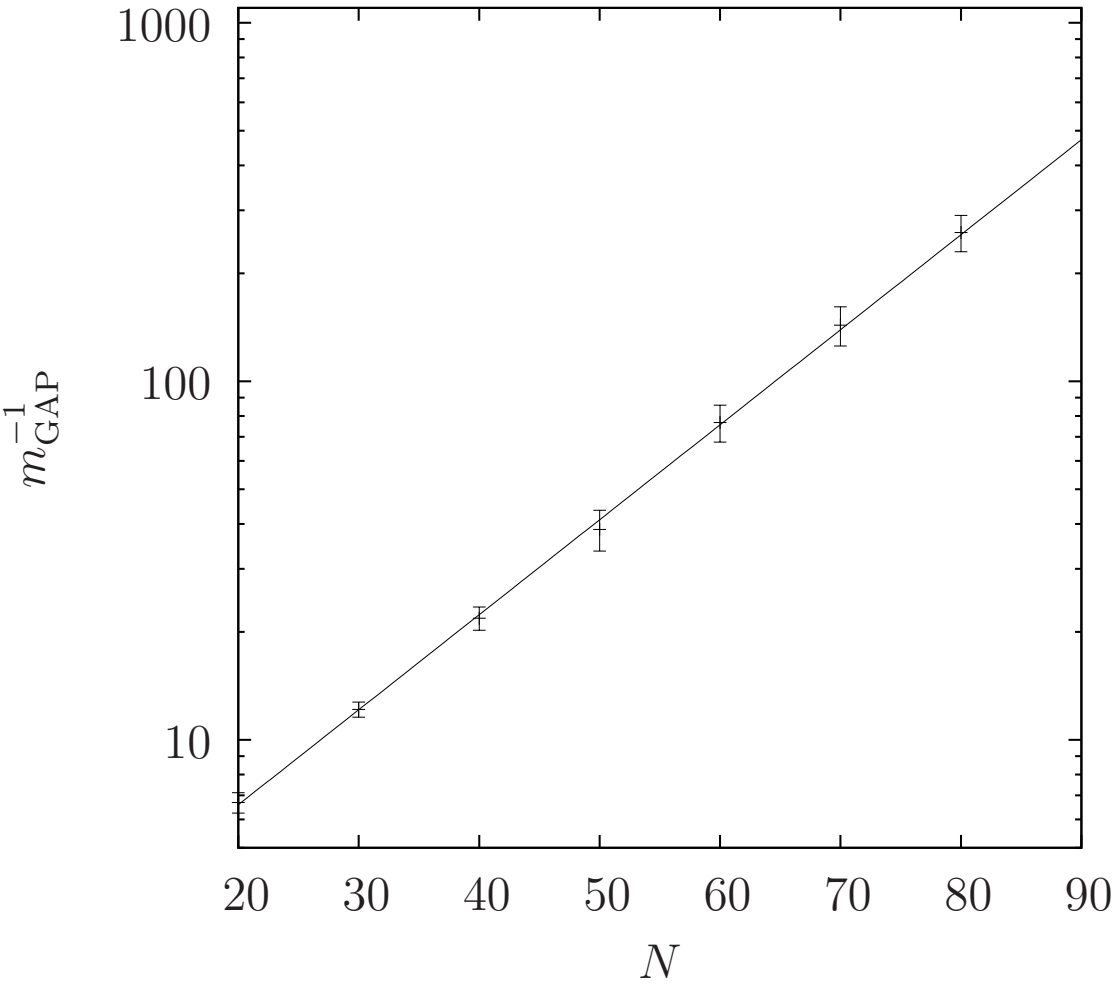

Figure 5.6: Inverse mass gap m_{GAP}^{-1} as a function of N of the QA algorithm for $r = 8$. The green line represents the fit of an exponential function to the data (see text).

Chapter 6

Conclusion and outlook

The 3-SAT problem for the case of special USA (unique satisfying assignment) realizations has been studied in a classical and in a quantum mechanical way. For the classical simulations Wang-Landau and Multicanonical sampling is utilized to estimate the density of states (DOS) of an ensemble of 4000 realizations for different N and r values. Using the DOS and other output of the sampling algorithms, many thermodynamic functions and distributions are obtained. Energy, specific heat and microcanonical and canonical distributions for USA realizations are discussed. Wang-Landau sampling is also employed in the ground-state overlap direction, rather than in the energy direction, and the canonical distribution is measured. A static property of our realizations, denoted as barrier B_0, which is a probability ratio in the canonical distribution, can be linear related to the logarithm of the ground state search time τ of the Multicanonic sampling in energy direction for high $r > 7$. For all values of r considered in this work, the typical search time for the unique ground state scales exponentially with the number of spins.

