Classical and Quantum Computation in Ground States and Beyond

Elizabeth Crosson

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2015

Reading Committee:

Marcel den Nijs, Chair

Aram Harrow

Chris Laumann

Program Authorized to Offer Degree:
Department of Physics

University of Washington

Abstract

Classical and Quantum Computation in Ground States and Beyond

Elizabeth Crosson

Chair of the Supervisory Committee:
Marcel den Nijs
Physics

In this dissertation we study classical and quantum spin systems with applications to the theory of computation. In particular, we examine computational aspects of these systems beyond their ground states by considering the effects of non-zero temperatures and excited energy states. The results within can be divided into four main categories.

In the first part we show that universal classical computations can be encoded into equilibrium thermal states of classical spin systems, at sufficiently low temperatures which are independent of the system size. This extension of ground-state spin computing to a more realistic setting involving non-zero temperatures can be viewed as a form of fault-tolerance for classical computation. The demonstration of this result relies on a strong understanding of the equilibrium distribution of these spin systems, including an exact solution for the partition function. We also pose open questions about the non-equilibrium dynamics of these thermal circuits, as well as about generalizations to quantum models.

In the second part we explore different strategies for optimization with the quantum adiabatic algorithm. We show that it is possible to increase the success probability for hard random instances by following the counterintuitive strategy of evolving along the Hamiltonian path more rapidly. All of the hard instances in the ensemble of our large-scale computational study were improved according to this strategy, and we explain the mechanism of this success in terms of the exchange of amplitudes that occurs at avoided crossings between energy levels.

In the third part we examine the performance of simulated quantum annealing in finding the minimum of an energy function which contains a high energy barrier. Classical simulated annealing takes exponential time to cross this energy barrier by thermal fluctuations, and so it becomes trapped in a local minimum. In contrast, quantum annealing can efficiently tunnel through the barrier and escape the local minimum. We use the path-integral quantum Monte Carlo method to sample from the thermal equilibrium state of a quantum annealer, and provide evidence that simulated quantum annealing inherits some of the advantages of quantum annealing which allow it to pass through the high barrier and minimize the energy function efficiently.

In the last part we show that the path-integral quantum Monte Carlo method leads to a provably efficient algorithm for approximating the partition function of any 1D generalized transverse Ising spin chain (with position-dependent local fields and frustrated interactions), at temperatures which are independent of the system size. Generalized transverse Ising models are used in the traditional forms of quantum annealing, which motivates our investigation of the computational complexity of simulating these models. Our main contribution is to formalize the QMC sampling procedure in terms of a Markov chain, and to show that this Markov chain is rapidly mixing in this 1D setting. The version of QMC we consider maps 1D quantum spin chains to 2D anisotropic classical spin models, and our result on rapid mixing can be interpreted as demonstrating that the non-equilibrium dynamics of the 2D classical system thermalize in an amount of time which scales polynomially with the system size.

TABLE OF CONTENTS

GLOSSARY

COMBINATORIAL OPTIMIZATION: Finding the extreme values of a function on a discrete domain, e.g. finding the minimum of a real-valued function $f : \{0,1\}^n \to \mathbb{R}$ whose domain is the set of n-bit strings.

FAULT-TOLERANCE: : A scheme for turning noisy components into a computer that functions correctly with high probability.

QUANTUM ADIABATIC ALGORITHM: A method for preparing the ground state $|\psi\rangle$ of any Hamiltonian H by following a parameterized path in the space of Hamiltonians and remaining in the ground state at all times according to the adiabatic theorem.

QUANTUM ADIABATIC OPTIMIZATION: An application of the quantum adiabatic algorithm to problems in combinatorial optimization. The function to be minimized is represented by a Hamiltonian whose ground state corresponds to the minimum of the function.

QUANTUM ANNEALING: A more realistic version of quantum adiabatic optimization in which the system is subject to thermal noise.

SIMULATED QUANTUM ANNEALING: A classical algorithm, based on quantum Monte Carlo, that simulates quantum annealing with a random walk that samples from the thermal equilibrium Gibbs distribution of the system.

SIMULATED ANNEALING: A classical algorithm for combinatorial optimization problems which is based on a random walk that simulates the nonequilibrium thermal effects in a classical spin system as it is cooled from high to low temperature.

STOQUASTIC HAMILTONIAN: A Hamiltonian which has all of its off-diagonal entries being real and non-positive in a particular basis. Equivalently, the Hamiltonian does not have a sign problem.

QUANTUM MONTE CARLO: A family of classical algorithms which use random walks to simulate stoquastic Hamiltonians.

PATH-INTEGRAL QUANTUM MONTE CARLO: A version of quantum Monte Carlo that performs a random walk on the space of worldlines of the quantum system.

MARKOV CHAIN: A mathematical formulation of a random walk, with the key property that the future of the walker only depends on its current state, and not it's past trajectory.

RAPID MIXING: A rapidly mixing Markov chain is a random walk that efficiently equilibrates to its stationary distribution.

ACKNOWLEDGMENTS

Firstly I wish to thank my advisor, Aram Harrow for his years of guidance, enthusiasm, and mentorship. Beginning when we first met, his interdisciplinary blend of physics, mathematics, and computer science has been a great influence on all my academic efforts. He also gave me the life-changing opportunity to be a long-term visiting student at MIT, where I found myself able to flourish in research and as a person more than ever before.

I also wish to thank Eddie Farhi, for teaching me to pay great attention to detail, and how persistent examination of detail can lead to new discoveries. It was at MIT that I learned the full value of collaboration in research, and I wish to thank all of my professors and classmates there for helping to shape my view of the research frontier in our field.

I'm very grateful to Dave Bacon for bringing me into the wonderful world of quantum computation, and to all of my "QW" classmates at the University of Washington for the many discussions we had while learning to orient ourselves in this field of research. I also wish to thank Marcel den Nijs in the physics department for his riveting courses on many-body physics, which have inspired my research directions ever since.

I am happy to thank the National Science Foundation, for their financial support which helped to make my graduate education possible. This work was supported in part by NSF grant CCF-1111382.

Finally, I wish to thank all my friends and family for the support they've given me over the years that has allowed me to pursue my academic ambitions. My greatest thanks goes to my Mother, Kate for inspiring my lifelong interest in reading that lead to my scientific curiosity, and for her lifetime of care and support that helped me through so many obstacles to achieve my dreams.

DEDICATION

to my Mother — look at how far we've come together!

Chapter 1

INTRODUCTION

1.1 The Physics of Spin Systems

Spin systems are a class of models in the study of many-body physics in which the degrees of freedom ("spins") are associated to the vertices V of a graph ("sites"). In classical systems the spin at each site takes values in a finite set (e.g. $S = \{-1, 1\}$), and the states of the system correspond to all allowed assignments of spin values to the vertices of the graph. In quantum systems each site is associated with a finite dimensional Hilbert space (e.g. a qubit $\mathbb{C}^2$), and the states of the system belong as usual to the tensor product of these finite dimensional spaces. These models can exhibit a variety of behaviors depending on the interactions between the spins at different sites, which are specified either by a classical energy function or a quantum Hamiltonian.

In this section we will define classical and quantum spin systems and review some of the main quantities of physical interest. Following this we review the subject of phase transitions, which provides an underlying foundation for understanding the results in each of the subsequent chapters. Chapter 2 introduces the thermal circuit model, which embeds a probabilistic computation into a classical spin system, and shows that the system can be made to compute correctly in thermal equilibrium below some constant threshold temperature. The stability of the computation being correctly performed across the spatial extent of the system, described in Section 2.5, can be viewed as a form of long-range order; this is a sign that the system has undergone a phase transition, since like all spin systems it is disordered at sufficiently high temperature. In chapter 3 we present different strategies for adiabatic optimization which improve the success probability of the algorithm by either taking advantage of the system's behavior at the quantum critical point at which the system

becomes gapless as in Section 3.2 and Section 3.3, or by changing the Hamiltonian path to remove the critical point as in Section 3.4. In chapters 4 and 5 we analyze the path-integral quantum Monte Carlo simulation method, which connects the equilibrium state of a quantum system to the non-equilibrium dynamics of a classical system. The question we address is whether the random walk underlying these non-equilibrium dynamics converges rapidly to the stationary distribution, or whether it undergoes a critical slowdown. In chapter 5 we prove the efficint convergence of this quantum Monte Carlo method for a class of 1D quantum systems at non-zero temperature, and the intuition for this result is supported by the fact that there are no thermal phase transitions for 1D spin systems with nearest neighbor interactions.

1.1.1 Classical Spin Systems

A classical spin system is specified by a state space $\Omega = S^V$ and a function $E : \Omega \to \mathbb{R}$ which assigns an energy $E(z)$ to each configuration $z \in \Omega$. Most attention is directed towards systems with local interactions, so that the energy function can be specified as a sum of local terms $E(z) = \sum_a E_a(z)$ where each E_a depends only on the spin value at a small number of sites. The most famous spin system is the ferromagnetic Ising model [Isi25],

$$E_{\text{Ising}}(z) = -\sum_{i \sim j} z_i z_j, \tag{1.1}$$

where $i \sim j$ means that the sites i and j are connected by an edge, and $z_i \in \{-1, 1\}$ is the spin value assigned to site i in the configuration $z \in \{-1, 1\}^V$. This model is 2-local, because each interaction term involves two spins (in general a system is k-local if every interaction term involves at most k spins). Additionally, if the interaction graph is a lattice in D spatial dimensions then the system is *spatially local*.

In a ferromagnetic Ising model the energy is decreased when adjacent vertices have the same spin value, and increased when the spins on neighboring sites differ. The ground state(s) of a spin system are defined as the minimum energy configurations; in the case of the ferromagnetic Ising model, there are two ground states, g_+ and g_-, which have the spin

on every site set to +1, or to -1, respectively. While the ground states of this model can be determined by inspection, we will see later that the general problem of determining the ground state(s) of local energy functions can be highly non-trivial.

When a spin system with energy function E is considered to be in thermal equilibrium at some non-zero temperature T, the properties of the system are determined by the Gibbs distribution,

$$\pi(z) = \frac{e^{-\beta E(z)}}{Z} \quad , \quad Z = \sum_{z \in \Omega} e^{-\beta E(z)}, \tag{1.2}$$

where $\beta = 1/T$ is the inverse temperature, and the normalizing constant Z is called the partition function. The Gibbs distribution specifies the probability $\pi(z)$ that the system is in the state z, and knowledge of the partition function can be use to determine thermal expectation values. For example the mean energy $\langle E \rangle$ is given by

$$\langle E \rangle = -\frac{\partial \log Z}{\partial \beta}. \tag{1.3}$$

An arbitrary observable A can be expressed as a derivative of the partition function by defining a modified energy function $E(\lambda) = E + \lambda A$ where λ is a real parameter, so that for the modified partition function $Z(\lambda)$ associated with $E(\lambda)$ we have

$$\langle A \rangle = -\frac{1}{\beta} \frac{\partial \log Z}{\partial \lambda}\Big|_{\lambda=0}. \tag{1.4}$$

The powerful relation (1.4) means that the thermodynamic average of any quantity can be computed from the partition function, so going from the energy function E of a system to its partition function Z can be considered as the central task in equilibrium statistical physics. An exact solution for the partition function allows for great insight into the properties of the system, and from the 1940s to the 1970s a beautiful series of results culminated in solutions for many translationally invariant systems on planar graphs [Bax82]. However, exact solutions are rare beyond translationally invariant systems in a low number of spatial dimensions. In chapter 2 we will describe a spin model, which is neither translationally invariant nor embeddable in low dimensional space, for which we were able to solve for the partition function exactly because of the relation of the model to a computational circuit.

1.1.2 Quantum Systems

Quantum spin systems place a finite dimensional Hilbert space $\mathbb{C}^d$ at each site of the graph, so that the state space of the system is a Hilbert space of dimension d^n,

$$\mathbf{\Omega} = \bigotimes_{v \in V} \mathbb{C}^d. \tag{1.5}$$

The parameter d is called the local dimension, and in the case $d = 2$ the quantum spins are called qubits. The interactions between the spins are specified by a Hamiltonian $\mathcal{H} : \mathbf{\Omega} \to \mathbf{\Omega}$, and just as in the classical case most attention is directed towards Hamiltonians with local interactions (which may or may not be spatially local). A 2-local Hamiltonian can be written in the form

$$\mathcal{H} = \sum_{i \sim j} \mathcal{H}_{ij} \tag{1.6}$$

where each $\mathcal{H}_{ij}$ only acts non-trivially on the spins at sites i and j, and acts as the identity on all other sites.

The quantum generalization of the ferromagnetic Ising model considered in the last section can be achieved by introducing a uniform transverse field,

$$\mathcal{H}_{\text{TIM}} = -\Gamma \sum_{v \in V} \sigma_i^x - \sum_{i \sim \text{J}} \sigma_i^z \sigma_j^z. \tag{1.7}$$

Here we will use the conventional "computational basis" for qubit states and operators,

$$\sigma^z|0\rangle = |0\rangle \quad , \quad \sigma^z|1\rangle = -|1\rangle \quad , \quad \sigma^x|+\rangle = |+\rangle \quad , \quad \sigma^x|-\rangle = -|-\rangle \tag{1.8}$$

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad , \quad \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1.9}$$

The ground state(s) of a quantum spin system are the eigenvectors of $\mathcal{H}$ corresponding to the smallest eigenvalue. When $\Gamma = 0$ the transverse Ising Hamiltonian is diagonal (in terms of the basis choice described above) and has essentially the same form as the classical Ising model from the previous section. All of the spins want to align, so the (unnormalized)

ground state is $|00...0\rangle + |11...1\rangle$. On the other hand, when $\Gamma \gg 1$ and transverse field term dominates, the spins are essentially uncoupled and the ground state is the uniform superposition of computational basis states,

$$|++...+\rangle = \sum_{z\in\{0,1\}^n} |z\rangle. \tag{1.10}$$

In the case of spatially local interactions on a 1D lattice, an exact solution for the entire eigenspectrum for any value of Γ can be found by a Jordan-Wigner transformation, but for general interaction graphs no solution to the ground state of this model is known at intermediate values of Γ. This contrasts with the classical case, where the ground state for ferromagnetic Ising models on general graphs can be determined by inspection.

In thermal equilibrium a quantum system with Hamiltonian $\mathcal{H}$ will be in the Gibbs state,

$$\rho = \frac{e^{-\beta\mathcal{H}}}{\mathcal{Z}} \quad , \quad \mathcal{Z} = \mathrm{tr}e^{-\beta\mathcal{H}}, \tag{1.11}$$

and the thermal expectation value of an observable A is given by $\langle A\rangle = \mathrm{tr}\,(\rho A)$.

As in the classical case, the solution of a physical model typically involves going from a Hamiltonian as in (1.6) to an explicit form of the ground state or thermal state that allows one to determine observables of interest.

1.2 The Computational Complexity of Spin Systems

The field of computational complexity formalizes the concept of computation, and examines how the resources needed to solve a particular computational task scale with increasing size of the problem. In spin systems the size of the problem is typically parameterized by the number of spin sites n, and in some cases by the number of local terms in the Hamiltonian or classical energy function. An algorithm that computes a physical property of the system, such as its partition function or the expectation value of an observable, is considered to be efficient if it runs in a time that scales polynomially with n. This is a stringent requirement, in part because the size of the state space of a classical spin system is exponential in n, and

in quantum systems the dimension of the Hilbert space is exponential in n! Despite this hurdle, efficient algorithms for computing physical properties are possible in many cases.

Formally, the run-time of an algorithm is defined in terms of the number of steps performed by a computational model called a Turing machine [Tur36]. The notion that the Turing machine model adequately captures the notion of what can be performed by any physical instantiation of a computer is called the Church-Turing thesis.

1.2.1 Turing Machines and Combinatorial Circuits

Deterministic Turing Machines

A deterministic Turing machine consists of the machine and a linear tape onto which elements from a finite alphabet, Γ, can be written into different cells on the tape. The machine itself has an finite set of internal states, Q, a method for reading a symbol written in a cell on the tape, a method to write a symbol in a cell on the tape, and the ability to move either left or right one cell. Formally the tape is assumed to be infinite or at least infinitely extendable. Initially an input, consisting of a string of symbols from a finite alphabet $\Sigma \subseteq \Gamma$ is placed on the tape and the physical machine is placed on the first symbol of this string. The machine is assumed to start itself with its internal states in a special state called the start state, $q_0 \in Q$, and then proceeds to "compute" as follows. At each time step the machine examines the symbol of the cell it currently occupies and its internal state. Conditional on these two variables the machine writes a symbol onto cell (possibly from a larger, but still finite, alphabet than the alphabet of the initial strings) it occupies, moves either left or right, and modifies its internal state. These rules can be specified by a function, the transition function, $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$, where L and R denote moving left or right. The machine does this indefinitely until it reaches one of a set of its internal states known as halt states. These halt states are labeled as either "reject" or "accept." If we initially input the string w and the machine halts in accept state, then we say the machine accepts w. Similarly if on input w the machine halts in a reject state, then we say that the machine rejects w.

Note that on some strings the Turing machine may neither accept or reject a string, but continue indefinitely. The language of a Turing machine is the set of all strings taken from the input alphabet for which the Turing machine accepts for that string being input.

Deterministic Turing machines capture many important features of real modern computers. For example, the running time of most algorithms on a variety of hardware are closely related to the similar problems cast in terms of the model described above. An important class of languages are those for which a deterministic Turing machine accepts a string from the language in a time bounded by a constant times a polynomial in the size of the string. Such languages are said to be in the complexity class P.

Non-Deterministic Turing machines

In order to classify the computational hardness of problems for which we we cannot yet conceive of any polynomial-time algorithms, it can be useful to consider models of computation with capabilities that go beyond deterministic Turing machines. A non-deterministic Turing machine is a model of computation which allows the Turing machine to explore multiple computational pathways at once, and halt whenever one of the states of any of the multiple branches it is exploring enters a halt state. In other words, a non-deterministic Turing machine doesn't have to chose one new state, new symbol and direction to move, but may branch to multiple such triples. The transition function is now not $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$, but instead is of the form $\delta : Q \times \Gamma \to P(Q \times \Gamma \times \{L, R\})$ where P denotes the power set. The time complexity of a non-deterministic Turing machine is the number of steps of executed by the Turing machine irrespective of the number of branches explored by the machine.

Because a non-deterministic Turing machine is allowed to explore multiple pathways in its computation it is not clear that such machines can be reasonably mapped (in terms of say, time complexity of problems) to modern computers. This question can be put on a more formal footing by defining the complexity class NP. NP is the class of languages which can be recognized by a non-deterministic Turing machine in a time bounded by a constant times

a polynomial in the size of the string being decided. The statement of whether NP is strictly larger than P is the famous P = NP problem, one of the most important open problems in the theory of computational complexity. That being said, most researchers do believe that P ≠ NP and that the model of a non-deterministic Turing machine is more powerful than computers we can physically construct.

Combinatorial Circuits

Another abstract model of computing is a circuit. Throughout the computation the information is encoded into a sequence of symbols from a finite alphabet which is called the state of the computation. The positions in the sequence are called *registers*. If desired one can pad the input with ancillary registers which can be thought of as "scratch space." The elementary operations, called *gates*, are taken to be functions which act on a small number of registers, and write their output to a small number of registers (e.g. the AND function which takes two bits as input and outputs a 1 if both bits are 1, and otherwise outputs 0). The gates are applied in sequence to the input (and ancillas), and the number of gates is called the size of the circuit, which measures the resource cost of the computation.

More formally, a *combinatorial circuit*, $C = (G, L)$, is a directed acyclic graph, $G = (V, E)$ with vertices V and edge set E, where the internal vertices, V_{in}, of the graph are logical gates, $L : V_{in} \to \mathcal{G}$ ($\mathcal{G}$ is the set of all logical gates), and the external vertices, V_{ex}, are the inputs and outputs to the circuit. External vertices that have no edges leading to them are inputs and external vertices that have no edges that lead away from them our outputs to the circuit. If a circuit has n input vertices and m output vertices, then to such a circuit we can assign a function $f : \{0,1\}^n \to \{0,1\}^m$ which results from propagating the input through the logic gates to the output vertices. Note that at this point we require that fan-outs in our circuit are represented by gates and do not allow circuits which have fan-in gates. A k-fan-out gate is the boolean function $f_k : \{0,1\} \to \{0,1\}^k$ given by $f_k(x) = (x, \ldots, x)$.

1.2.2 Complexity in Classical Spin Systems

While the ground state of a ferromagnetic Ising model can be easily determined by inspection, one of the first results formalizing the hardness of finding the ground state(s) of classical spin systems was provided by Barahona [Bar82]. There it was shown that any instances of MAX-CUT could be reduced to finding the ground state of a 3D Ising model with $\{+1, -1\}$ nearest-neighbor couplings, and therefore it is NP-hard to find such ground states. This result also holds for 2D Ising models with $\{+1, -1\}$ nearest-neighbor couplings and $\{+1, -1\}$ local fields. From a physical point of view the hardness of finding the ground state of these systems is due to a large degeneracy of low energy configurations, which causes the system to form a glassy phase at low temperature.

Another milestone in the computational complexity of spin systems occured when Jerrum and Sinclair [JS93] gave a polynomial-time algorithm for approximating the partition function of ferromagnetic Ising models, at any temperature $1/\text{poly}(n) < T < \text{poly}(n)$ and on arbitrary graphs. Their algorithm is based on a rapidly mixing Markov Chain which performs non-local updates on configurations that correspond to terms in the high-temperature expansion of the partition function. However, in the same work these authors also show that approximating the partition function of an Ising model on an arbitrary graph with $\{+1, -1\}$ nearest-neighbor couplings is a very hard problem, as hard as counting the number of solutions to an instance of 3-SAT (which is clearly harder than the NP-complete problem of deciding whether a 3-SAT instance has *a* solution or not).

1.2.3 Complexity in Quantum Spin Systems

While classical computing technology improved at a great pace in the 20^{th} century, the task of simulating quantum systems (beyond special cases and small system sizes) remained apparently difficult for classical computers, in direct part because the size of the classical memory needed to store an arbitrary state of a system of n quantum spins grows exponentially in n. This motivated Feynman to propose the idea of simulating quantum systems by using

computers which exhibit quantum effects [Kis85], since (at least in principle) a quantum computer could overcome this memory problem by encoding the quantum state of a physical system into a computer consisting of quantum bits.

While quantum computers allow for efficient algorithms for quantum simulation, and were later found to efficiently perform other tasks such as integer factoring [Sho94] which are thought to be classically hard, their computational power is not unlimited. Just as NP was defined in order to classify problems which are hard classically, Kitaev defined a quantum version of NP ([KSV02], [AN02]), which today is called QMA, to classify problems which are as hard as performing nondeterministic quantum computations - and therefore thought to be hard even for real-world quantum computers. Just as the boolean satisfaction problem SAT is an important NP-complete problem, Kitaev demonstrated that a Hamiltonian satisfaction problem, called the local Hamiltonian problem, is QMA-complete.

The essence of the local Hamiltonian problem is to decide whether the ground state energy of a local Hamiltonian is above or below some threshold value. It may not be surprising that this problem can be very hard for Hamiltonians which are gapless and/or frustrated, but Kitaev was able to formalize this hardness by showing that a particular Hamiltonian – called the Feynman-Kitaev Hamiltonian – could have as its ground state a superposition of the time steps of an arbitrary quantum computation. Therefore the problem of estimating the ground state energy of this Hamiltonian is linked to the question of whether the associated quantum circuit has an accepting input, making the problem as hard as determining whether a nondeterministic quantum computer will accept or reject it's input.

In 2006, Bravyi et al [BDOT08] brought attention to a special subset of Hamiltonians which are called "stoquastic", by showing that the stoquastic local Hamiltonian problem is in the classical complexity class MA, and is therefore not thought to be as hard as QMA as in the case of general Hamiltonians. Stoquastic Hamiltonians are those which Suzuki [SMK77] recognized as being free of the sign problem, and are therefore amenable to quantum Monte Carlo methods. These Hamiltonians are conjectured to allow for polynomial-time classical simulations in many cases, but to date only a few results along these lines are known; simu-

lating adiabatic evolutions with frustration-free stoquastic Hamiltonians, and approximating the partition function for ferromagnetic transverse Ising models (which are stoquastic) on arbitrary graphs. It has also been shown that any instance of the stoquastic local Hamiltonian problem can be mapped to a local Hamiltonian problem for a generalized transverse Ising model [BH14]. In chapter 5 we give a polynomial-time algorithm for approximating the partition function for any 1D generalized transverse Ising model, at temperatures which are independent of the system size.

1.3 Physics and Optimization

Optimization is the task of finding the "best" element of a set, according to some criteria. A standard form for the optimization problems we will consider is as follows: given a real-valued function $f : \Omega \to \mathbb{R}$ on a domain Ω, find the element $x^* \in \Omega$ that achieves the minimum value $f_{\min} = \min_{x \in \Omega} f(x)$. The function f is sometimes called a *cost function*, and the element x^* is called the *global minimum* of f. Note that minimizing f corresponds to maximizing $-f$, so this form of the problem accommodates maximization problems as well. In general the domain Ω does not need to be finite (or even discrete), but in this work we will examine cases where Ω is a finite set.

There is a longstanding and fruitful correspondence between computational optimization problems and finding the ground states which minimize the energy of quantum and classical physical systems. If we associate Ω with the state space of a classical spin system and view the cost function f as a classical energy function, then the problem of minimizing f is recast into physical language as the problem of finding the ground state of a spin system. In this section we will review two highly influential physics-inspired algorithms for the abstract mathematical task of optimization: the classical simulated annealing algorithm [KGV83] which emulates the process of cooling a system down into thermal equilibrium at low temperatures, and the quantum algorithm known as either quantum adiabatic optimization [FGGS00] (when it is effectively run at zero temperature) or as quantum annealing [KN98](when non-zero temperature effects are important).

1.3.1 Simulated Annealing

Physical systems in thermal contact with an environment will eventually equilibrate to states which minimizes their free energy, $F = E - TS$, but sometimes a system can be trapped in a quasi-equilibrium (a local minimum of the free energy) which can make the process of arriving at the true equilibrium prohibitively slow. For example, defects in the lattice structure of a metal material correspond to quasi-equilibria, and these relax very slowly at room temperature. Annealing is a method, discovered by ancient metallurgists millenia ago, for accelerating this equilibration by first heating the material up to a high temperature and then slowly cooling it down to a state that has a smaller free energy at lower temperatures.

Simulated annealing attempts to minimize a cost function by treating it as the energy of a physical system, and by performing a random walk to sample the Gibbs distribution of this energy function at various temperature[KGV83]. The Gibbs distribution for a cost function f is

$$\pi(z) = \frac{e^{-\beta f(z)}}{Z} \quad , \quad Z = \sum_{z \in \Omega} e^{-\beta f(z)}. \tag{1.12}$$

To simulate the processes that drive a physical system to equilibrium when the temperature is changed, the proposal is to perform a random walk on the state space Ω. The transition probability $P(i, j)$ to move between states $i, j \in \Omega$ is chosen to satisfy detailed balance,

$$\pi(i)P(i, j) = \pi(j)P(j, i),$$

which ensures that the random walk will eventually converge to π. The algorithm proceeds by initializing the system to a random state at high temperature β_0, and simulating the random walk while also cooling the system through a schedule of lower temperatures $\beta_0 < \beta_1 < ... < \beta_k$. While it is difficult to predict in advance the number of steps of the random walk which will be needed to converge, the algorithm does well at producing low energy solutions in practice and is widely used in applications.

1.3.2 Quantum Annealing

The Quantum Adiabatic Algorithm (QAA) can be used on a quantum computer as an optimization method [FGGS00] for finding the global minimum of a classical cost function $f : \{0,1\}^n \to \mathbb{R}$. The cost function is encoded in a problem Hamiltonian H_P which acts on the Hilbert space of n spin-$\frac{1}{2}$ particles,

$$H_P = \sum_{z \in \{0,1\}^n} f(z)|z\rangle\langle z|. \tag{1.13}$$

The Hamiltonian H_P is diagonal in the computational basis, and its ground state corresponds to the bit string that minimizes f. To reach the ground state of H_P the system is first initialized to be in the ground state of a beginning Hamiltonian, which is traditionally taken to be

$$H_B = \sum_{i=1}^{n} \left(\frac{1 - \sigma_x^i}{2} \right). \tag{1.14}$$

The ground state of H_B, which can be prepared efficiently, is the uniform superposition of computational basis states

$$|\psi_{init}\rangle = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle. \tag{1.15}$$

The system is then acted upon by the time-dependent Hamiltonian

$$H(t) = (1 - \frac{t}{T})H_B + \frac{t}{T}H_P \tag{1.16}$$

from time $t = 0$ to $t = T$ according to the Schrodinger equation

$$i\frac{\mathrm{d}}{\mathrm{dt}}|\psi(t)\rangle \ = H(t)|\psi(t)\rangle. \tag{1.17}$$

For a problem instance with a unique string w that minimizes f, the probability of obtaining w at time $t = T$,

$$P(T) = |\langle w|\psi(t = T)\rangle|^2, \tag{1.18}$$

is a metric for the success of the method on that particular instance.

By the adiabatic theorem, if we prepare the system initially in the ground state of H_B and evolve for a sufficiently long time T, then the state of the system at the end of the

evolution will have a large overlap with the ground state of H_P. Specifically, the adiabatic approximation requires $T > \mathcal{O}(g_{min}^{-2})$, where g_{min} is the minimum difference between the ground state energy and the first excited state energy during the course of the evolution.

1.4 Quantum Monte Carlo Simulations

The possibility of classically simulating thermal states of a limited sub-class of quantum systems using Quantum Monte Carlo methods was first proposed by Suzuki, Miyashita and Kuroda almost 40 years ago [SMK77]. The key limitation of their method is that the Hamiltonian must have all real nonpositive off-diagonal matrix elements in a particular basis, which has been called being "free of the sign problem." In mathematics these matrices are called sub-stochastic [Hor86] for their connection with stochastic matrices, and are also known as Perron-Frobenius matrices, after the authors who recognized some of their special properties over 100 years ago [Per07, Fro12]. In condensed matter physics, such Hamiltonians have been described as having a "stochastic matrix form"[CCMP05], and are sometimes known as Rokhsar-Kivelson Hamiltonians after authors who applied the special properties of these matrices to understand the square lattice quantum dimer model [RK88]. The term "stoquastic" is a portmanteau of "quantum" and "stochastic" which first appeared in [BDOT08], and since then these Hamiltonians have come to attention as a sub-class of quantum systems which may allow for provably efficient classical simulations.

Despite the seemingly restrictive definition, the class of stoquastic Hamiltonians includes many quantum systems of physical and computational interest. A system of spinless particles moving on a graph, with position dependent interactions, is described by a Hamiltonian which is stoquastic with respect to the position basis. Transverse Ising models with arbitrary diagonal interactions are stoquastic in the computational basis (here we take the transverse field to be uniform and pointing in the $+X$ direction, if it is pointing in the $-X$ direction the system can be made stoquastic by a simple change in basis), and this example is particularly notable because of quantum annealing proposals involving thermal states of these Hamiltonians. Many other quantum spin models are stoquastic (such as the ferro-

magnetic Heisenberg model) or can be made stoquastic by a change of basis (such as the anti-ferromagnetic Heisenberg model on bipartite graphs).

If the matrix entries of H are real and non-positive (in a particular basis), then the matrix entries of the thermal density matrix ρ are real and non-negative (in that same basis). The key property of non-negative matrices which was realized by Perron and Frobenius is that that all of the coefficients of the eigenvector corresponding to the largest eigenvalue of a non-negative matrix have the same phase. Since the largest eigenvalue of ρ corresponds to the ground state of H, the Perron-Frobenius theorem implies that there is a choice of phase for which the ground state amplitudes are non-negative in the computational basis. This opens up the possibility of treating the ground state wave function as an (unnormalized) classical probability distribution, and the idea behind the quantum Monte Carlo method we will consider is to treat the off-diagonal matrix entries of a stoquastic Hamiltonian as transition probabilities of a random walk that converges to the thermal state. In chapters 4 and 5 we analyze a particular version of QMC, called path-integral quantum Monte Carlo (PI-QMC), which uses a random walk on a state space which consists of all possible worldlines of the quantum system.

Although the PI-QMC method had been well-known for decades and is widely used in practice, there are still very few rigorous guarantees on the convergence time of the underlying random walk. In chapter 5 we give a formal analysis of the convergence of the PI-QMC method when it is applied to 1D generalized transverse Ising models. To do this we will review some definitions and techniques from the mathematical theory of Markov Chains in section Section 5.3.4. The time it takes for a Markov chain to converge to its stationary distribution is governed by the spectral gap of the transition matrix (which encodes the transition probabilities of the random walk). The standard belief in computational physics is that the spectral gap of the QMC transition matrix should be related to the spectral gap of the Hamiltonian, and that the convergence time of the QMC method scales with the correlation length in the quantum system, but a rigorous proof of this conjecture has remained elusive. In [HF13] it was shown that there are pathological examples of Hamiltonians which

are gapped, and which have a unique ground state, for which the convergence time of PI-QMC is exponentially slow. Besides the need to understand how pervasive these counterexamples are in the quantum world, we also take the view that provably efficient simulation algorithms are valuable for the perspective they yield on the complexity of these systems.

Chapter 2

UNIVERSAL CLASSICAL COMPUTATION IN THERMAL EQUILIBRIUM

In this chapter, which is based on [CBB10], we begin with a model of classical deterministic computing in which the ground state of the classical system is a spatial history of the computation. In its most primitive form, systems constructed in this model cannot compute in an error free manner when working at non-zero temperature. However, by exploiting a mapping between the partition function for this model and probabilistic classical circuits we are able to show that it is possible to make this model effectively error free. We achieve this by using techniques in fault-tolerant classical computing and the result is that the system can compute effectively error free if the temperature is below a critical temperature. We further link this model to computational complexity and show that a certain problem concerning the thermal equilibrium distribution of classical spin systems is complete for the complexity class MA.

Finally, our mapping leads naturally to problems which are complete for the complexity class promise Merlin-Arthur [Bab85, GS86], a result which can be regarded as a finite temperature version of the Cook-Levin theorem [Coo71, Tra84] from the theory of computational complexity. The new computational model we consider thus serves as a bridge between statistical mechanics and the classical computational complexity of probabilistic computation. We also obtain a computational complexity result concerning the difficulty of identifying when a system admits a mapping to a probabilistic circuit by showing that a restricted version of this problem is NP-complete. This implies that the problem of identifying when a physical system can be seen to be performing a computation is itself computationally intractable.

The outline of the chapter is as follows. In Section 2.1 we introduce the model of com-

puting with spins in the ground state using a particular energy function for this system. In Section 2.2 we show that at non-zero temperature the model from the previous section fails to properly compute. We do this for an extremely simple and previously explored model, but present the result using two different methods, one related to reinterpreting the transfer matrix and the other related to classical circuits applied to simple thermal ensembles. Generalizing these methods allows us to discuss the energy functions arising in Section 2.1 as probabilistic circuits. This more general formulation we discuss using two different methods in Section 2.3. Motivated by the mappings discovered in Section 2.3, we then return to our original model and define the broader class of energy functions consistent with these models in Section 2.4. At this point we also point out how our models differ from classical models of quantum-dot cellular automata. The more general setting leads us to ask questions about how algorithmically hard it is to decide if a given physical system supports computation in these models. We show that this problem, even in a very weak form, is NP-complete and thus likely to be intractable. In Section 2.5 we discuss how to design ground state spin models that, unlike the generic case, do compute fault-tolerantly at non-zero temperature. Our construction is intimately related to the original fault-tolerance construction of von Neumann [von56] and we show that von Neumann's threshold for fault-tolerance is related to a critical temperature in our system. Finally in Section 2.6 we discuss how the model we consider is related to the Cook-Levin theorem from computational complexity and show how this leads to certain natural problems about our model being **Promise-**MA-complete.

2.1 Deterministic classical computing in ground states

Here we introduce the model of classical ground state spin computing. This model, in particular restricted cases, is relevant to a variety of different models considered elsewhere. For instance, quantum dot cellular automata can be partially modeled by classical ground state spin computing [LTPB93, LTP94]. However the model we consider is more general than these specific instances. This larger generality comes along with certain unphysical assumptions: our models contain, for example, three-body interactions which are not nec-

essarily easily achievable in a physical device. Physically we imagine that models such as the one we consider might emerge in certain limits where effective many-body interactions can emerge. However, irrespective of the physical implementation, the model provides a set of classical many-body interacting systems which can be mapped onto probabilistic classical circuits. This connects statistical mechanics with classical probabilistic circuits thus bridging computer science and physics in a new and interesting manner.

We further note that a quantum version similar to this model has been studied by Mizel *et al* [MMC02, Miz04]. This later model contains significant difference owing to the quantum nature of the computation: in particular the ground state does not contain the computation laid out spatially, but instead exists in a superposition of the computation being carried out spatially. These quantum models are also connected to universal adiabatic quantum computing schemes [KR06, OT08, BL08] and piecewise variations on these schemes [BF08, BF10, OBL09, Ore09]. By studying the classical analogy of these schemes, we hope to shed light on how these later quantum methods can be made fault-tolerant.

2.1.1 The Model

Consider the following physical system. For each edge of a combinatorial circuit associate a single subsystem with only two possible states, 0 and 1, i.e. associate a bit to every edge. Consider next a logic gate which has incoming edges labeled by bits $i_1, i_2, \ldots, i_j$ and outgoing edges labeled by $t_1, t_2, \ldots, t_k$. For such a logic gate, l, define the following energy function

$$E_l(i_1, i_2, \ldots, i_j, t_1, t_2, \ldots, t_k) = \begin{cases} 0 & \text{if } f_l(i_1, i_2, \ldots, i_j) = (t_1, t_2, \ldots, t_k) \\ \Delta & \text{otherwise} \end{cases}, \tag{2.1}$$

where f_l is the function the logic gate is computing. Note that this energy contributes 0 when the inputs and outputs follow the rules of the logic gate, but is Δ otherwise. Now define the energy for a particular configuration of the bits associated with edges as the sum over all logic gates of these energy terms:

$$E_C(s) = \sum_{l \in L} E_l \tag{2.2}$$

where each E_l acts on the appropriate input and output bits and s is the collection of bits for the entire system. This energy is a function of the labels on all of the edges. Its zero energy configurations are ground states which consist of configurations which correctly compute the logical function corresponding to the circuit C. Note that at this point the ground state is degenerate: every valid input (and corresponding output) defines a valid zero energy configuration. We call an energy function constructed in this fashion from a circuit a *circuit energy function.*

In addition to imposing energy constraints to enforce logic gates, we may also impose energy constraints to fix a particular input to the circuit. Suppose that the input vertices are $v_1, v_2, \ldots, v_n$ and we wish to force the input $x_1, x_2, \ldots, x_n$, where $x_i \in \{0, 1\}$. Then to each edge whose bit is labeled s_i lead away from vertex v_i we can associate an energy term $E(s_i) = 0$ if $s_i = x_i$ and $E(s_i) = \Delta$ otherwise. Taking a sum over all of these terms for the input vertices (and corresponding edges) we can thus add a term such that the ground state configuration consists now only of the input $x_1, \ldots, x_n$ propagated to the output of the circuit, $f(x_1, \ldots, x_n)$. In particular for input $x \in \{0, 1\}^n$ define the

$$E_{C,x}(s) = E_C(s) + \sum_{i=1}^{n} E_{x_i}(s_i) \equiv E_C(s) + E_x(s), \tag{2.3}$$

where

$$E_{x_i}(s_i) = \begin{cases} 0 \text{ if } s_i = x_i \text{ where } s_i \text{ corresponds to input } v_i \\ \Delta \text{ if } s_i \neq x_i \text{ where } s_i \text{ corresponds to input } v_i \end{cases} . \tag{2.4}$$

We call an energy function made up of a circuit energy function plus an input forcing energy function a *computed circuit energy function.*

The above description of how to take a combinatorial circuit, C, plus its input, $(x_1, \ldots, x_n)$, and converting it into a many-spin energy function is what we term the *classical ground state spin computing* (CGSSC) model. Just as in quantum dot cellular automata [LTPB93, LTP94], the computation occurs when the system is in its ground state and is spatially spread out across the device. Unlike in quantum dot cellular automata, however, this model can implement logic gates by using interactions beyond just pairwise interacting spins (or

pseudo-spins in the case of the quantum dot configurations.) For example, constructing a quantum wire in the CGSSC model will correspond directly to an identical model in the semi-classical limit of quantum dot cellular automata models. Gates in quantum dot cellular automata, however, are implemented in a way which is not directly analogous to the CGSSC model. In section 2.4 we return to this issue and define a more general set of energy functions wherein our results still hold and compare this with quantum cellular automata models.