In the second part of this work, the quantum adiabatic (QA) algorithm is considered for solving the 3-SAT problem. For small systems, the Lanczos algorithm and the

full diagonalization method of the QA Hamiltonian are used to get information about the energy spectrum. By use of quantum Monte Carlo (QMC) together with Parallel Tempering, larger systems with up to $N = 80$ are simulated for the first time. The classical barrier B_0 is related to the quantum mechanical mass gap. Using the relation between B_0 and the mass gap the typical runtime of the QA algorithm is obtained and it scales exponentially with the number of spins N. This fact is shown for high $r = 8$ for the first time.

The correlation of the classical barrier B_0 to the mass gap is a first evidence that the QA algorithm is not a better method to solve the 3-SAT problem than classical local search algorithms. However, our simulations cover only one linear combination of driver and problem Hamiltonian. For some problems it may help to change this linear combinations, the so called path, of driver and problem Hamiltonian [65, 66]. But for a new problem it is a priori unclear which path provides the best chance to solve it efficiently. Another idea to fix the QA algorithm is to use an heuristic algorithm in combination with a modified version of the QA Hamiltonian. We performed N different runs of the algorithm. For each run i we add a term $+\sigma_i^z$ applied on spin i to the driver Hamiltonian followed by a run with $-\sigma_i^z$ and measure the position of the quantum phase transition. For a single realization with $N = 60$ we find that by applying the correct sign to the spin, the quantum phase transition moves to smaller λ. So the problem of measuring the state the end of the algorithm is transformed to a problem of measuring the position of the quantum phase transition. The shift of the quantum phase transition may or may not cancel out for $N \to \infty$ or for a large ensemble of realizations. The Parallel Tempering in the λ-space also deserves future research, because it is unclear why it helps so much when the system is driven through the quantum phase transition. Another interesting approach to quantize optimization problems is to take advantage of the Jarzinsky equality and simulate apart from the thermodynamical equilibrium [67].

The classical part of this work lacks a parallel version of the Wang-Landau algorithm

that scales up to a large number of processors ($p > 100$) for the 3-SAT problem. This is not only relevant for the 3-SAT problem but also for many other problems in statistical physics, since the development of the hardware points towards building computers with a much higher number of relatively slow processors. Concerning 3-SAT, it would be promising to construct the so-called quenched mean, which is the mean over the disorder of an ensemble, of the physical quantities that were discussed here to fully understand our realizations. Since the first excited states seem to play a prominent role for local search algorithms, it would also be interesting to analyze their structure. The Wang-Landau algorithm can be used to achieve a sample of first excited states and to analyze them for backbones, clusters and overlaps.

Appendix A

Wang Landau algorithm

The following code is the heart of our parallel Wang-Landau algorithm. It is implemented using C and MPI API. Each cycle of the `while` loop we perform `SWEEPS_PER_FLATTEST` number of sweeps and processor zero sums up the histogram from the other processors. When the whole histogram is flat each processor calculates $h(E_i)\log(g(E_i))$ and sends this to processor zero, where $\log(g_{weight}(E_i))$ is achieved and broadcasted back to the other processors. Then $h(E_i) = 0$ is set to zero and the factor f is decremented. The algorithm ends either when we have reached the accuracy that we want, or when the counter `sw` exceeds the limit `MAX_SWEEPS`. The names for the send and receive buffers get an additional `M` as they are defined as an array of length M (number of clauses) followed by a short form of their type, e.g. `d` for a double precission array.

```
f = 1;
sw = 0;
while(sw < MAX_SWEEPS && f > 0.0000001){
  ++sw;
  for (i=0;i < SWEEPS_PER_FLATTEST;i++) sweep();
  synchronizeEnergyInterval();
```

```
  memcpy(sendBufMll, h, sizeof(h));
  MPI_Reduce(sendBufMll, recvBufMll, (M+1), MPI_LONG_LONG, +
       MPI_SUM, 0, MPI_COMM_WORLD);
  if (myRank == 0){
    flatFlag = isFlatNew(recvBufMll);
  }
  MPI_Bcast(&flatFlag, 1, MPI_INT, 0, MPI_COMM_WORLD);

  if (flatFlag != 0) {
    for (i=0;i <= M;i++) sendBufMd[i] = ((double) h[i])*log_g[i];
    MPI_Reduce(sendBufMd, log_g, (M+1), MPI_DOUBLE, +
         MPI_SUM, 0, MPI_COMM_WORLD);
    for (i=0;i <= M;i++) if (recvBufMll[i] == 0) recvBufMll[i] = 1;
    for (i=0;i <= M;i++) log_g[i] = (double)log_g[i] / (recvBufMll[i]);
    MPI_Bcast(log_g, M+1, MPI_DOUBLE, 0, MPI_COMM_WORLD);

    for (i=0;i <= M;i++) h[i] = 0;
    f *= 0.5;
    flatFlag = 0;
  }
}
normalize(log_g);
```