2.1.2 Example

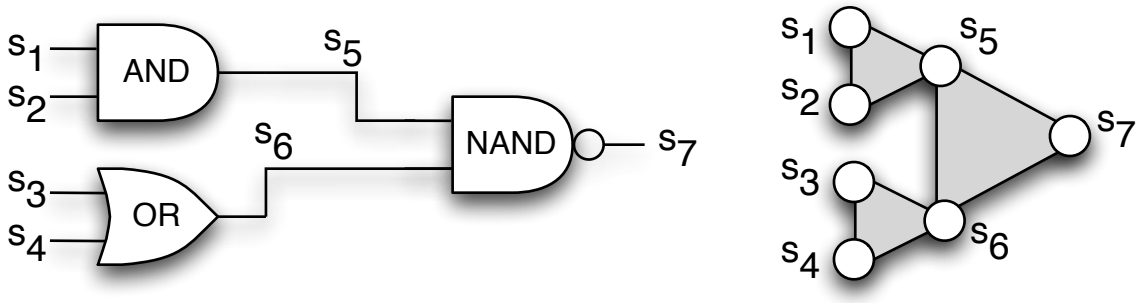


Figure 2.1: Left: An example circuit whose circuit energy function is given in the text. The bits of the ground state spin energy function are labeled s_i. Right: the spin system corresponding to this circuit. Here the spins connected by a triangles have interactions between them as specified in the main text.

To be explicit, let's consider an example circuit involving four inputs, a few logical gates, and one output as diagramed in Fig. 2.1. The energy function for the AND gate is then, for

example,

$$\begin{aligned} E_{AND}(s_1, s_2, s_3, s_4, s_5) &= \Delta((1-s_1)(1-s_2)s_5 + (1-s_1)s_2s_5 + s_1(1-s_2)s_5 + s_1s_2(1-s_5)) \\ &= \Delta(s_5 + s_1s_2 - 2s_1s_2s_5). \end{aligned} \tag{2.5}$$

The full circuit energy function is given by

$$\begin{aligned} E_C(s_1, s_2, s_3, s_4, s_5) &= \overbrace{\Delta(s_3 + s_4 + s_6 - s_3s_4 - 2s_3s_6 - 2s_4s_6 + 2s_3s_4s_6)}^{\text{OR}} \\ &\quad + \underbrace{\Delta(s_5 + s_1s_2 - 2s_1s_2s_5)}_{\text{AND}} + \underbrace{\Delta(1 - s_7 - s_5s_6 + 2s_5s_6s_7)}_{\text{NAND}}. \end{aligned} \tag{2.6}$$

Suppose that we wish to force the input to be $s_1 = s_2 = s_3 = 0$ and $s_4 = 1$. Then we would add the term

$$E_{C,x} = E_C + \Delta(s_1 + s_2 + s_3 + (1 - s_4)). \tag{2.7}$$

2.2 The need for fault-tolerance at non-zero temperature

Having defined a spin model whose ground state deterministically computes a circuit, we can now consider the physically important question of how the model behaves in thermal equilibrium at non-zero temperature. In order to preserve the intended computation the conditional probability of output $f(i_1, i_2, \ldots, i_n) = (t_1, t_2, \ldots, t_m)$, given the forced input $(i_1, i_2, \ldots, i_n)$, in the Gibbs distribution for this circuit energy function must be large enough to distinguish this output from a completely random outcome. It is easy to see that for at least some circuits C and inputs x, ground state computation will fail to correctly evaluate the function corresponding to the circuit as the size of the circuit being implemented grows. This follows directly from examining one of the most basic models in statistical physics, the one-dimensional Ising model [Isi25]. This was pointed out in the context of quantum-dot cellular automata in [LTP94] (see also [UFMI00, WL04]). Here we reproduce this argument presenting it in two different forms. We do this not just to be pedagogical, but also because these two methods will generalize from the one dimensional case to more general circuit computing energy functions.

Consider the circuit computing energy function corresponding to inputing the bit 0 into a series of n identity gates:

$$E_I(s_1, \ldots, s_{n+1}) = \Delta\delta_{s_1,1} + \Delta \sum_{i=1}^{n} \delta_{s_i,\neg s_{i+1}}, \tag{2.8}$$

where $\neg s$ represents the negation of s: $\neg 0 = 1$ and $\neg 1 = 0$. It is convenient here, and in the sequel, to work with $\{+1, -1\}$ valued variables instead of the $\{0, 1\}$ valued s_i's. Thus we will define uppercase spin variables to be ± 1 valued versions of the s_i's via $S_i = 1 - 2s_i$. Then the above energy function can be written as

$$E_I(S_1, \ldots, S_{n+1}) = \frac{\Delta}{2}(1 - S_1) + \frac{\Delta}{2} \sum_{i=1}^{n} (1 - S_i S_{i+1}), \tag{2.9}$$

which we recognize as the one dimensional Ising model with ferromagnetic couplings and a boundary term which is a local field. The ground state of this is simply all $s_i = 0$ ($S_i = +1$), i.e. the initial 0 has been copied by identity gates down the line.

The thermal ensemble arising from this energy function is given by

$$Pr(S_1, \ldots, S_{n+1}) = \frac{\exp(-\beta E_I(S))}{Z}, \tag{2.10}$$

where Z is the partition function,

$$Z = \sum_{S \in \{+1,-1\}^{n+1}} \exp(-\beta E_l(S)), \tag{2.11}$$

and $\beta = (k_B T)^{-1}$ is the inverse temperature. It is well known that the one dimensional Ising model does not order at finite temperature in the thermodynamic limit [Isi25]. From our perspective, we observe that the system will fail to correctly transmit the 0 down the line at finite temperature except in a window of size $k_B T < \frac{\Delta}{n}$ which goes to zero as the system size goes to infinity.

2.2.1 Transfer Matrix Reinterpreted as a Probabilistic Circuit

The standard method for solving this model is the transfer matrix method which we review and then reinterpret. To the energy function add a term dependent on a variable γ for the

last spin:

$$E'_I(S,\gamma) = E_I(S) + \gamma S_{n+1}. \tag{2.12}$$

If we calculate Z' for this energy function, we can then use this to calculate the probability that S_{n+1} is $+1$ via $Pr(S_{n+1} = +1) = \langle \frac{1}{2}(1 + S_{n+1})\rangle = \frac{1}{2}(1 + \langle S_{n+1}\rangle)$ where

$$\langle S_{n+1}\rangle = -\frac{1}{\beta}\frac{\partial \ln Z'}{\partial \gamma}\bigg|_{\gamma=0} = -\frac{1}{\beta Z'}\frac{\partial Z'}{\partial \gamma}\bigg|_{\gamma=0}. \tag{2.13}$$

Define the two by two matrices

$$[M_i]_{S_{i+1},S_i} = \exp\left[-\frac{b}{2}(1 - S_i S_{i+1})\right], \tag{2.14}$$

where $b = \beta\Delta$, as well as the column vector

$$[v]_{S_1} = \exp\left[-\frac{b}{2}(1 - S_1)\right], \tag{2.15}$$

and the row vector,

$$[w^T]_{S_{n+1}} = \exp[-\beta\gamma S_{n+1}]. \tag{2.16}$$

Explicitly the partition function Z' is

$$Z' = \sum_{S\in\{+1,-1\}^{n+1}} \exp\left[-\beta\gamma S_{n+1} - \frac{b}{2}\left(1 - S_1 + \sum_{i=1}^{n}(1 - S_i S_{i+1})\right)\right]. \tag{2.17}$$

This can then be written as

$$Z' = \sum_{S\in\{+1,-1\}^{n+1}} [w^T]_{S_{n+1}} \prod_{i=n}^{1} [M_i]_{S_{i+1},S_i} v_{S_1} = w^T M_n M_{n-1} \ldots M_1 v, \tag{2.18}$$

where the last equation is a row vector, a matrix product, and a column vector thus giving a scalar. This is the transfer matrix form of the solution: the partition function is written as a product of "transfer matrices" and, in this case, sandwiched between an "initial" and "final" vector.

Define the following matrices and vectors:

$$[P_i]_{S_{i+1},S_i} = \frac{[M_i]_{S_{i+1},S_i}}{1 + \exp(-b)} \tag{2.19}$$

$$[p_{in}]_{S_1} = \frac{[v]_{S_1}}{1 + \exp(-b)}. \tag{2.20}$$

We can rewrite Z' as

$$Z' = (1 - \exp(-b))^{-(n+1)} w^T P_n P_{n-1} \dots P_1 p_{in}. \tag{2.21}$$

It is at this point that we begin to see our *reinterpretation* of the transfer matrix method. In particular we note that p_{in} is a vector of probabilities (that is it sums to unity) and P_i is a stochastic matrix (its columns sum to unity). In other words encoded into Z' is a classical preparation of a probabilistic bit, followed by an evolution according to probabilistic gates. Suppose that we let $p_{out} = P_n P_{n-1} \dots P_1 p_{in}$ denote the output of this probabilistic circuit. Then

$$Z' = (1 - \exp(-b))^{n-1} \langle w \rangle_{p_{out}} \tag{2.22}$$

where the sample is taken from the output of the probabilistic circuit. Notice that at $\gamma = 0$, w is a vector made up of all components equaling 1. In this case, $\langle w \rangle_{p_{out}}|_{\gamma=0} = 1$ since then this term is a sum over all the outputs to the probabilistic circuit. Further

$$-\frac{1}{\beta} \frac{\partial Z'}{\partial \gamma}\bigg|_{\gamma=0} = (1 - \exp(-b))^{n+1} \langle \sigma \rangle_{p_{out}} \tag{2.23}$$

where $\sigma = (+1, -1)^T$. Thus

$$\langle S_{n+1} \rangle = \langle \sigma \rangle_{p_{out}}. \tag{2.24}$$

In other words the expectation value of the last spin in the chain is equal to the expectation value for the output of the circuit starting with p_{in} and then applying the probabilistic gates $P_1, \dots, P_n$. Note that the combinatorial factors $(1 + \exp(-b))^{n+1}$ have canceled out.

The above description tells us that we can think about the one dimensional Ising spin chain with a forced boundary term as a probabilistic circuit related to our ground state spin computation. In particular the probabilistic circuit starts with the possibility of an incorrectly initialized boundary. This is formulated in the preparation probability vector p_{in}. Then with probability $\frac{\exp(-b)}{1-\exp(-b)}$ the ground state spin computation of an identity gate is replaced by a bit flip gate. The probability of the final spin of the chain being in a particular state is then directly given by the probability that the probabilistic circuit outputs a particular value.

In this model the probability of the ground state properly computing the desired identity circuit is such that the information is washed out at finite temperature. To see this we must actually calculate $\langle S_{n+1} \rangle$. This can be done most easily by diagonalizing the gates. In particular if we define the Hadamard matrix,

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.25}$$

then

$$P_i = HDH^{-1} = HDH \tag{2.26}$$

where

$$D = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1-\exp(-b)}{1+\exp(-b)} \end{bmatrix}. \tag{2.27}$$

Thus

$$\langle S_{n+1} \rangle = \sigma^T H D^n H p_{in} \tag{2.28}$$

which can be reduced to

$$\langle S_{n+1} \rangle = \left(\frac{1 - e^{-b}}{1 + e^{-b}} \right)^{n+1} = \left[\tanh \frac{b}{2} \right]^{n+1}. \tag{2.29}$$

From this expression we see that the probability that the final spin is aligned with the input quickly drops to 1/2 as a function of n (unless $b > n$, i.e. in a small window of temperature below $\frac{\Delta}{n}$.) Thus the circuit fails with high probability as we make the chain longer and longer. This is exactly the result of [WL04]. In this respect, ground state spin computing in this model is not tolerant to errors arising from working at finite temperature. Note that we do not consider scaling of the temperature to stay within the small window where the computation is correctly performed to be a fault-tolerant method as it requires unreasonable physical resources as the system size gets larger.

2.2.2 A Second Approach Using Controlled-Not Gates

Having shown in the previous subsection how one can reinterpret the transfer matrix method as a probabilistic circuit for the one dimensional Ising model with forced boundary, we next

present a second way to derive this observation. This method is a direct variation on the method used in [Egg74] to study Ising models on Cayley trees.

On the $n+1$ spins, define the controlled-not on positions i and $i+1$ as the function, C_i, which maps $\{+1,1\}^{n+1}$ to $\{+1,-1\}^{n+1}$ via

$$C_i(S_1,\ldots,S_i,S_{i+1},S_{i+2}\ldots,S_n) = (S_1,\ldots,S_i,S_{i+1}S_i,S_{i+2},\ldots,S_n). \tag{2.30}$$

We call this a controlled-not because it functions as a deterministic gate which, controlled on the spin S_i either does nothing to the spin in position $(i+1)$ (if $S_i = +1$), or flips the spin in position $(i+1)$ (if $S_i = -1$.) Further define the function which is the composition of controlled-not's starting at the first spin and working forward (note that $C_i \circ C_{i+1} \neq C_{i+1} \circ C_i$):

$$C = C_n \circ C_{n-1} \circ \cdots \circ C_2 \circ C_1. \tag{2.31}$$

Notice that if we start with spin 1 in the state S_1 and all of the other spins $S_2,\ldots,S_{n+1}$ in $+1$, then the action of C is to *copy* S_1 down the line: $C(S_1,+1,\ldots,+1) = (S_1,S_1,\ldots,S_1)$. In this sense C is the operation of applying identity gates as information is propagated down the spin chain. Note that the C_i are bijections as is C and that C_i is its own inverse. Thus

$$C^{-1} = C_1 \circ C_2 \circ \cdots \circ C_{n-1} \circ C_n. \tag{2.32}$$

Explicitly,

$$C(S_1,S_2,S_3,\ldots,S_{n+1}) = (S_1,S_1S_2,S_1S_2S_3,\ldots,\prod_{i=1}^{n+1} S_i) \tag{2.33}$$

and

$$C^{-1}(S_1,S_2,S_3,\ldots,S_{n+1}) = (S_1,S_1S_2,S_2S_3,\ldots,S_nS_{n+1}). \tag{2.34}$$

Consider the energy function on $n+1$ spins:

$$E'(S'_1,S'_2,\ldots,S'_{n+1}) = \frac{\Delta}{2}\sum_{i=1}^{n+1}(1-S'_i). \tag{2.35}$$

Then clearly a system in the thermal state described by this energy function has each spin in $+1$ with probability $1-p = 1/(1+\exp(-b))$ and -1 with probability $p = \exp(-b)/(1+$

$\exp(-b)$). Suppose that we apply the C function to this system. That is consider the above system in its thermal ensemble and consider physically applying the controlled-not gates. Then certainly this can be thought of as a system in which the first bit is prepared in 0 with probability $1-p$, with probability $1-p$ the controled-nots successfully copy the bit down the line, and with probability p the output of the controlled-not is flipped. This is exactly the probabilistic circuit as described in the previous section.

How is this related to our original E? Let the unprimed variables be the spins after C has been applied to the system. Then

$$Pr(S_1,\ldots,S_{n+1}) = \sum_{S'_1,\ldots,S'_{n+1}\in\{\pm 1\}} Pr(S_1,\ldots,S_{n+1}|S'_1,\ldots,S'_n)Pr(S'_1,\ldots,S'_{n+1}) \tag{2.36}$$

where

$$Pr(S_1,\ldots,S_{n+1}|S'_1,\ldots,S'_n) = \delta_{C(S'_1,\ldots,S'_{n+1}),(S_1,\ldots,S_{n+1})} = \delta_{(S'_1,\ldots,S'_{n+1}),C^{-1}(S_1,\ldots,S_{n+1})}. \tag{2.37}$$

Therefore

$$Pr(S_1,\ldots,S_{n+1}) = Pr(C^{-1}(S_1,\ldots,S_{n+1})). \tag{2.38}$$

This implies that the probability distribution produced by taking the thermal ensemble for $E'(S'_1,\ldots,S'_{n+1})$ and applying C to the system is equivalent to a thermal ensemble with a new energy function

$$E'(C^{-1}(S_1,\ldots,S_{n+1})). \tag{2.39}$$

A quick calculation using Eq. (2.34) finds that

$$E'(C^{-1}(S_1,\ldots,S_{n+1})) = \frac{\Delta}{2}(1-S_1) + \frac{\Delta}{2}\sum_{i=1}^{n}(1-S_iS_{i+1}) \tag{2.40}$$

which we see is equal to our original energy function E. Reversing the above argument we thus see that the energy function E can equally well be thought of as the ensemble corresponding to E' followed by the application of the mapping C. Further this later ensemble and computation has a clear interpretation in terms of a probabilistic initialization followed by probabilistic gates: when we apply controlled-nots with error target values then this

causes a bit flip error in copying the information down the line. Thus we see that we can derive the same probabilistic gate expression for the one-dimensional Ising model as that which arose by reinterpreting the transfer matrix as a probabilistic circuit.

2.3 Deriving the partition function by recognizing a probabilistic circuit

We now turn to the more general setting of a circuit energy function for a generic circuit $C = (G, L)$. We begin by focusing on circuit energy functions without a forced input.

Let E_C be the energy function for the circuit C. For each logic gate, l, with inputs $i_1, i_2, \ldots, i_j$ and output $t_1, t_2, \ldots, t_k$, define the following *tensor*,

$$M^{(l)}_{i_1,\ldots,i_j,t_1,\ldots,t_k} = \begin{cases} 1 & \text{if } f_l(i_1, i_2, \ldots, i_j) = (t_1, t_2, \ldots, t_k) \\ e^{-b} & \text{otherwise} \end{cases}, \tag{2.41}$$

where $b = \beta\Delta$ as before. Given this definition, we can now write the partition function Z as the contraction of a tensor network [MS08] plus a sum over all inputs and outputs. What is a tensor network? Take a graph and map a tensor to every vertex in such a manner that the tensor has an index for every edge of the relevant vertex. One can then perform a sum over two tensor indices connected via edges in the graph. A tensor network is thus a graph where vertices are tensors, and edges are tensor indices. Free edges are indices which have not been summed over and connected edges are ones for which a sum has been performed. The entire network (graph) then represents itself a tensor with a number of unsummed indices equal to the number of free edges.

Given the tensors $M^{(l)}$ and the graph given by the circuit (which will have free edges for the inputs and outputs), we can build a tensor network out of these $M^{(l)}$s. This tensor network will itself be a tensor network having indices for the inputs and outputs. Call this tensor $M^C_{i_1,\ldots,i_n,t_1,\ldots,t_m}$ for inputs $i_1, \ldots, i_n$ and output $t_1, \ldots, t_m$. Then it is easy to see that the partition function is equal to

$$Z = \sum_{i_1,\ldots,i_n,t_1,\ldots,t_m \in \{+1,-1\}} M^C_{i_1,\ldots,i_n,t_1,\ldots,t_m} \tag{2.42}$$

In other words, the partition function is given by the sum over all inputs, and all outputs of the value of the tensor network.

What does this have to do with probabilistic circuits? Well now instead of using the tensors M, instead use

$$P^{(l)}_{i_1,\ldots,i_j,t_1,\ldots,t_k} = \frac{M^{(l)}_{i_1,\ldots,i_j,t_1,\ldots,t_k}}{1+(2^k-1)e^{-b}} \tag{2.43}$$

or in other words

$$P^{(l)}_{i_1,\ldots,i_j,t_1,\ldots,t_k} = \begin{cases} \frac{1}{1+(2^k-1)e^{-b}} \text{ if } f_l(i_1,i_2,\ldots,i_j) = (t_1,t_2,\ldots,t_k) \\ \frac{e^{-b}}{1+(2^k-1)e^{-b}} \text{ otherwise} \end{cases} \tag{2.44}$$

It is easy to check that this defines a *stochastic* matrix, i.e. a probabilistic gate. Indeed it is a probabilistic gate which performs the correct function of the gate with probability $\frac{1}{1+(2^k-1)e^{-b}}$ and randomly flips the output to one of the other outputs with equal probability $\frac{e^{-b}}{1+(2^k-1)e^{-b}}$.

Notice, importantly, that the difference between $P^{(l)}$ and $M^{(l)}$ is a constant which depends on only on the number of output bits. So suppose we consider the tensor network for the circuit made now with $P^{(l)}$s and not $M^{(l)}$s. Then we can write the partition function as

$$Z_C = K \sum_{i_1,;i_n,t_1,\ldots,t_m \in \{+1,-1\}} P_{i_1,\ldots,i_n,t_1,\ldots,t_m} \tag{2.45}$$

where K is the simple combinatorial factor

$$K = \prod_l (1+(2^{k_l}-1)e^{-b}) \tag{2.46}$$

and k_l is the number of outputs of the lth gate. Now examine $P_{i_1,\ldots,i_n,t_1,\ldots,t_m}$. This is nothing more than the probability that we get output $t_1,\ldots,t_m$ given that we had input $i_1,\ldots,i_n$ to the probabilistic circuit with probabilistic gates $P^{(l)}$. In other words, we can express the partition function as

$$Z_C = K \sum_{inputs} \sum_{output} \Pr(\text{output}|\text{input}) \tag{2.47}$$

Thus the partition function is really hiding a probabilistic computation. Notice further that we can explicitly calculate that partition function because $\sum_{output} \Pr(\text{output}|\text{input}) = 1$,

$$Z_C = K2^n \tag{2.48}$$

where n is the number of input bits.

Now we turn to the case where we force the input to the circuit. Let $x_1, \ldots, x_n$ be the inputs to the circuit C and $E_{C,x}$ be the energy function for the forced computation. Define the modified energy function

$$E_{C,x,y} = E_{C,x}(s) + E_y(s), \tag{2.49}$$

where

$$E_y(s) = \sum_{i=1}^{m} \gamma_i E_{y_i}(s) \tag{2.50}$$

with

$$E_{y_i}(s) = \begin{cases} \gamma_i t_i & \text{if } y_i = 1 \\ \gamma_i (1 - t_i) & \text{if } y_i = 0 \end{cases}, \tag{2.51}$$

and the sum is over the output bits, t_i. Call the partition function associated with this setup $Z_{C,x,y}$.

Note that $Z_{C,x,y}|_{\gamma_1 = \cdots = \gamma_m = 0}$ is equal to the partition function for energy function with a forced input x, $Z_{C,x}$. Define

$$P_{in}(i_1, \ldots, i_n) = \prod_{j=1}^{n} P_{in,j}(i_j), \tag{2.52}$$

where

$$Pr_{in,j}(i_j) = \begin{cases} \frac{1}{1+e^{-b}} & \text{if } i_j = x_j \\ \frac{e^{-b}}{1+e^{-b}} & \text{if } i_j \neq x_j \end{cases}. \tag{2.53}$$

Then it is easy to see that

$$Z_{C,x} = K' \sum_{i_1, ; i_n, t_1, \ldots, t_m \in \{+1, -1\}} P_{i_1, \ldots, i_n, t_1, \ldots, t_m} P_{in}(i_1, \ldots, i_n), \tag{2.54}$$

where

$$K' = K(1 - e^{-b})^n. \tag{2.55}$$

Thus we see that we can interpret $Z_{C,x}$ as a sum over a probabilistic circuit, but now with a probabilistic input corresponding to each bit being flipped with a probability $\frac{e^{-b}}{1+e^{-b}}$. Again, because we are summing over all outputs, we can explicitly perform this sum

$$Z_{C,x} = K'. \tag{2.56}$$

We are interested in computing the probability that the thermal ensemble for the forced input circuit has output given by $y_1, \ldots, y_n$. This can be calculated by

$$\begin{aligned} Pr(t_1 = y_1, \ldots, t_m = y_m) &= -\frac{1}{\beta Z_{C,x,y}} \frac{\partial^m Z_{C,x,y}}{\partial\gamma_1 \cdots \partial\gamma_m}\bigg|_{\gamma_1=\cdots=\gamma_m=0} \\ &= -\frac{1}{\beta Z_{C,x}} \frac{\partial^m Z_{C,x,y}}{\partial\gamma_1 \cdots \partial\gamma_m}\bigg|_{\gamma_1=\cdots=\gamma_m=0} \end{aligned} \tag{2.57}$$

Note that

$$Z_{C,x,y} = K' \sum_{i_1,\ldots,i_n,t_1,\ldots,t_m \in \{+1,-1\}} e^{-\beta E_y(s)} P_{i_1,\ldots,i_n,t_1,\ldots,t_m} P_{in}(i_1, \ldots, i_n) \tag{2.58}$$

so that the partial derivatives may be evaluated only over the first term. There we find that

$$\frac{\partial^m e^{-\beta E_y(s)}}{\partial\gamma_1 \cdots \partial\gamma_2}\bigg|_{\gamma_1=\cdots=\gamma_n=0} = -\beta \prod_{i=1}^{m} \delta_{t_i,y_i}. \tag{2.59}$$

Thus we find that

$$Pr(t_1 = y_1, \ldots, t_m = y_m) = \sum_{s_1,\ldots,s_n} P_{s_1,\ldots,s_n,y_1,\ldots,y_n} P_{in}(s_1, \ldots, s_n). \tag{2.60}$$

This is the main result of this section: the probability of the ground state spin computing model output bits is equal to the probability of running the probabilistic circuit described by P on inputs described by input probability distribution P_{in}. The gates in P are simply noisy versions of the deterministic gates which fail with a fixed probability related to the temperature.

This result is interesting in two manners. First it allows one to use techniques designed for probabilistically failing gates to be ported over to our model. For instance this will allow us to use methods for constructing fault-tolerant circuits. It is also interesting in that it shows that certain many-body quantum systems can be efficiently simulated. Computed circuit energy functions can be simulated by directly implementing the probabilistic gate that these systems correspond to. Here we have shown that output bits for the circuit energy function are related to the output bits for the probabilistic computation. A straightforward generalization shows that this is also true of other bits in the system. We can efficiently simulate systems with computed circuit energy functions by executing the probabilistic computation these energy functions represent.

2.3.1 The Gate Model Derivation

Next let us turn to the same analysis as the prior subsection, but now using the gate trick as we did for the one-dimensional model in Section 2.2.2. To do this we must first define the equivalent of the controlled-not gate. Consider a logical gate l with input $i_1, \ldots, i_j$ and output $t_1, \ldots, t_k$ which computes the function $(t_1, \ldots, t_k) = f(i_1, \ldots, i_j)$. Define the function C_l from $\{0,1\}^j \times \{0,1\}^k$ to $\{0,1\}^j \times \{0,1\}^k$ as

$$C_l(i_1, \ldots, i_j, t_1, \ldots, t_k) = \begin{cases} (i_1, \ldots, i_j, f_l(i_1, \ldots, i_j)) & \text{if } t_1 = \cdots = t_k = 0 \\ (i_1, \ldots, i_j, 0, \ldots, 0) & \text{if } (t_1, \ldots, t_k) = f(i_1, \ldots, i_j) \\ (i_1, \ldots, i_j, t_1, \ldots, t_k) & \text{otherwise} \end{cases} \tag{2.61}$$

This function thus computes the function on input $(i_1, \ldots, i_j, 0, \ldots, 0)$, uncomputes the function on input $(i_1, \ldots, i_j, f_l(i_1, \ldots, i_j))$, and does nothing otherwise. Note that C_l is a bijection and is self-inverse, $C_l^{-1} = C_l$. Defining an order to gates in our circuit such that the circuit computes properly, l_1 first, l_2 second, etc. Then we can define the function on all our bits of

$$C_C = C_{l_r} \circ \cdots \circ C_{l_2} \circ C_{l_1} \tag{2.62}$$

Suppose we initially start with input bits initialized to an input $i_1 = x_1, \ldots, i_n = x_n$ and all other bits initialized to 0. Then if we apply C_C to these bits we will place the result of the

computation in their appropriate locations across the circuit.

Next define an energy function on all of our system. This is most easily defined in terms of the inputs to the full circuit and the outputs to the gates $l \in L$,

$$E'(s') = \sum_{i'_1,\ldots,i'_m} \Delta(\delta_{i'_1,\neg x_1} + \cdots + \delta_{i'_m,\neg x_m}) + \sum_{l \in L} E'_l(t'_1,\ldots,t'_{k_l}). \tag{2.63}$$

and

$$E'_l(t'_1,\ldots,t'_{k_l}) = \begin{cases} 0 & \text{if } t'_1 = \cdots = t'_{k_l} = 0 \\ \Delta & \text{otherwise} \end{cases}. \tag{2.64}$$

This energy function thus assigns for non-global inputs an energy penalty for output to gates being anything different that 0^{k_l}. For global inputs it adds a penalty for every input which is not the correct input from x. The thermal ensemble resulting from this system is then trivially described: the input qubits are in the correct input x_i with probability $\frac{1}{1+\exp(-b)}$ and the internal output bits of a gate are grouped together and have a probability $\frac{1}{1+(2^{k_l}-1)\exp(-b)}$ being all zeros and probability $\frac{e^{-b}}{1+(2^{k_l}-1)\exp(-b)}$ to be anything else.

Imagine applying C_C to the state described by the ensemble for $E'(s')$. Clearly the ground state of $E'(s')$ is such that the circuit C will be correctly computed. Further, because the ensemble related to $E'(s')$ has erred inputs, the resulting probability distribution will have erred inputs. Finally the gates will function properly only when the output was properly initialized to all 0s. This occurs with probability $\frac{1}{1+(2^{k_l}-1)\exp(-b)}$ for logic gate l and otherwise the gate fails by an equally probable error state with probability $\frac{e^{-b}}{1+(2^{k_l}-1)\exp(-b)}$. In other words the ensemble is exactly the one describing the probabilistic circuit in Section 2.3. Let us now show that this corresponds to the thermal ensemble of our original E_C construction.

As in Section 2.2.2 the trick here is to express the probability distribution for the ensemble after applying C_C as

$$Pr(s) = \sum_{s'} Pr(s|s')Pr(s'), \tag{2.65}$$

where

$$Pr(s|s') = \delta_{C_C(s'),s} = \delta_{s',C_C^{-1}(s)}, \tag{2.66}$$

such that

$$Pr(s) = Pr(C_C^{-1}(s)). \tag{2.67}$$

Thus the probability distribution resulting from after the application of C_C to the $E'(s')$ thermal ensemble is equal to the probability distribution arising from $E'(C_C^{-1}(s))$.

We will now show the $E'(C^{-1}(s))$ is exactly the energy function we would define for a ground state spin computing construction for our circuit C. Recall that

$$C_C^{-1} = C_{l_1} \circ C_{l_2} \cdots \circ C_{l_r}. \tag{2.68}$$

Note that in evaluating $E'(C^{-1}(s))$, which is a sum of terms, we can evaluate the action action of C_C^{-1} on each of these terms separately and then re-sum. Thus we first examine

$$C_C^{-1}\left[\sum_{i'_1,\ldots,i'_m} \Delta(\delta_{i'_1,\neg x_1} + \cdots + \delta_{i'_m,\neg x_m})\right] = \sum_{i'_1,\ldots,i'_m} \Delta(\delta_{i'_1,\neg x_1} + \cdots + \delta_{i'_m,\neg x_m}) \tag{2.69}$$

This is because the inputs to the circuit commute through the individual gate elements C_{l_i}. Indeed this observation along with the order of C_C^{-1} allows one to derive the term arising from each $E'_l(C_l(i_1,\ldots,i_{j_l},t_1,\ldots,t_{k_l}))$ term independently. In particular the only term which is not Δ for this energy function is

$$E'_l(C_l(i_1,\ldots,i_{j_l},f_l(i_1,\ldots,i_{j_l}))) = 0 \tag{2.70}$$

But this simply means that

$$E'_l(i_1,\ldots,i_{j_l},t_1,\ldots,t_{k_l}) = 0 \tag{2.71}$$

only when $(t_1,\ldots,t_{k_l}) = f_l(i_1,\ldots,i_{j_l})$ and is Δ otherwise. This is just our normal definition of the circuit energy function.

Thus we see that using the C_C construction one can construct a probabilistic circuit from the thermal ensemble for $E'_{C,x}(s')$ which is the probabilistic circuit described in Section 2.3. Further we have shown that this can be thought of as arising from the original energy function $E_{C,x}$. This provides an alternative derivation of our main result.

2.4 Conditions for the probabilistic circuit interpretation

Having shown that classical ground state spin computing at non-zero temperature can be mapped to probabilistic circuits it is interesting to return to our original model and consider what generalizations of the energy function allow for this mapping to occur. The crucial assumption during our derivation of the probabilistic re-interpretation of the partition function was that we could convert the tensors $M^{(l)}$ of Eq. (2.41) to tensors $P^{(l)}$ of Eq. (2.43) in such a way that $P^{(l)}$ could be interpreted as a probablistic gate. This requires (1) the division of tensor indices into input and output indices and (2) for each input to the circuit the normalization to make the outputs probabilities (positive and sum to unity) is the same.

Let us begin by showing that not all tensors and divisions into input and output vertices can result in a probabilistic interpretation. This can be demonstrated with a simple example on two bits:

$$E(s_1, s_2) = \begin{cases} 0 & \text{if } s_1 = s_2 = 0 \\ \Delta & \text{if } s_1 = s_2 = 1 \\ 2\Delta & \text{otherwise} \end{cases} . \tag{2.72}$$

If we chose s_1 as an input and s_2 as an output, this gives rise to the tensor

$$M = \begin{bmatrix} 1 & e^{-2b} \\ e^{-2b} & e^{-b} \end{bmatrix} . \tag{2.73}$$

In order that the first column of this matrix sum to unity we must divide by $1+e^{-2b}$. However when we do this the second column will not sum to 1. A similar argument occurs if we had chosen s_1 as the output and s_2 as the input. Thus there is no way to scale M in such a way that it can be interpreted as a probabilistic gate.

A more relevant example of a failure for a model to not be amenable to our scaling arising in quantum-dot cellular automata models. In quantum dot cellular automata models bistable quantum dots are engineered in such a way that the ground state enacts a classical computation as in our model. Most of the treatments of these models deal with the quantum mechanics of these devices [LTPB93, LTP94, WS98]. One can, however, build a classical

model of these systems as was done, for example, by Wang in [WL04]. In this approximation one derives a classical statistical mechanical many spin system as in our model. When one does this for a quantum-dot cellular automata wire one ends up with exactly the model we can consider above in Section 2.2. However when one considers this for the majority gate constructions (see [LTPB93, LTP94, WL04]) one obtains a model for which our technique cannot be applied. In particular the majority gate that they propose has a classical energy function which is

$$E_M(S_1, S_2, S_3, S_m) = -\Delta(S_1 S_m + S_2 S_m + S_3 S_m) = -\Delta(S_1 + S_2 + S_3) S_m. \tag{2.74}$$

Here S_1, S_2, S_3 are the input spins and S_m is the output spin. (Note that we have used one less spin than in the traditional presentation, but this does not affect our conclusions as one can think of the extra spin as being connected to the central spin by an identity gate.) Note that the lowest energy configuration *for a given* input correctly computes the majority function. That is if we fix S_1, S_2, and S_3, then the lowest energy value of S_m is given by the majority of the inputs, $S_m = MAJ(S_1, S_2, S_3)$. Note, however, that if we sum over all outputs for input $S_1 = S_2 = S_3 = +1$

$$\sum_{S_m | S_1 = S_2 = S_3 = +1} \exp(-b E_M(S_1, S_2, S_3, S_m)) = e^{-3b} + e^{-3b}, \tag{2.75}$$

but if one uses input $S_1 = S_2 = +1$ and $S_3 = -1$ one obtains

$$\sum_{S_m | S_1 = S_2 = +1, S_3 = -1} \exp(-b E_M(S_1, S_2, S_3, S_m)) = e^{b} + e^{-b}. \tag{2.76}$$

Thus we cannot properly globally normalize the entire tensor in such a way as to to obtain a probabilistic gate.

However, the same two-spin energy function can be used to describe a fanout gate, in contrast to the many-spin interaction used in Eq. (2.1),

$$E_F(S_1, S_2, \ldots, S_{k+1}) = -\Delta S_1 \sum_{i=2}^{k+1} S_i. \tag{2.77}$$

Here S_1 is the input and the other spins are the output. By changing the direction of the circuit, we can show that this element can be represented as a probabilistic gate. I.e. the majority gate used in quantum dot cellular automata can not be mapped to a probabilistic circuit, but reversing the role of the inputs and outputs, this gate can be used to construct a fan-out gate. To see this note that, as for our previous fanout model, the ground state of this system, where the computation is performed, is degenerate. However now, as opposed to a single excited state there are multiple excited states. For example if we consider a fan-out to two spins we obtain an M as given in Table 2.1. From this table we see that while the outputs

S_1	S_2	S_3	$e^{-\beta E_F}$
$+1$	$+1$	$+1$	e^{2b}
$+1$	$+1$	-1	1
$+1$	-1	1	1
$+1$	-1	-1	e^{-2b}
-1	$+1$	$+1$	e^{-2b}
-1	$+1$	-1	1
-1	-1	1	1
-1	-1	-1	e^{2b}

Table 2.1: The 2 output fan-out Boltzman factor.

that are in error have different Boltzman factors, we see that the sum over the output spins is indeed independent of the output. Indeed when we interpret this as a probabilistic gate by dividing out by $K = 2 + e^{2b} + e^{-2b}$, then we see that this is a probabilistic gate which, instead of having a global failure where all possible output failures have equal probability, the fan-out spins have independent error probabilities. In other words it is as if the fan-out occurred and then with probability $\frac{e^{-b}}{e^{-b}+e^{b}}$ each of the outputs is *independently* flipped. We will use this fan-out gate in the next section when we discuss fault-tolerance.

What is a necessary and sufficient condition for an energy function to allow for a proper normalization to a probabilistic gate? If we insist that this work for all temperatures, it is easy to give this answer. Let $E^{(l)}(i_1, \ldots, i_j, t_1, \ldots, t_k)$ denote the energy function for a logic gate with inputs $i_1, \ldots, i_j$ and output $t_1, \ldots, t_k$. Let $\mathcal{E}(i_1, \ldots, i_k) = \{E(i_1, \ldots, i_k, t_1, \ldots, t_k) | t_1, \ldots, t_k \in \{0, 1\}\}$ denote the set of energies for a given input. Then the necessary and sufficient condition is that the $\mathcal{E}(i_1, \ldots, i_k)$'s must be permutations of each other. The sufficiency of this condition is obvious. Necessity follows from noting that we are requiring this to hold for all temperatures and thus the sum over outputs of the $\exp(-\beta E)$ must be a rearrangement of this sum and therefore a permutation of the output energies. Note that this condition implies that the ground states of a logic gate energy function must be degenerate for all possible inputs (but note that this alone is not sufficient for the condition to hold.)