Appendix B

Mathematica program

Here we describe the Mathematica program to diagonalize the QA Hamiltonian. At first we encode our realization, for example with $N = 6$, in a diagonal matrix. The energies of all possible states in the order of the computational basis states form the diagonal elements of the matrix. Hence, the first number corresponds to the state with all spins down and it violates four clauses:

```
Hproblem =
DiagonalMatrix[{4, 4, 5, 5, 1, 3, 2, 3, 3, 3, 3, 4, 2, 3, 2, 3, 4,
  3, 6, 4, 4, 5, 4, 3, 2, 0, 3, 1, 4, 3, 3, 1, 3, 1, 2, 1, 1, 3, 1,
  3, 4, 2, 2, 2, 2, 3, 1, 3, 2, 1, 4, 3, 2, 5, 3, 5, 4, 2, 5, 4, 4,
  5, 4, 5}];
```

Note that there is only one entry, at position 26, with energy zero, so the realization is USA. The next step is to initialize the driver Hamiltonian:

```
sx = {{0, 1}, {1, 0}};
eins = {{1, 0}, {0, 1}};
```

```
sx1 = KroneckerProduct[eins, eins, eins, eins, eins, sx];
sx2 = KroneckerProduct[eins, eins, eins, eins, sx, eins];
sx3 = KroneckerProduct[eins, eins, eins, sx, eins, eins];
sx4 = KroneckerProduct[eins, eins, sx, eins, eins, eins];
sx5 = KroneckerProduct[eins, sx, eins, eins, eins, eins];
sx6 = KroneckerProduct[ sx, eins, eins, eins, eins, eins];
Hdriver = sx1 + sx2 + sx3 + sx4 + sx5 + sx6;
```

The variable `steps` defines the number of different λ-values, the resulting energies are written in the array `listAll` and the difference between the energy of the ground state and the energy of the first excited state is written in the array `listMass`.

```
For[i = 1, i <= steps, i++,
  lambda = i/steps // N;
  H = (1 - lambda)*Hdriver + lambda*Hproblem;
  eigensystem = Eigensystem[H];
  eigen = eigensystem[[1]];
  eigen = Sort[Re[eigen]];

  For[k = 1, k <= 2^vars, k++,
    listAll[[k, i, 2]] = eigen[[k]];
    listAll[[k, i, 1]] = lambda;
  ];
  listMass[[i, 1]] = lambda;
  listMass[[i, 2]] = eigen[[2]] - eigen[[1]];
]
```

With the ListLinePlot command of Mathematica we can plot the energies of all states of the quantized system

```
ListLinePlot[listAll]
```

as well as the mass gap $m_{GAP}(\lambda)$

```
ListPlot[listMass]
```

The code above was used to produce the figure 5.3a.

Bibliography

[1] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. on Computing **26**, 1484 (1997)

[2] L.K. Grover, in *28th Annual ACM Symposium on the Theory of Computing* (ACM, Philadelphia, 1996), p. 212

[3] L. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, I.L. Chuang, Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, Nature **414**, 883 (2001)

[4] C.Y. Lu, D.E. Browne, T. Yang, J.W. Pan, Demonstration of a compiled version of Shor's quantum factoring algorithm using photonic qubits, Phys. Rev. Lett. **99**(25), 250504 (2007)

[5] J.A. Jones, M. Mosca, R.H. Hansen, Implementation of a quantum search algorithm on a quantum computer, Nature **393**, 344 (1998)

[6] T.D. Ladd, R. Laflamme, Y. Nakamura, C. Monroe, J.L. O'Brien, Quantum computers, Nature **46**, 45 (2010)

[7] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000)

[8] W.K. Wootters, W.H. Zurek, A single quantum cannot be cloned, Nature **299**(5886), 802 (1982)