2.4.1 Identifying Probabilistic Circuits in Thermal Systems

Having established a necessary and sufficient condition for the possibility of scaling the $M^{(l)}$ tensors as probabilistic gates, a natural follow up question is how one can determine whether it is possible to take an energy function $E(s_1, \ldots, s_n)$ and determine whether or not it can be interpreted as a probabilistic circuit. Here we will show that this problem is computationally intractable, even when we are given information about the breakdown of $E(s_1, \ldots, s_n)$ into terms which we wish to interpret as gates.

Suppose that we are told the $E(s_1, \ldots, s_n)$ is made up of a sum over logic gates and preparations

$$E(s_1, \ldots, s_n) = \sum_l E_l, \tag{2.78}$$

i.e. we are given a specification of the E_l in terms of a constant number of bits and we are told that each of these E_l will correspond to a logic gate or a preparation of initial inputs. We can then ask: can we interpret this as a probabilistic circuit with or without prepared inputs? Every E_l involves some inputs and output bits, but in general we will not be told which of these are inputs and which are outputs. In general an $M^{(l)}$ may thus have many

different tuples of input bits and output bits such that we may interpret $P^{(l)}$ as a probabilistic gate by proper global normalization. For example gates which have doubly stochastic $P^{(l)}$ tensors may be run either backwards or forwards (a doubly stochastic matrix is a stochastic matrix whose rows and columns sum to 1.) Another example is given by a fan-out gate in our original energy function model

$$E_f(S_1, S_2, S_3) = \begin{cases} 0 & \text{if } S_1 = S_2 = S_3 \\ \Delta & \text{otherwise.} \end{cases} \tag{2.79}$$

In this case we may interpret the $M^{(l)}(S_1, S_2, S_3)$ arising from this energy function as having input S_1 and outputs S_2 and S_3, input S_2 and outputs S_1 and S_3, or input S_3 and outputs S_2 and S_3. Contrary to this, suppose that the energy function is that of the one-to-two fanout described in Eq. (2.77),

$$E_F(S_1, S_2, S_3) = -\Delta S_1(S_2 + S_3). \tag{2.80}$$

The $M^{(l)}(S_1, S_2, S_3)$ for this energy function has only one possible orientation between inputs and output: S_1 is an input and S_2 and S_3 are outputs. This in general will lead to the following question: given the possible input and outputs tuples for the $M^{(l)}$s is it possible to assign these in such a way as to interpret the resulting total tensor as a probabilistic circuit. We will now show that this problem is NP-complete and is therefore (unless P = NP), in general, intractable.

To see that the problem is NP-complete proceed as follows. Assign to each bit of our physical system a boolean variable, x_i. Assume that a given logical energy function E_l is involved in only a constant number of bits. Since E_l is involved in a constant number of bits we can explicitly calculate all of the sets of inputs and output bits for which $M^{(l)}$ can be globally normalized so as to make it a probabilistic gate. The variables x_is are going to represent whether the corresponding bit is associated is an input or and output. In particular, TRUE, will represent that the bit is an input and FALSE will represent that the bit is an output. Thus from the sets of input and output bits for which one can consistently label inputs to $M^{(l)}$ we can construct a boolean function on the relevant bits such that this

function is true if and only if a valid assignment of boolean variables respects a possible input and outputs. Thus, for example, for E_f above in Eq. (2.79), the boolean function would be

$$[(x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3)] \tag{2.81}$$

Constructing one such boolean function for each logic gate we can then construct a boolean function on all bits x_i which is the conjunction (logical and) of all of the gate boolean functions. Thus we see that by calculating for each E_l the possible combinations of inputs and outputs that have an allowed interpretation in terms of a probabilistic circuit, we have a one-to-one mapping of the problem to the problem of deciding whether there is a satisfying assignment for the boolean function made up of the conjunction of logical terms with only a constant number of involved bits. In order to demonstrate that the problem is NP-complete we need now only show that the boolean function we can construct in this manner are general enough so as to yield a version of the satisfaction problem which is NP-complete. In particular it is clear that this problem is in NP, since we can evaluate in polynomial time whether a given assignment of inputs and outputs can be interpreted as a probabilistic circuit. To demonstrate NP-completeness we need to show that it is at least as hard as another NP-complete problem.

We will show that the problem is NP-complete by showing that some instance of the problem can be mapped to the one-in-three satisfiability problem of Schaefer [Sch78] (now sometimes called the monotone one-in-three satisfiability problem.) Define $R(x, y, z)$ as the boolean function which is true iff exactly one of (x, y, z) is true. Notice that this is exactly the boolean formula of Eq. (2.81) (with $x_1 = x$, $x_2 = y$, and $x_3 = z$.) In the one-in-three satisfiability problem one is given a boolean function which is the conjunction (logical and) of clauses each of which are of the form $R(x, y, z)$ for a choice of variables x, y, z in the problem. Schaefer proved that the one-in-three satisfiability problem is NP-complete. Actually there is a slight subtlety here as the problem Schaefer considers must allow repeated variables. We will now show that it is possible to take a one-in-three satisfiability problem and convert it into an instance of the problem of spotting whether a given energy function can support

interpretation as a classical probabilistic circuit.

As mentioned above the one-in-three satisfiability problem has as input a conjunction of boolean functions which are all of the form $R(x, y, z)$, and where some boolean variables can be repeated,

$$f(x_1, \ldots, x_n) = \bigwedge_{j=1}^{r} R(a_i, b_i, c_i), \tag{2.82}$$

where $a_i, b_i, c_i \in \{x_1, \ldots, x_n\}$ and none of the a_i, b_i, and c_i are the same variable for fixed i. Let $X_i = \{a_j | a_j = x_i\} \cup \{b_j | b_j = x_i\} \cup \{c_j | c_j = x_i\}$ denote the set of a, b, c variables that are x_i. We will define an energy function on $3r + \sum_{i=1}^{n} |X_i|$ spins which, when converted into the problem of trying to assign consistent inputs and outputs will yield exactly the boolean function f of Eq. (2.82). Call the first $3r$ spins S_i and the last $X_{tot} = \sum_{i=1}^{n} |X_i|$ spins T_i. First define the energy function

$$E_M(S_1, \ldots, S_{3r}) = \sum_{j=1}^{r} E_f(S_{3j-2}, S_{3j-1}, S_{3j}), \tag{2.83}$$

where E_f is the fan-out energy function defined in Eq. (2.82). When one converts this into a logical expression for whether the spins are inputs or outputs, one gets exactly the one-in-three logical clause of Eq. (2.81).

However, at this point, the variables that will arise in these clauses are all different. This is what the other X_{tot} spins are for. Define the following energy function for the "identity" gate:

$$E_I^{(k)}(P_1, P_2, \ldots, P_k, Q_1, Q_2, \ldots, Q_k) = \begin{cases} 0 & \text{if } P_1 = Q_1, P_2 = Q_2, \ldots, P_k = Q_k \\ \Delta & \text{otherwise} \end{cases}. \tag{2.84}$$

This energy function corresponds to a logical reversible gate which acts as identity from the P_i spins to the Q_i spins. This energy function has the property that the P_i variables are all either inputs or all outputs (and the Q_i variables are also all either inputs or outputs, opposite to the P_i variable assignment.) Thus we can use gates of this form to effectively make the spins of the E_M gates constructed above the same variable. In particular define

the following energy function

$$E_e(S_1,\ldots,S_{3r},T_1,\ldots,T_{X_{tot}}) = \sum_{i=1}^{n} E_I^{(|X_i|)}(S_{X_i[1]},S_{X_i[2]},\ldots,S_{X_i[|X_i|]},T_{X_t^{(i)}},T_{X_t^{(i)}+1},\ldots,T_{X_t^{(i)}+|X_i|}), \tag{2.85}$$

where $X_t^{(i)} = \sum_{j<i} |X_i|$ and $X_i[j]$ is the jth element of X_i translated over to a S_i variable, i.e. a_i corresponds to S_{3i-2}, b_i corresponds to S_{3i-1} and c_i corresponds to S_{3i}.

Now consider the energy function

$$E(S_1,\ldots,S_{3r},T_1,\ldots,T_{X_{tot}}) = E_e(S_1,\ldots,S_{3r},T_1,\ldots,T_{X_{tot}})) + E_M(S_1\ldots,S_{3r}), \tag{2.86}$$

and stipulate that every term in the sum of E_e will correspond to one gate and every term in E_M will correspond to one gate. Then if we convert this problem over into a boolean satisfaction problem as describe above, where each spin is a boolean variable representing whether the spin is an input or an output, we will obtain a boolean expression that is equivalent to the problem described in Eq. (2.82). Thus we have shown that we can, for a given one-in-three satisfaction problem, Eq. (2.82) construct an energy function, along with labeling of the terms in this function corresponding to different gates, such that determining whether this energy function can be reinterpreted as a probabilistic computation is equivalent to the original one-in-three satisfaction problem. This implies that the problem of determining whether such probabilistic computations exist is at least as hard as this NP-complete problem. Combined with the fact the problem is in NP this means that the problem of deciding whether a given energy function can be thought of as enacting a probabilistic circuit is NP-complete.

Thus we have shown that the problem of determining a many-spin statistical mechanics system enacts a ground state spin computation, even given the decomposition of this system into logical gates, is NP-complete. We note in passing that this result is of perhaps some philosophical significance: determining whether a system can be thought of as a computer is shown to be computationally intractable (assuming $\mathsf{P} \neq \mathsf{NP}$).

2.4.2 Examples of models with a gate Interpretation

So far we have considered spin models in a very general setting unrelated to physical theories. We point out, here, however, that there are a variety of models that have arisen in physics which admit such an interpretation. We have already seen that that one dimensional Ising model admits an interpretation as a series of identity gates which err with some probability. Here we point out some other models which admit such interpretations.

Ising Models on Trees

A tree is a connected graph, $T = (V, E)$, with vertex set V, edge set E, and no cycles. Assign to each spin a vertex, S_v for $v \in V$. To each edge $e = (v, w)$ assign an non-zero energy for the Ising coupling, $J : E \to \mathbb{R} - \{0\}$. Then the Ising model on the tree has an energy function given by

$$E(s_1, \ldots, s_{|V|}) = \sum_{(v,w)\in E} J_{(v,w)} S_v S_w. \tag{2.87}$$

Fix a root vertex $v_r \in V$. First consider the model at zero temperature. At $T = 0$, following our discussion of the fan-out in Section 2.4, it is easy to see that the ground state of this model can be thought of as a series of fan-out gates originating from the root vertex v_r with possible bit flips applied to the fan-out depending on the sign of $J_{(v,w)}$. At $T \neq 0$ we see that, as in the discussion of the fan-out gate, each of the individual fan-out wires (with possible bit flips), will fail independently with a probability of

$$p_{(S_v,S_w)} = \frac{e^{\beta J_{(v,w)}}}{e^{\beta J_{(v,w)}} + e^{-\beta J_{(v,w)}}}. \tag{2.88}$$

Note that a failure for an antiferromagnetic coupling (bit flip) corresponds to an identity gate. Ising models on trees have been studied in a variety of settings [Lyo89, VTB09] and in fact the interpretation of this model in terms of a probabilistic circuit has been used extensively in the computer science literature, where the model goes under the name of *broadcasting* [EKPS00].

Z_2 Lattice Pure Gauge Theories

Perhaps slightly more interesting that Ising models on trees is the fact that Z_2 lattice gauge theories [BDI75] can be cast as a probabilistic gate circuits. For definiteness, first consider a (pure) Z_2 lattice gauge theory on a two dimensional square lattice with closed boundaries. In this model one places spins on the edges of the lattice and defines the energy function

$$E = \sum_{p \in P} \sum_{S_1, S_2, S_3, S_4 \in \delta p} J_p S_1 S_2 S_3 S_4 \tag{2.89}$$

where P is the set of all plaquettes of the square lattice, and δp are the spins surrounding plaquette p. Consider an individual term in this sum, and assume for simplicity that for all plaquettes, $p \in P$ that $J_p = J < 0$. We can view the individual term $JS_1S_2S_3S_4$ as a gate element with either 0, 1, 2, or 3 inputs and 4, 3, 2, and 1 outputs respectively. For example if we view it as a gate with 2 inputs and 2 outputs, then, in its ground state this corresponds to a probabilistic gate which, with equal probability, changes the input to a set of spins with the same parity of -1s as the input. In other words the logical evolution is

$$\begin{aligned}
(+1,+1) &\rightarrow 50\%\ (+1,+1) \text{ and } 50\%\ (-1,-1) \\
(+1,-1) &\rightarrow 50\%\ (+1,-1) \text{ and } 50\%\ (-1,+1) \\
(-1,+1) &\rightarrow 50\%\ (-1,+1) \text{ and } 50\%\ (-1,+1) \\
(-1,-1) &\rightarrow 50\%\ (+1,+1) \text{ and } 50\%\ (-1,-1).
\end{aligned} \tag{2.90}$$

The gate with 3 inputs and 1 outputs is deterministic and is given by $f(S_1, S_2, S_3) = S_1S_2S_3$. The gate with 0 inputs corresponds to a preparation of 4 spins which have an even number of -1 spins and this state is prepared with equal probability. The gate with 1 input and 3 outputs is a probabilistic gate which outputs an even number of -1s if the input is $+1$ all such possibilities occurring with equal probability. Similarly it outputs an odd number of -1s if the input is -1 all such possibilities occurring with equal probability. By picking boundary terms to serve as input and output bits, it is clear that it is possible to order the energy terms in the above sum in such a way that we can interpret the ground state

as a probabilistic cellular automata like model using the above gates. At finite gate these probabilistic gates all fail with some probability, where failure results in an equally likely failed output for the gate. While we have define this for the two dimensional square lattice, it is clear that the above construction can result in a probabilistic circuit for a far greater number of lattices, the main criteria being that there exists a way to label inputs and outputs in such a way that a proper circuit is constructed.

2.5 Fault-tolerance at non-zero temperature

Having shown how the computed circuit energy functions give rise to thermal ensembles which can be interpreted as probabilistic circuits with probabilistic inputs, we now discuss how it is possible to make the model of classical ground state spin computation fault-tolerant at finite temperature. Indeed this is now rather simple given that we have a mapping between the behavior of these systems at finite temperature and probabilistic circuits which have gates and preparations failing with a fixed probability. We will need, however, to use the more general class of energy functions considered in Section 2.4.

The general theory of computing in the presence of gates which fail goes back at least as far as the work of von Neumann [von56]. Von Neumann considered a model in which each logic gate failed with exactly the probability ϵ. In order to get this model to compute reliably, von Neumann used two major techniques. The first techniques is that one needs to suitably encode information: in order to do this one encodes a single bit across a bundle of wires in a logic circuit and interprets this as 0 (or 1) when a majority of the wires are in the 0 (or 1.) Given such an encoding it is then easy to compute in a parallel manner such that the effect of gates failing is simply to decrease the proportion of 0s (or 1s) in an encoded 0 (or 1.) However, the fact that the gates fail does cause the ratio of correct bits in an encoded bundle of bits to decrease. In order to deal with this von Neumann used a second technique whereby faulty gates were used to perform error correction. Actually this portion of von Neumann's analysis is not quite satisfactory, as it requires the use of a random permutation. However Pippenger [Pip85] has shown that one can de-randomize this construction (one can

obtain reasonable parameters for this method by using the expanders defined in [LPS86]). The end result of this is that one can show that if one wishes to compute a logical circuit of size n to an accuracy $1-\delta$ (that is, have the circuit fail with probability δ) with gates that fail at $\epsilon < \epsilon_t$ for some fixed threshold ϵ_t, one can do this using a circuit with the unreliable gates which is only a $O(\log^k \frac{1}{\delta})$ larger, where k is a fixed constant. This effectively means that one can create for all effective purposes reliable logic gates out unreliable logic gates with an overhead which is logarithmic in the inverse of the error desired.

Having described von Neumann's construction one can now see how it is possible to make classical ground state spin computing fault-tolerant. In particular we have shown above how the thermal ensemble for such systems gives rise to probabilistic gates. For gates constructed using energy functions like Eq. (2.1) we have shown that the thermal ensemble arising for these functions is equivalent to a circuit in which gates fail with probability $\frac{(2^k-1)e^{-b}}{1+(2^k-1)e^{-b}}$ if the gate has k output bits. For $k=1$ this is $\frac{e^{-b}}{1+e^{-b}}$. We would like to use this for von Neumann's ϵ failure probability. One complication arises, however, which is that in von Neumann's model the fan-out gates do not fail. These gates are needed during the error correcting stage of his procedure. We get around this by using fan-out gates constructed from energy functions like those in Section 2.4. In particular define the fan-out gate via Eq. (2.77). As shown in Section 2.4 this gives rise in our probabilistic gate setting in which each fan-out resulting wire fails with probability $\frac{e^{-b}}{e^{-b}+e^{b}}$. These independent failures can then be reinterpreted as errors for the gates that follow them in von Neumann's error correcting construction. One should note that there is a subtlety here in that if we desire to recover von Neuman's error model where the gates fail with exactly probability ϵ then a complicated calculation is needed in order to figure out ϵ in this model. However, for large enough β the effect will mostly be that the error probabilities add such that ϵ will be the sum of the fan-out error and the gate error.

Putting this together we can conclude that there is a critical temperature below which one can use fault-tolerant constructions to design ground state spin energy functions which compute correctly with probability $1-\delta$ by using $O(\log^k \frac{1}{\delta})$ more interactions in the energy

functions. This means that while the general model of ground state spin computing is not tolerant to errors, for a cleverly designed setup one can overcome this difficulty and build robust devices, assuming that the temperature is low enough to keep the error probability below the error threshold.

2.6 A generalized Cook-Levin theorem for non-zero temperature

In this section we make a connection between our model of classical ground state spin computing and the theory of computational complexity. We review the Cook-Levin theorem, which shows that the question of whether a non-deterministic Turing machine accepts or rejects is equivalent to asking whether an instance of the boolean satisfaction problem SAT has a satisfying assignment. In other words, SAT is NP-complete. We consider the complexity class MA, a classical probabilistic analog of NP, and generalize the main idea in the proof of the Cook-Levin theorem to show that estimating certain expectation values at the boundary of a thermal circuit is equivalent to deciding whether a probabilistic circuit accepts or rejects, and so this problem is MA-complete.

2.6.1 Review of the classical Cook-Levin theorem

The Cook-Levin theorem [Coo71, Tra84] is a fundamental result in the theory of computational complexity. Here we briefly review this result. For more details one is referred, for example, to the classic textbook [Sip05]. To define the Cook-Levin problem we introduce the SAT problem.

> **Boolean Satisfaction Problem** (SAT): Given as input a description of a Boolean function from n boolean inputs to one output, $f : \{0,1\}^n \to \{0,1\}$, decide whether there exists a set of inputs $x_1, x_2, \ldots, x_n$, with each $x_i \in \{0,1\}$, such that $f(x_1, x_2, \ldots, x_n) = 1$.

Cast in terms of languages, the SAT language is the set of strings describing a Boolean function such that the Boolean function has some input which evaluates to 1. The Cook-

Levin theorem then states that the SAT is complete for NP. This means that every language in NP can be converted by a deterministic Turing machine in polynomial time to a SAT language and every SAT language can be converted into a language in NP. Thus, in a real sense, SAT captures what it means for a lanuage to be in NP. For example, if we could efficiently solve a SAT problem in polynomial time we would be able to solve every problem in NP in polynomial time and vice versa. This is especially important considering that numerous problems beside SAT have been identified as NP-complete. This class of problems are essentially "the most difficult" in NP and are correspondingly the problems for which no known polynomial time algorithms for all instances of the problems in the class are thought to exist.

We are most interested in the proof of the Cook-Levin theorem. Before reviewing this it is useful to provide another equivalent characterization of NP which uses, instead of a non-deterministic Turing machine, a prover and a verifier. The prover is assumed to have unlimited computational power while the verifier is limited to running a polynomial time algorithm on a deterministic Turing machine. A language, L, is in NP if the prover can supply to a verifier a polynomial sized proof p that a instance x is in the language. This proof must always be correct, i.e. the verifier must always be convinced, by using a Turing machine from P, that the instance x is the in the language. Similarly the prover must never be able to convince the verifier incorrectly. In other words if x is not in L, then the verifier must always reject. For example SAT is in NP by this definition since the prover can simply supply a satisfying input to the boolean function if the function can be satisfied, and the verifier can compute the boolean function on this input in polynomial time. Similarly if the function does not admit a satisfying assignment the verifier can never be convinced that the instance is not in the language because no string the prover can give will lead to a satisfied boolean function. Note that in the prover-verifier setting we do not require that the prover try to convince the verifier that an instance x is not in L, only that if it is true that x is not in L that the prover cannot convince the verifier that x is in L.

Given the prover/verifier definition of NP we can now discuss the Cook-Levin theorem. The Cook-Levin theorem states that SAT is NP-complete. A problem, P, is NP-complete if and only if every problem in NP can be transformed into P by a deterministic Turing machine in polynomial time. Thus SAT, since it is NP-complete, captures much of what is difficult about problems in all of NP: if there were a polynomial time algorithm for SAT then there would be a polynomial time algorithm for every problem in NP.

The proof that SAT is NP-complete has two directions. We have already stated the first direction, that SAT is a NP problem. The meat of the proof is in the other direction, that every problem in NP can be converted into a SAT problem. Here we do this in a slightly non-standard manner, using the prover-verifier definition of NP. In particular for any problem, L, in NP we know that there exists a verifier who takes as input the problem instance, x and a certificate p, runs these on a deterministic Turing machine and verifies whether the instance is in the language. Imagine now the time history of such a verification by a Turing machine. Initially the Turing machine is at the beginning of the string, in its initial state, and the input string is on the tape. At the next time step, the machine can change its state, move left, right or stay in place, and possibly change the symbol on the tape where the machine is located. Notice, however that the contents of a cell on the tape, whether the machine is at that location, and the internal state of the machine if it is on that cell at the next time step is a function only of the contents of the cell and its neighbors at the prior step, whether the machine was on one of those cells, and its internal state if it was. In other words the computation can be described *locally* in space and time. Given this we can now describe how to convert a verifier, which is just a deterministic Turing machine and the strings representing the instance x and the proof certificate p into a problem in SAT.

Suppose that the verifiers deterministic Turing machine runs in time $p(n)$ where $n = |x| + |p|$ is the size of the instance and the proof and p is a polynomial function. Define the following boolean variables, $L_{i,j,k}$, $H_{i,k}$, S_{qk} to represent the history of the deterministic Turing machine in verifying the proof. We think about the first two variables as being arranged in a large $O(p(n)) \times (p(n)+1)$ tableau onto which the computational history of the

Variable	Represents
$L_{i,j,k}$	True if tape cell $-p(n) \leq i \leq p(n)$ contains symbol j at time $0 \leq k \leq p(n)$
H_{ik}	True if the Turing Machine is at tape cell $-p(n) \leq i \leq p(n)$ at time $0 \leq k \leq p(n)$
S_{qk}	True if Turing machine is in state q at time $0 \leq k \leq p(n)$

Table 2.2: Variables in the Cook-Levin theorem.

deterministic Turing machine is carried out. We can now define a SAT problem over these variables which is satisfiable iff the deterministic Turing machine accepts the input string x, p. This SAT formula is conjunction (logical and) of a clauses which enforce (a) that the initial contents of the tape are x, (b) that the machine is in the correct initial state, (c) the initial location of the head of the tape, (d) that there is only one symbol per tape cell, one state per time, one tape position per time, (e) that the tape cell is unchanged unless it is the one which can be written upon, (f) the machine properly updates according to the Turing machine transition function, and (g) the machine finishes in an accepting state. For example condition (b) consists of clauses with a single variable $S_{q_0,0}$ where q_0 is the initial state of the Turing machine. The most complicated clause in this construction is the one that encodes the local update rule for the machine, (f). This can be done by adding the clauses

$$(H_{i,k} \wedge S_{q,k} \wedge L_{i,j,k}) \rightarrow (H_{i+d,k+1} \wedge S_{q',k+1} \wedge L_{i,j',k+1}) \tag{2.91}$$

for valid transitions of the Turing machine (that is for q, q', j, j', d representing a valid change of state, symbol, and direction of movement for the read/write head of the Turing machine.) Putting this together one thus sees that once can take the deterministic verifier along with the instance and x and construct a polynomial sized boolean formula f which is satisfied iff the verifier accepts the proof on the instance. Note that we have not forced a constraint on the proof input to the circuit, but that the definition of NP in a prover-verifier setting leads to the satisfiability of the formula f iff the instance x is in the language.

We can now see that connection between the Cook-Levin theorem and the classical ground state spin computing model. In particular a central part of the proof of this theorem is that one can construct a satisfiability formula directly from the history of a computation. In a similar manner classical ground state spin computing constructs an energy function whose ground state is the history of a computation. The conversion of a logic circuit to such an energy function is essentially the arithmetization technique of computational complexity [Sha92]. Note however that the important role here of the input and outputs to the constructed formula: in the case where the output is forced, we are led to NP-complete problems, but when the input is forced then we are led to efficient constructions. Motivated by this it is interesting to consider how the above constructions works but now at finite temperature.

2.6.2 Boundary expectation value problem is MA*-Complete*

We will now show that it is possible to use classical ground state spin computing to define, in analogy with $3-\mathsf{SAT}$ a **Promise-**MA-complete problem. MA stands for Merlin-Arthur and is essentially a probabilistic generalization of the complexity class NP [Bab85, GS86]. It is defined as the follows:

> **Promise-MA**: A language $L = L_{yes} \cup L_{no} \subset \Sigma^*$, $L_{yes} \cap L_{no} = \emptyset$, is a promise problem in Merlin-Arthur (**Promise-**MA) if there is a probabilistic polynomial-time Turing machine V and a polynomial p such that for all strings $x \in L$,
>
> - If $x \in L_{yes}$ then there exists a w such that $|w| = poly(|x|)$ and $V(x, w)$ accepts with probability at least $2/3$.
> - If $x \in L_{no}$ then for all w such that $|w| = poly(|x|)$ then $V(x, w)$ accepts with probability at most $1/3$.

Here Σ is the alphabet over which the language is defined. This version of the definition of MA is a promise problem, since $L_{yes} \cup L_{no}$ does not necessarily cover all possible elements of Σ^*.

The reason that this class is called Merlin-Arthur is because it can be defined in terms of an interactive proof system between a computationally unbounded prover, Merlin, and a more limited verifier, Arthur, who can perform feasible probabilistic computations (polynomial time probabilistic computations with bounded error, BPP.) Merlin is trying to convince Arthur that a certain instance x is in a language L_{yes}. He does this by communicating a polynomial size proof w to Arthur. Arthur can then take this proof and run a polynomial time probabilistic computation on this proof the result of which is: if $x \in L_{yes}$, then Arthur is, with high probability, convinced the x is in L_{yes}, while if $x \in L_{no}$ no matter what proof Merlin supplies Arthur cannot be convinced that $x \in L_{yes}$ except with a small probability.

We define the following problem:

Boundary Expectation Value (BEV): Let k be a positive integer and β a k digit number. Suppose we are given an energy function on n bits, $E(s_1, \ldots, s_n)$ which can be written as a sum of energy functions

$$E(s) = \sum_l E_l(s) \tag{2.92}$$

in such a way that this energy function can be interpreted as a circuit energy function (see Eq. (2.1)) with $j > 0$ input bits and 1 output bit, and each term in E_l is specified with k bits of precision. Consider the thermal ensemble generated by this energy function at inverse temperature β. We promise that either there exists input bits $i_1, i_2, \ldots, i_j$ and output bit t such that for the reduced probability distribution on these bits

$$Pr(i_1, i_2, \ldots, i_j, t = 1) > \frac{2}{3} \tag{2.93}$$

or for any input bits $i_1, i_2, \ldots, i_j$,

$$Pr(i_1, i_2, \ldots, i_j, t = 1) < \frac{1}{3} \tag{2.94}$$

The problem **BEV** is to determine whether the first of these conditions, Eq. (2.93), holds. In other words yes instances of this problem satisfy Eq. (2.93) for some inputs $i_1, \ldots, i_j$ and no instances satisfy Eq. (2.94) for all $i_1, \ldots, i_j$.

We claim that **BEV** is **Promise-**MA-complete. The proof of this is fairly straightforward given our prior discussion of the Cook-Levin theorem.

First we claim that **BEV** is in **Promise-**MA. To show this we must show that there exists an interactive proof of the form described above for the problem **BEV**. To see this note that a computationally unbounded Merlin can sample from the thermal ensemble for an instance of this problem and can check whether there exists an input $i_1, \ldots, i_j$ such that Eq. (2.93) holds or whether for all inputs $i_1, \ldots, i_j$, Eq. (2.94) holds. This will then serve as the proof, w, in the definition of MA. Given the description of β and the circuit energy function, Arthur can construct a probabilistic Turing machine which implements the circuit related to this energy function using our construction of a probabilistic circuit from Section 2.3. Then Arthur can run this probabilistic Turing machine on the proof that Merlin supplies. It is clear that if x is a yes instance of **BEV**, then the proof w Merlin supplies will convince Arthur that he has a yes instance of this interactive protocol. Further if x is a no instance of **BEV** it is also clear that no proof that Merlin supplies will convince Arthur that x is a yes instance. Thus it is clear that **BEV** is in **Promise-**MA.

The other direction of the proof requires that we show that every problem in **Promise-**MA can be converted into a **BEV** problem. To do this we follow the idea of the Cook-Levin theorem in that, for a MA problem $x \in L_{yes}$, we convert the probabilistic Turing machine V that verifies the proof w of the MA problem into a thermal circuit for which the associated **BEV** problem has (x, w) as the satisfying input. Similarly, for an MA problem $x \in L_{no}$, the thermal circuit corresponding to V must correspond to a no instance of **BEV**, since otherwise there would exist inputs $w = i_1, i_2, \ldots, i_j$ which would cause $V(x, w)$ to accept with probability greater than $1/3$. Clearly to do this we will make use of our probabilistic circuit representation result from Section 2.3. The main tool that we need in such a construction is that we need to be able to construct circuit energy functions which perform probabilistic gates of a form we desire, for a fixed inverse temperature β. To do this we can proceed as follows. We can use our fault-tolerant constructions to produce circuits which operate with near deterministic behavior. Then in order to augment a deterministic computation we

need a source of randomness, indeed in most standard definitions of a probabilistic Turing machine one requires randomness which is close to uniformly random across the random bits being used (or in other words one requires bits which are nearly equal probability of being in 0 or 1.) To obtain such randomness in our constructions, notice that if we do not force our inputs to be 0 or 1 then there is, in a thermal ensemble, an equal probability to be 0 and 1. This allows one to create random bits in an circuit energy function. However in order to get this to work with the deterministic part of our Turing machine, we must find a method to get this randomness into *encoded* bits in our circuits. This can be accomplished by simply taking the equal probability 0 and 1 bits and running them through the circuit which performs error correction on the bundle corresponding to these bits. Thus we see that we can construct at finite β and circuit energy function which is near deterministic and which can also have input bits which are nearly uniformly random. Thus we can proceed just as in the Cook-Levin theorem. For a tableau describing the history of the probabilistic Turing machine, V, (when this probablistic Turing machine is simply a deterministic Turing machine aided by bits of randomness, we can construct an energy circuit function which implements the probabilistic Turing machine spatially across this tableau.

Thus we have shown that **BEV** is **Promise-**MA-complete. While the proof of this result was rather straightforward and followed quickly from understanding the Cook-Levin theorem it is interesting to note that very few MA-complete problems are known. In fact the only other non-natural problem which is **Promise-**MA-complete to our knowledge is the stoquastic 6-SAT problem [BDOT08]. The nearest comparable result is the classic result for the complexity of finding the ground state and computing the partition function for the Ising model of Barahona [Bar82].

2.7 Quantum Models

Finally we would like to briefly mention the quantum models which show some similarity with our classical model. In particular these models were the inspiration for investigating this problem and thus this serves as a good location to introduce the open problems concerning

these problem.

One of the quantum models relevant to this discussion are universal adiabatic quantum computing schemes. In these models one shows how to adiabatically drag a many-body quantum system from one easily preparable ground state to the ground state of another many-body quantum system whose ground state is a superposition over the history of a quantum circuit. Thus, similar to our models, the end result is a ground state which encodes a computation: but in this case the computation is a superposition over the history of the computation. These models show that adiabatic quantum computing, which previously had been only used for optimization problems [FGG+01] can also be used for universal quantum computation [KR06, OT08, BL08]. However the final state of these models are systems with small energy gaps and thus the effects of working at non-zero temperature will have a profound effect on these models. Indeed it is for exactly this reason that such universal adiabatic quantum computing schemes are not known to be fault-tolerant (see, however, [Lid08].) Similar reasoning follows for an earlier model of ground state quantum computing due to Mizel and co-workers [Miz04, MMC02].

The problem of having a system whose ground state correctly computes but whose excited states do not is exactly the problem we have addressed in this paper for the classical ground state spin computing. An interesting question then, when considering the quantum models, is whether there is a similar reinterpretation quantum thermal ensembles as spatially enacted quantum computations. This is an important open problem which might conceivably lead to fault-tolerant methods for adiabatic quantum computing.

2.8 Conclusion

We have introduce a set of energy functions on many bits (spins) which have the property that their ground state can be thought of as a spatially distributed deterministic computation. At temperature greater than zero we have shown that the thermal ensemble arising from these models can be reinterpreted as a spatially distributed probabilistic computation. Further, above zero temperature we see that the gates of a ground state spin computer can become

unreliable and fail to execute the desired computation with high-fidelity. However with the mapping of the thermal ensemble to probabilistic circuits we have shown how it is possible to make versions of the desired deterministic circuits which are fault-tolerant. We have shown that determining whether a given classical energy function can be thought of as enacting a ground state computation is NP-complete. Finally we have shown that a problem concerning the thermal ensembles arising in our model give rise to a rare complete problem for the complexity class **Promise-**MA. The models we have considered here are mostly devoid of connections to actual physical systems of interest, besides connections to quantum-dot cellular automata. An interesting an important open question about these models is whether they arise in naturally occurring physical systems, or a suitably engineered system. Another interesting question is the rate at which the ground state spin model thermalizes: a proof that the system reaches thermal equilibrium in a time polynomial in the size of the circuit being implement would be another step towards making this model more physically relevant.

Chapter 3

DIFFERENT STRATEGIES FOR QUANTUM ADIABATIC OPTIMIZATION

In this chapter, which is based on [CFL+14], we present the results of a numerical study of the performance of the Quantum Adiabatic Algorithm on randomly generated instances of MAX 2-SAT with a unique assignment that maximizes the number of satisfied clauses. We sidestep the usual question of determining how the run time T needed to achieve a certain success probability scales with the instant size n, and instead we work at a fixed bit number, $n = 20$, and look at strategies for improving the success probability for hard instances at this number of bits. In particular, we explore strategies that do not necessarily require a run time $T > \mathcal{O}(g_{min}^{-2})$ as in section 1.3.2.

We generated over 200,000 instances of MAX 2-SAT on 20 bits with a unique optimal satisfying assignment, and selected the subset for which the numerically exact time-simulation of QAA governed by equations 1.13 through 1.17 finds success probabilities of less than 10^{-4} at $T = 100$. In the following sections we describe three strategies which increase the success probability for all of these instances. First, we found that evolving along the Hamiltonian path more rapidly at a particular time scale increases the success probability for all 137 instances. Second, initializing the system in a random first excited state of the transverse-field Hamiltonian H_B at the start of the evolution leads to an average success probability close to 0.05 for the majority of hard instances, saturating the bound that arises from probability conservation. Finally, we show that adding a random local Hamiltonian to the middle of the adiabatic path often increased the success probability, and that on average this strategy increased the success probability for all of our hard instances.

3.1 Instance Selection

We sought to accumulate an ensemble of instances of MAX 2-SAT on $n = 20$ bits that are hard for the QAA as introduced in [FGGS00] and recalled in section 1.3.2. Our instances are constructed by randomly generating 60 distinct clauses, each involving two distinct bits, and retaining the instance only if there is a unique assignment w that minimizes the number of violated clauses. We keep only those instances that have a unique minimal assignment because degenerate ground states of H_P make the energy gap zero, and because we wish to avoid the complication of having success probabilities depend on the number of optimal solutions.

We generated $202,078$ instances and selected all those having a low success probability at $T = 100$, using $P(100) < 10^{-4}$ as our cutoff, resulting in a collection of 137 hard instances. To speed up the search for these instances, we used a mean-field algorithm to approximate the QAA in equations 1.13 through 1.16 with $T = 100$, and then we discarded the instances that had a final mean-field energy of 0.5 or less above the energy of the optimal assignment. We checked that instances that are easy for the mean-field algorithm would also have a high success probability under the full Schrodinger evolution by sampling a separate population of $15,000$ instances, and found that whenever the mean-field algorithm produced a final energy less than 0.5 above the ground state energy the instance had success probability $P(100) > 0.2$ according to the Schrodinger evolution. The use of this filter allowed us to discard $3/4$ of the initial $202,078$ instances, and for the remainder we numerically integrated the Schrodinger evolution with $T = 100$.

The success probabilities at $T = 100$ for the test population of $15,000$ instances are given in figure 3.1. Most of the instances we generate have high success probability at $T = 100$ (in fact, over half of this population had $P(100) > 0.95$), and hard instances at this time scale and number of bits are rare. This is why we needed to generate roughly $200,000$ total instances, and search through them using over 20,000 hours of CPU time, to obtain our ensemble of 137 instances which have $P(100) < 10^{-4}$.

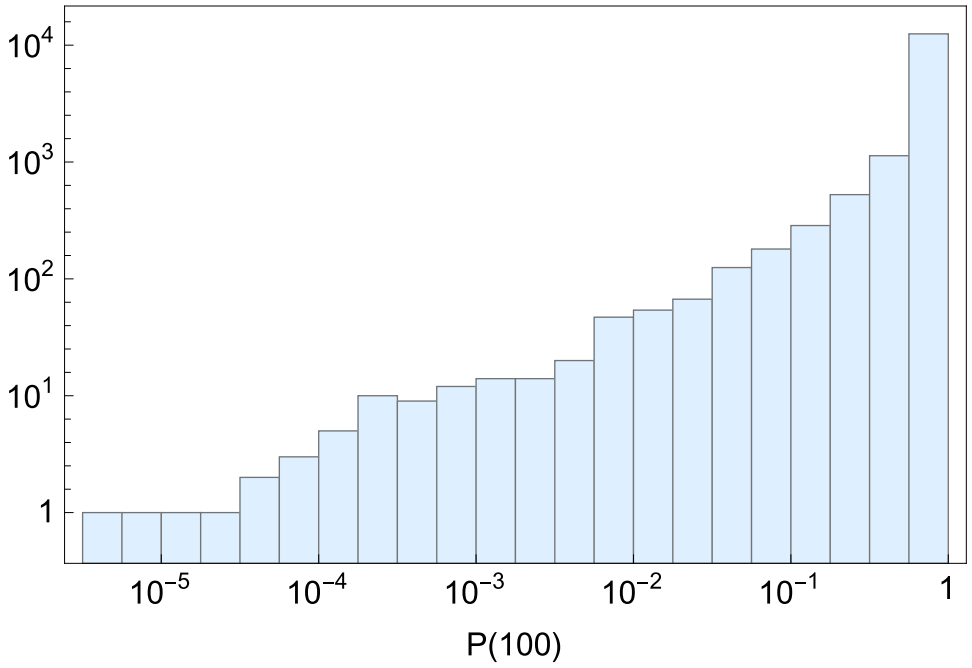


Figure 3.1: The distribution of success probabilities for 15000 instances.