[9] T. Kadowaki, H. Nishimori, Quantum annealing in the transverse Ising model, Phys. Rev. E **58**(5), 5355 (1998)

[10] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem, Science **292**(5516), 472 (2001)

[11] M. Born, V.A. Fock, Beweis des adiabatensatzes, Zeitschrift für Physik A **51**, 165 (1928)

[12] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, O. Regev, Adiabatic quantum computation is equivalent to standard quantum computation, SIAM Journal on Computing **37**, 166 (2007)

[13] A.P. Young, S. Knysh, V.N. Smelyanskiy, Size dependence of the minimum excitation gap in the quantum adiabatic algorithm, Phys. Rev. Lett. **101**(17), 170503 (2008)

[14] A.P. Young, S. Knysh, V.N. Smelyanskiy, First-order phase transition in the quantum adiabatic algorithm, Phys. Rev. Lett. **104**(2), 020502 (2010)

[15] R. Schützhold, G. Schaller, Adiabatic quantum algorithms as quantum phase transitions: First versus second order, Phys. Rev. A **74**(6), 060304 (2006)

[16] S. Cook, in *The Millennium Prize Problem* (Clay Mathematical Institute, Cambridge, 2000)

[17] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, New York, 1979)

[18] S.A. Cook, in *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing* (ACM, New York, 1971), p. 151

[19] P. Chen, K. Keutzer, in *ICCAD '99: Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design* (IEEE Press, Piscataway, 1999), p. 132

[20] H. Kautz, B. Selman, in *ECAI '92: Proceedings of the 10th European conference on Artificial intelligence* (John Wiley & Sons, Inc., New York, 1992), p. 359

[21] E. Clarke, D. Kroening, F. Lerda, in *In Tools and Algorithms for the Construction and Analysis of Systems* (Springer, Barcelona, 2004), p. 168

[22] M. Davis, G. Logemann, D. Loveland, A machine program for theorem-proving, Commun. ACM **5**(7), 394 (1962)

[23] B. Selman, H.A. Kautz, B. Cohen, in *DIMACS Series in discrete mathematics and theoretical computer science* (American Mathematical Society, 1996), p. 521

[24] http://www.princeton.edu/ chaff/

[25] R. Monasson, R. Zecchina, Entropy of the k-satisfiability problem, Phys. Rev. Lett. **76**(21), 3881 (1996)

[26] S. Kirkpatrick, G. Györgyi, N. Tishby, L. Troyansky, in *NIPS, The Statistical Mechanics of k-Satisfaction* (1993), p. 439

[27] M. Mezard, G. Parisi, R. Zecchina, Analytic and algorithmic solution of random satisfiability problems, Science **297**(5582), 812 (2002)

[28] M. Mézard, R. Zecchina, Random k-satisfiability problem: From an analytic solution to an efficient algorithm, Phys. Rev. E **66**(5), 056126 (2002)

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

[30] D. Rolf, Improved bound for the PPSZ/Schning-algorithm for 3-SAT, Electronic Colloquium on Computational Complexity **159**, 2006 (2005)

[31] http://www.satlib.org/

[32] S. Horie, O. Watanabe, *Technical Report: Hard Instance Generation for SAT* (Tokyo Institute of Technology, Tokyo, 1996)

[33] H.H. Hoos, T. Stützle, Local search algorithms for SAT: An empirical evaluation, J. Autom. Reasoning **24**(4), 421 (2000)

[34] H. Jia, C. Moore, B. Selman, in *In Proceedings of SAT 2004* (Springer, Vancouver, 2004), p. 199

[35] W. Barthel, A.K. Hartmann, M. Leone, F. Ricci-Tersenghi, M. Weigt, R. Zecchina, Hiding solutions in random satisfiability problems: A statistical mechanics approach, Phys. Rev. Lett. **88**(18), 188701 (2002)

[36] S. Kirkpatrick, C.D. Gelatt, Jr., M.P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983)

[37] F. Wang, D.P. Landau, Efficient, multiple-range random walk algorithm to calculate the density of states, Phys. Rev. Lett. **86**(10), 2050 (2001)

[38] B.A. Berg, T. Neuhaus, Multicanonical ensemble: A new approach to simulate first-order phase transitions, Phys. Rev. Lett. **68**(1), 9 (1992)

[39] R.G. Miller, The Jackknife - a review, Biometrika **61**(1), 1 (1974)

[40] L. Zhan, A parallel implementation of the Wang-Landau algorithm, Computer Physics Communications **179**(5), 339 (2008)