3.2 The Hare Beats the Tortoise

Considering the success probability $P(T)$ as a function of the total evolution time T, we find that all of our instances with low success probability at $T = 100$ exhibit higher success probability at lower values of T. Figure 3.2 depicts this phenomenon for a single hard instance, which happened to be the first instance we carefully examined. We will refer to this instance as instance #1. We see a distinct peak of success probability at $T_{\max} = 12$ with $P(T_{max}) = 0.05$, which is to be compared with $P(100) = 5 \times 10^{-5}$ and $P(200) = 5 \times 10^{-6}$.

In figure 3.3 we plot the three lowest energy levels of instance #1, and we see a small energy gap which corresponds to an avoided crossing near $s = 0.66$. To see why changing the Hamiltonian more rapidly increases the success probability, figure 3.4 gives the instantaneous expectation of the energy, $\langle\psi(t)|H(t)|\psi(t)\rangle$, as a function of t for $T = 10$ and $T = 100$, together with the three lowest energy eigenvalues. We see that when the Hamiltonian is

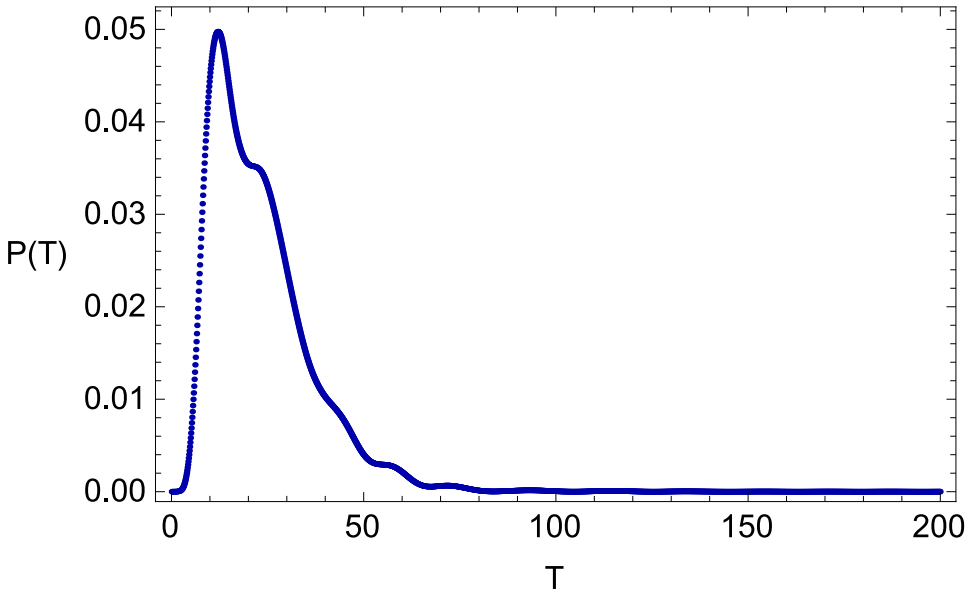


Figure 3.2: The success probability as a function of total evolution time T for instance #1.

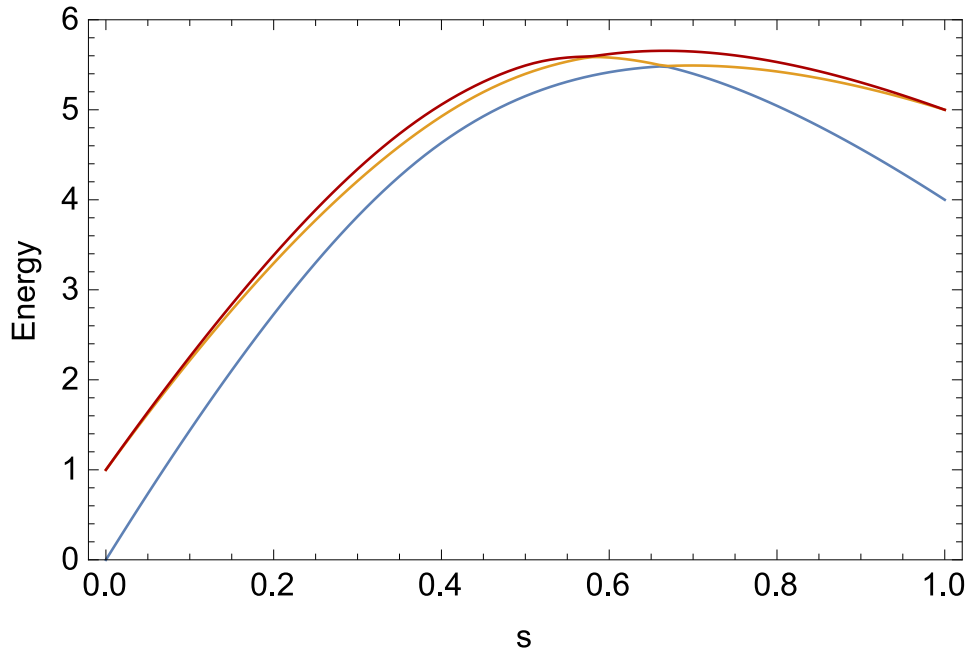


Figure 3.3: The lowest three energy levels for instance #1.

changed slowly, the $T = 100$ case, the system remains close to the ground state for all time $t < 0.66T$, but then switches to closely following the first excited state after the avoided crossing, and arrives with most of its amplitude in the first excited state subspace of H_P with virtually no overlap with $|w\rangle$.

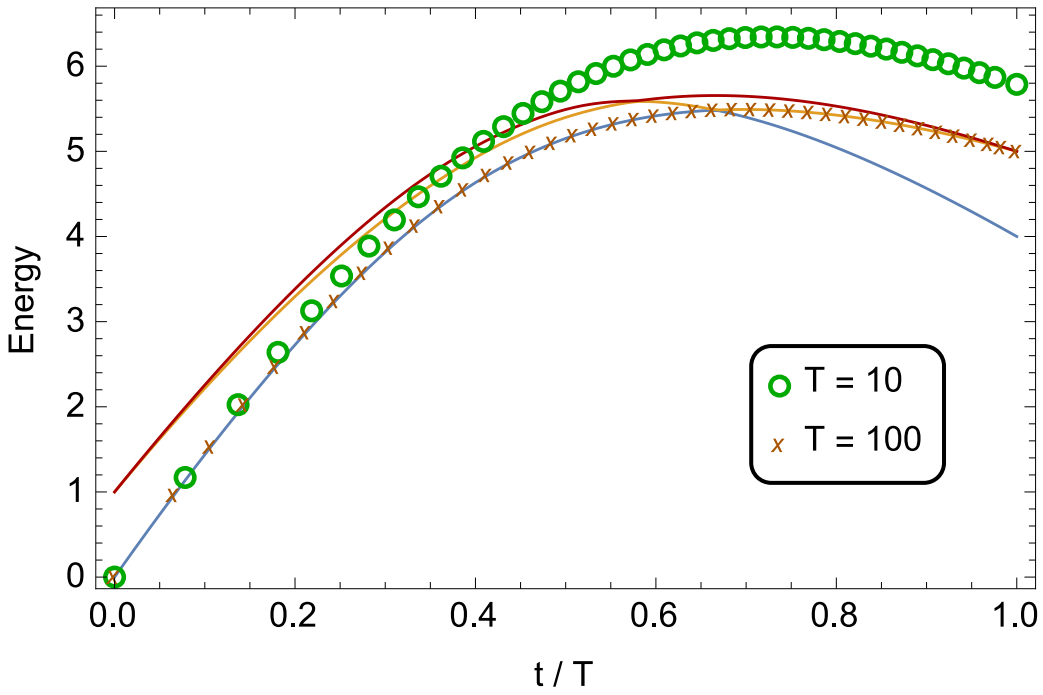


Figure 3.4: The lowest three energy levels for instance #1, superimposed with the instantaneous expectation of the energy as a function of t for $T = 10$ and $T = 100$.

In figure 3.5 we track the overlap of the rapidly evolved system ($T = 10$) with the lowest two energy eigenstates of $H(t)$, and see that the overlap of the system with the ground state immediately after the crossing corresponds to the overlap with the first excited state immediately before it. When evolving more rapidly, leaking substantial amplitude into the first excited state prior to the crossing is responsible for the increased probability of finding the system in the ground state at the end of the evolution.

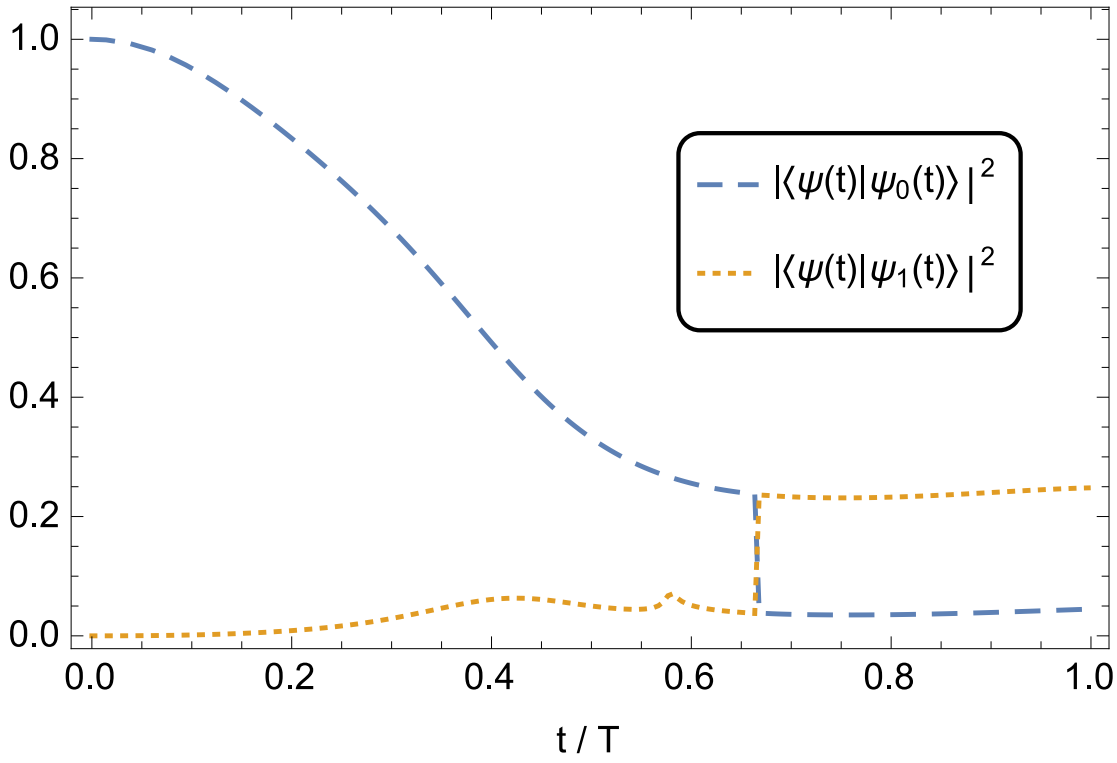


Figure 3.5: The overlap of the rapidly evolved system ($T = 10$) with the lowest two instantaneous energy eigenstates of $H(t)$, labeled here as $|\psi_0(t)\rangle$ and $|\psi_1(t)\rangle$. The bump in the overlap with the first excited state near $s = 0.58$ coincides with the avoided crossing between levels 2 and 3, as seen in figure 3.3.

Having described this phenomenon for a single instance, we now present evidence that it generalizes to many other hard instances. For each of our 137 hard instances we determined the location $T_{\max}$ where the success probability is maximized in the interval $[0, 40]$, and in figure 3.6 we compare the success probability at $T_{\max}$ with the success probability at $T = 100$. It is notable that every data point appears to the right of the 45° line, indicating that every one of our instances was improved by evolving the Hamiltonian more rapidly. The minimum improvement $P(T_{\max})/P(100)$ for this batch of instances is 108, and the median improvement is 809.

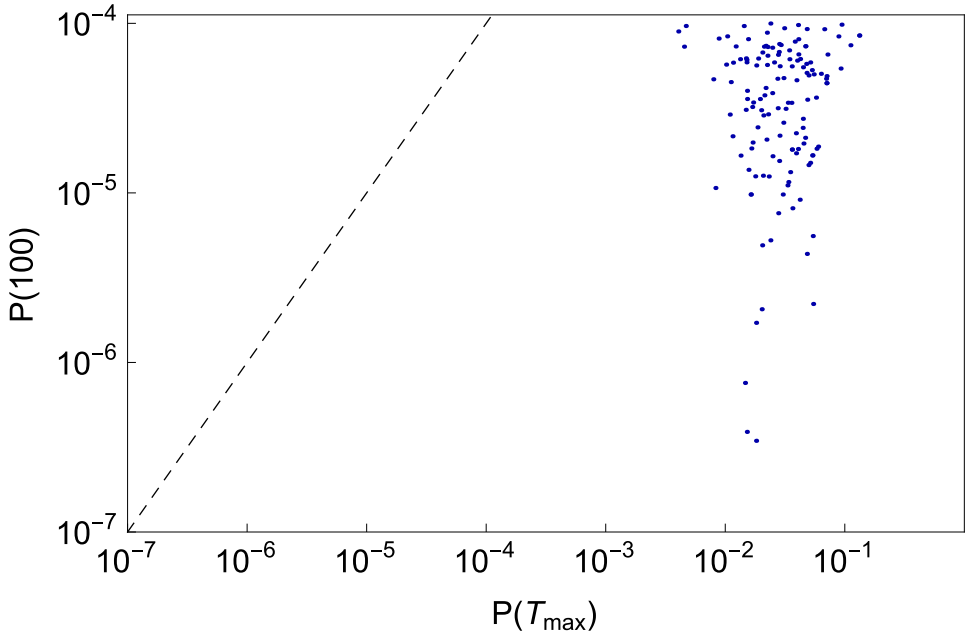


Figure 3.6: A log-log scatter plot comparing $P(T_{\max})$ with $P(100)$, where the value of $T_{\max}$ depends on the instance.

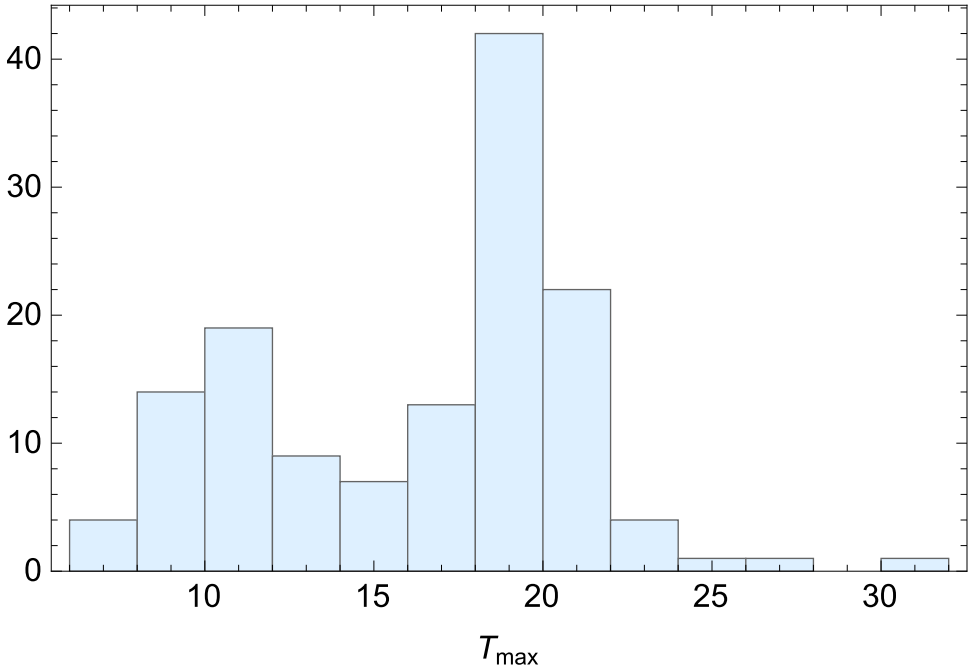


Figure 3.7: The distribution of the times T_{max} at which the success probabilities of our hard instances are maximized in the interval $[0, 40]$.

From an algorithmic perspective, it may not be possible to efficiently estimate the value of $T_{\max}$ for each instance in advance. Fortunately, the phenomenon we are describing is sufficiently robust that we can choose a fixed short time such as $T = 10$ and still gain a substantial improvement for every instance. In figure 3.8 we compare the success probabilities at $T = 10$ and $T = 100$. Here the minimum improvement $P(10)/P(100)$ is 15, and the median improvement is 574.

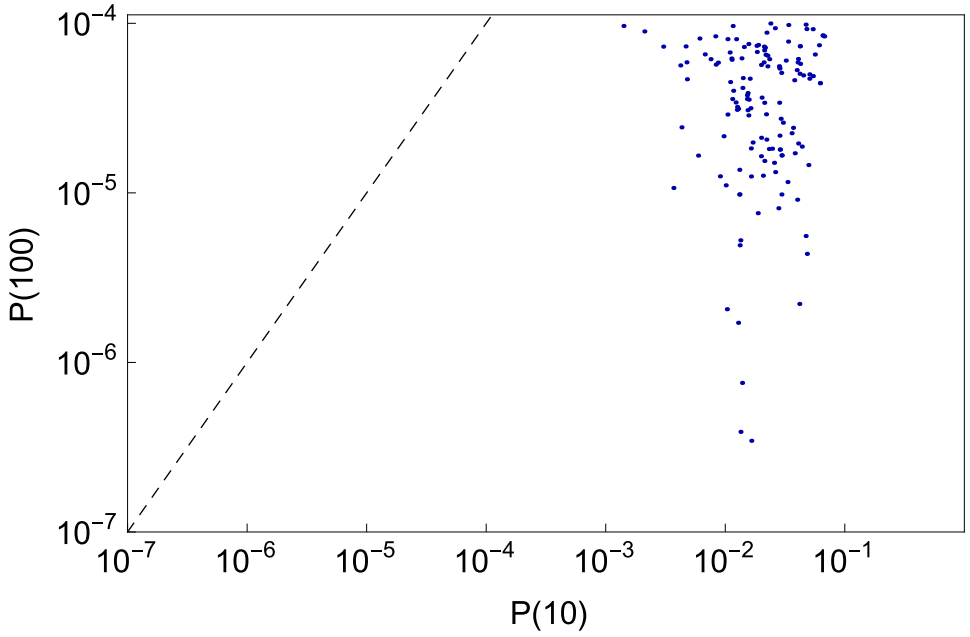


Figure 3.8: A log-log scatter plot comparing $P(10)$ with $P(100)$.

3.3 Going Lower by Aiming Higher

In the previous section we saw that having a substantial overlap with the first excited state before the avoided crossing increases the overlap with the ground state at the end of the evolution. In this section we attempt to directly exploit this effect by preparing the system at $t = 0$ to be in one of the 20 first excited states of H_B, obtained by taking the ground state (in equation 1.15) and flipping one of its qubits from $(|0\rangle + |1\rangle)/\sqrt{2}$ to $(|0\rangle - |1\rangle)/\sqrt{2}$. We did this for each of the 20 first excited states for each of our 137 hard instances. For each

instance the average success probability over the 20 excited states of H_B is given in figure 3.9, and the maximum success probability for every instance is given in figure 3.10.