[41] S.A. Cook, D.G. Mitchell, in *Finding hard instances of the Satisfiability problem: A survey* (American Mathematical Society, 1997), p. 1

[42] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, Determining computational complexity from characteristic 'phase transitions', Nature **400**(6740), 133 (1999)

[43] J. Lukic, A. Galluccio, E. Marinari, O.C. Martin, G. Rinaldi, Critical thermodynamics of the two-dimensional $\pm J$ Ising spin glass, Phys. Rev. Lett. **92**(11), 117202 (2004)

[44] M. Belaidouni, J.K. Hao, in *Selected Papers from the 5th European Conference on Artificial Evolution* (Springer-Verlag, London, 2002), p. 192

[45] H. Chen, C. Gomes, B. Selman, in *Principles and practice of constraint programming* (Springer-Verlag, 2001), p. 408

[46] B.A. Berg, U. Hansmann, T. Neuhaus, Simulation of an ensemble with varying magnetic field: A numerical determination of the order-order interface tension in the d=2 ising model, Phys. Rev. B **47**(1) (1993)

[47] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Quantum computation by adiabatic evolution (2000). URL http://arxiv.org/abs/quant-ph/0001106

[48] K. Marzlin, B.C. Sanders, Inconsistency in the application of the adiabatic theorem, Phys. Rev. Lett. **93**(16), 160408 (2004)

[49] D.M. Tong, Quantitative condition is necessary in guaranteeing the validity of the adiabatic approximation, Phys. Rev. Lett. **104**(12), 120401 (2010)

[50] M.H.S. Amin, Consistency of the adiabatic theorem, Phys. Rev. Lett. **102**(22), 220401 (2009)

[51] Y. Zhao, Reexamination of the quantum adiabatic theorem, Phys. Rev. A **77**(3), 032109 (2008)

[52] L. Landau, Zur Theorie der Energieubertragung, Physics of the Soviet Union **2**, 46 (1932)

[53] L. Landau, Non-adiabatic crossing of energy levels, Proceedings of the Royal Society of London **137**(6), 696 (1932)

[54] J. Roland, N.J. Cerf, Quantum search by local adiabatic evolution, Phys. Rev. A **65**(4), 042308 (2002)

[55] A.M. Childs, E. Farhi, J. Preskill, Robustness of adiabatic quantum computation, Phys. Rev. A **65**(1), 012322 (2001)

[56] M.H.S. Amin, D.V. Averin, J.A. Nesteroff, Decoherence in adiabatic quantum computation, Phys. Rev. A **79**(2), 022107 (2009)

[57] M.H.S. Amin, Effect of local minima on adiabatic quantum optimization, Phys. Rev. Lett. **100**(13), 130503 (2008)

[58] M. Žnidarič, Scaling of the running time of the quantum adiabatic algorithm for propositional satisfiability, Phys. Rev. A **71**(6), 062305 (2005)

[59] M.P. Nightingale, V.S. Viswanath, G. Müller, Computation of dominant eigenvalues and eigenvectors: A comparative study of algorithms, Phys. Rev. B **48**(10), 7696 (1993)

[60] M. Suzuki, Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems, Communications in Mathematical Physics **51**(2), 183 (1976)

[61] J.B. Kogut, An introduction to lattice gauge theory and spin systems, Rev. Mod. Phys. **51**(4), 659 (1979)

[62] R.H. Swendsen, J.S. Wang, Replica Monte Carlo simulation of spin-glasses, Phys. Rev. Lett. **57**(21), 2607 (1986)

[63] D.J. Earl, M.W. Deem, Parallel tempering: Theory, applications, and new perspectives, Phys. Chem. Chem. Phys. **7**(23), 3910 (2005)

[64] T. Neuhaus, M. Peschina, K. Michielsen, H. De Raedt, Failure of quantum adiabatic computation in the median of three satisfiability, submitted (2010)

[65] A. Boulatov, V.N. Smelyanskiy, Quantum adiabatic algorithms and large spin tunnelling, Phys. Rev. A **68**(6), 062321 (2003)

[66] E. Farhi, J. Goldstone, S. Gutmann, Quantum adiabatic evolution algorithms with different paths (2002). URL http://arxiv.org/abs/quant-ph/0208135

[67] M. Ohzeki, H. Nishimori, Quantum annealing: An introduction and new developments, to appear in J. Comp. Theor. Nanoscience (2010)

Jül-4334
November 2010
ISSN 0944-2952