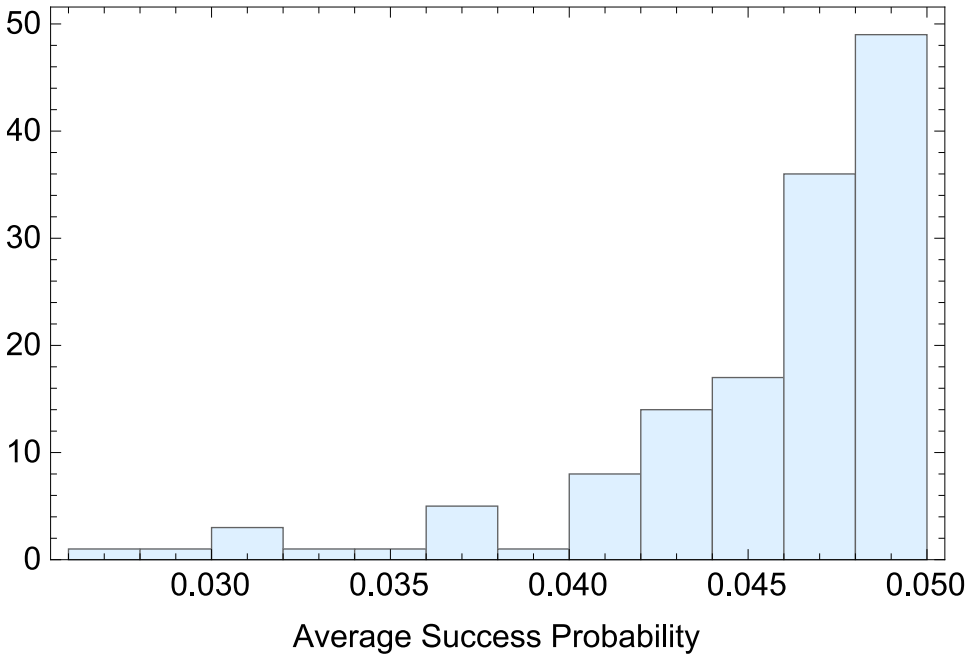


Figure 3.9: The average success probability at $T = 100$ for 137 instances obtained by initializing the system in each of the 20 first excited states of H_B.

As shown in figure 3.9, this strategy produces an average success probability near $1/20$ for most of our 137 instances. This saturates the upper bound given by probability conservation, since the sum of the success probabilities associated with the 20 orthonormal initial states cannot exceed 1.

A similar strategy to ours was used in [NSK12] to overcome an exponentially small gap in a particular Hamiltonian construction by initializing the system in a random low energy state. The authors argue that this technique is useful whenever there are a small number of low lying excited states that are separated from the remaining space by a large energy gap. The possibility of using non-adiabatic effects to drive a system from its ground state on one side of a phase transition to its ground state on the other side was considered in [CCF+11]

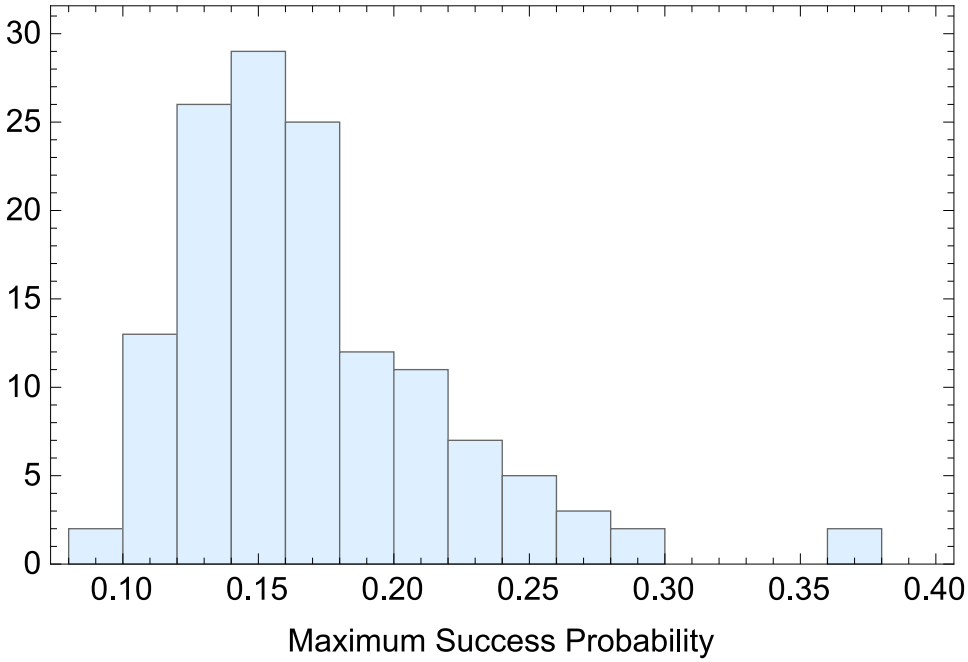


Figure 3.10: The maximum success probability at $T = 100$ for 137 instances obtained by initializing the system in each of the 20 first excited states of H_B.

as a problem in quantum control theory. Here we quantify the viability of this strategy for particularly hard instances of MAX 2-SAT at 20 bits.

3.4 The Meandering Path May Be Faster

The traditional time-dependent Hamiltonian in equation 1.16 represents a path in Hamiltonian space which is a straight line between H_B and H_P. Here, as was previously considered in [FGG02], we modify this path by adding an extra randomly chosen Hamiltonian H_E,

$$H(t) = \left(1 - \frac{t}{T}\right) H_B + \frac{t}{T}\left(1 - \frac{t}{T}\right) H_E + \frac{t}{T} H_P. \tag{3.1}$$

A reasonable constraint on H_E is that it be a sum of local terms with the same interaction graph as the problem Hamiltonian H_P, but should not use any other information specific to the particular instance. We consider three categories of H_E:

1. **Stoquastic** with zeroes on the diagonal. Each 2-local term of H_E is a linear combination of 1 and 2-qubit Pauli operators from the set $\{I\sigma^x,\, \sigma^x I,\, \sigma^z\sigma^x,\, \sigma^x\sigma^z,\, \sigma^x\sigma^x,\, \sigma^y\sigma^y\}$. For each 2-local term, the 6 real coefficients are sampled from a Gaussian distribution with mean zero, and are then normalized so that their squares sum to 1. Moreover, the coefficients are kept only if the local Hamiltonian term constructed in this way is stoquastic (i.e. all of the off-diagonal matrix elements are real and non-positive).

2. **Complex** with zeroes on the diagonal. Each 2-local term of H_E is a linear combination of 1 and 2-qubit Pauli operators chosen from the set $\{I\sigma^x,\, \sigma^x I,\, I\sigma^y,\, \sigma^y I,\, \sigma^z\sigma^x\ ,\, \sigma^x\sigma^z,\, \sigma^x\sigma^x,\, \sigma^y\sigma^y,\, \sigma^z\sigma^y,\, \sigma^y\sigma^z,\, \sigma^y\sigma^x,\, \sigma^x\sigma^y\}$. For each 2-local term, the 12 real coefficients are sampled from a Gaussian distribution with mean zero, and are then normalized so that their squares sum to 1.

3. **Diagonal**. Each 2-local term of H_E is a linear combination of 1 and 2-qubit Pauli operators chosen from the set $\{I\sigma^z, \sigma^z I, \sigma^z\sigma^z\}$. For each 2-local term, the 3 real coefficients are sampled from a Gaussian distribution with mean zero, and are normalized so that their squares sum to 1.

The reason that we work with zero diagonal H_E in the first two categories is to be sure that we are exploring purely quantum strategies for increasing the success probability, since the diagonal elements of H_E could be seen as time-dependent classical modifications to the energy landscape of H_P. The reason that we separate stoquastic path change from general complex path change is that ground states of stoquastic Hamiltonians have various special properties which may limit their computational power. Ground state local Hamiltonian problems are known to have lower computational complexity when the Hamiltonians are restricted to be stoquastic [BDOT08, BBT06]. Moreover, ground state properties of stoquastic Hamiltonians can be determined using Quantum Monte Carlo (a collection of classical methods for finding properties of quantum systems) at system sizes of up to a few hundred qubits (for a general review see [**?**], for an application to the QAA see [FGH$^+$12]).

Non-stoquastic Hamiltonians have a "sign problem" that prevents this, and we know of no efficient simulation techniques for non-stoquastic Hamiltonians at system sizes of more than roughly 20 qubits. The traditional QAA Hamiltonian defined by equations 1.13, 1.14, and 1.16 is stoquastic, and we are interested in seeing whether non-stoquastic path change can increase the computational power of this algorithm.

As a first demonstration of the potential for path change to increase success probabilities, we return to instance #1 which had $P(100) = 5 \times 10^{-5}$. In figure 3.11 we plot the spectrum for this instance with a particularly successful choice of complex H_E, and see that the avoided crossings in figure 3.3 have been eliminated.

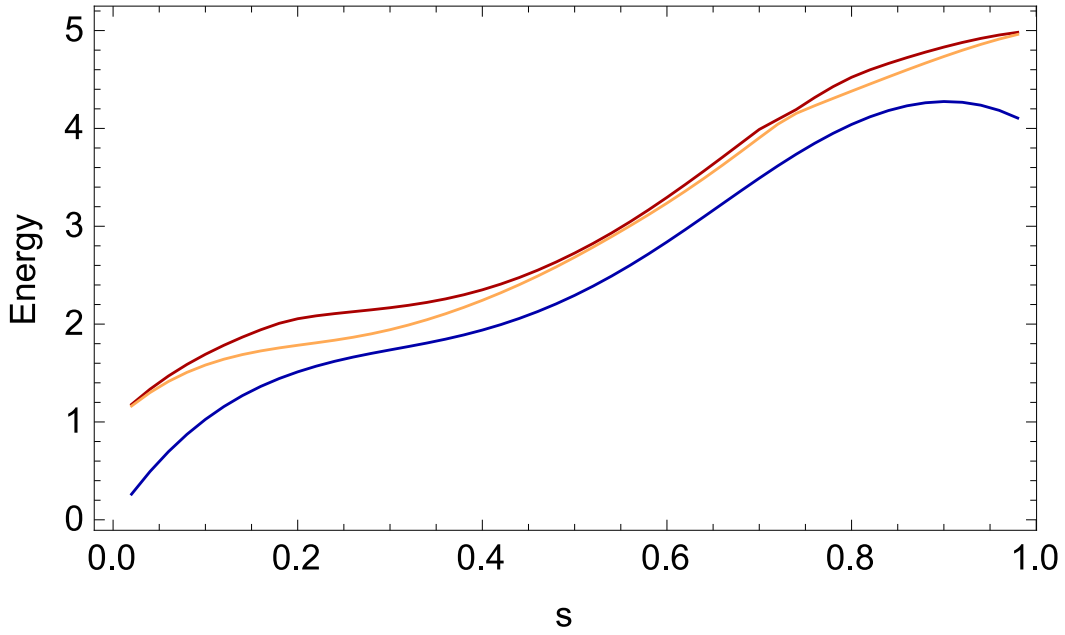


Figure 3.11: The energy spectrum of instance #1 with a particular choice of complex H_E which gives $P(100) = 0.91$.

We tested the performance of this strategy by simulating 25 trials of stoquastic, complex, and diagonal path change for each of our 137 hard instances (which all have $P(100) < 10^{-4}$ when $H_E = 0$). The path changes are all chosen independently so that there are no correlations between the instances. Simulating these path change trials for all of our instances

required over 25,000 hours of CPU time. The full distribution of success probabilities we obtained at $T = 100$ is given in figure 3.12.

While stoquastic path changes almost always increase the success probability above 10^{-4}, we see that they rarely produce success probabilities near 1. This is shown in the distribution of the maximum success probability we obtained for each instance with 25 trials of path change, shown in figure 3.13.

To take into account the spread in the distribution we estimate the geometric mean of the failure probabilities obtained by many trials of path change. For each instance we use the 25 trials to compute

$$\chi = \left(\prod_{i=1}^{25} \text{failure probability of the } i\text{-th trial} \right)^{1/25} . \tag{3.2}$$

We take $1 - \chi$ to be the effective success probability of a single trial of the path change strategy. In figure 3.14 we give the distributions of $1 - \chi$ for our ensemble of 137 hard instances.

We find that all three types of path change increase the effective success probability for all 137 of our hard instances, with complex path change typically producing the largest increase in the effective success probability.

To check whether the widening of the spectral gap seen in figure 3.11 also occurs for other successful trials of path change, we computed the minimum spectral gap g_{min} between the ground state and the first excited state for a subset of our path change trials. For each of our 137 hard instances we computed g_{min} for the most successful path change trial of each of the three types, and also for a randomly selected trial of each of the three types. Figure 3.15 compares these minimum gaps to the corresponding success probabilities.

We see a correlation between high success probability and large gaps, and almost no data with large gaps and low success probabilities.

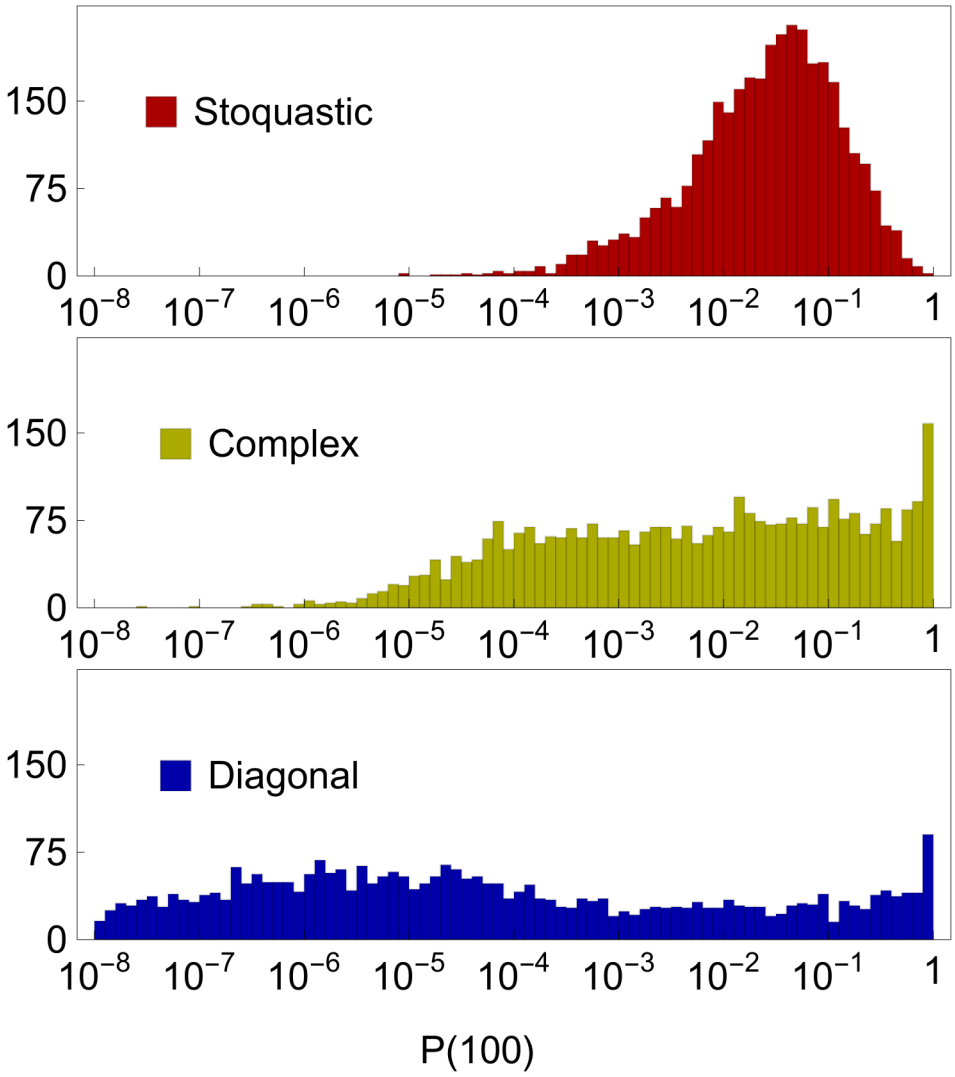


Figure 3.12: The distribution of success probabilities for 137 hard instances, when each is run with 25 randomly sampled path changes.

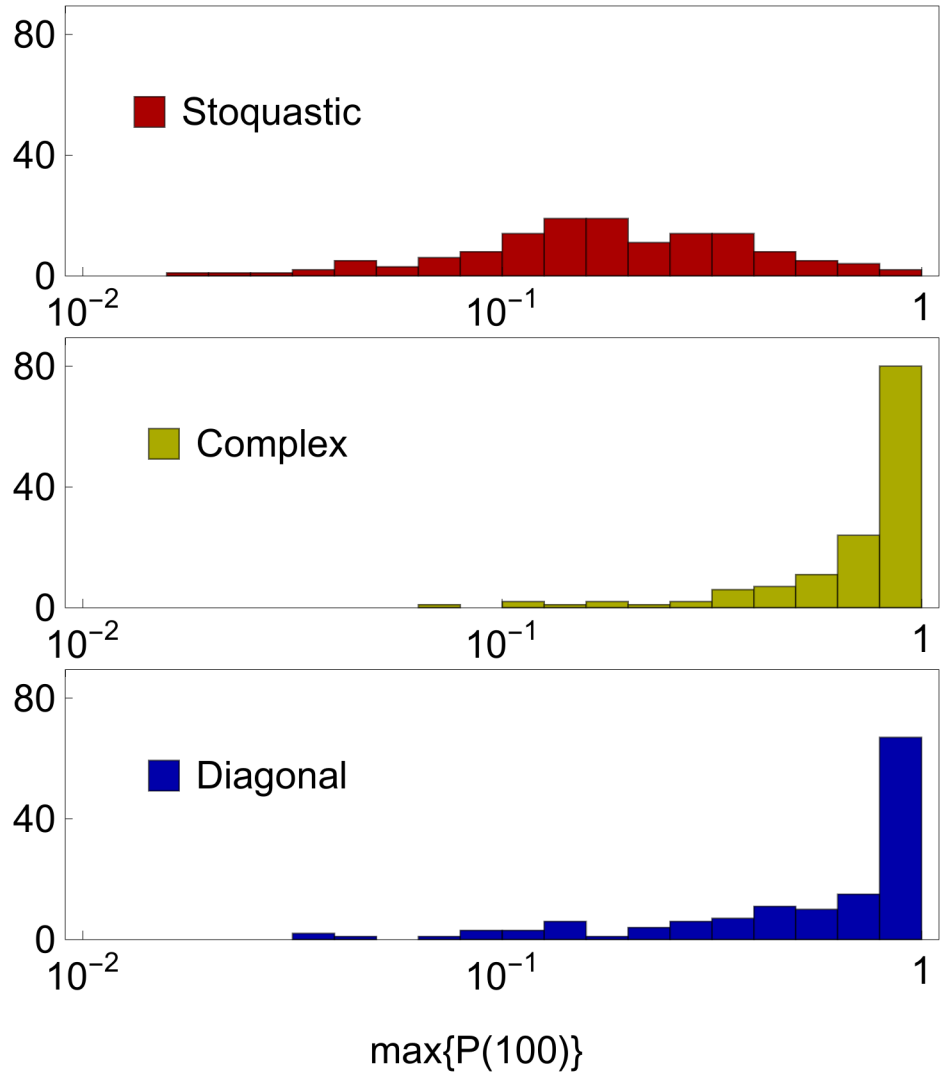


Figure 3.13: The maximum success probabilities for each of the 137 hard instances, when each is run with 25 randomly sampled path changes.

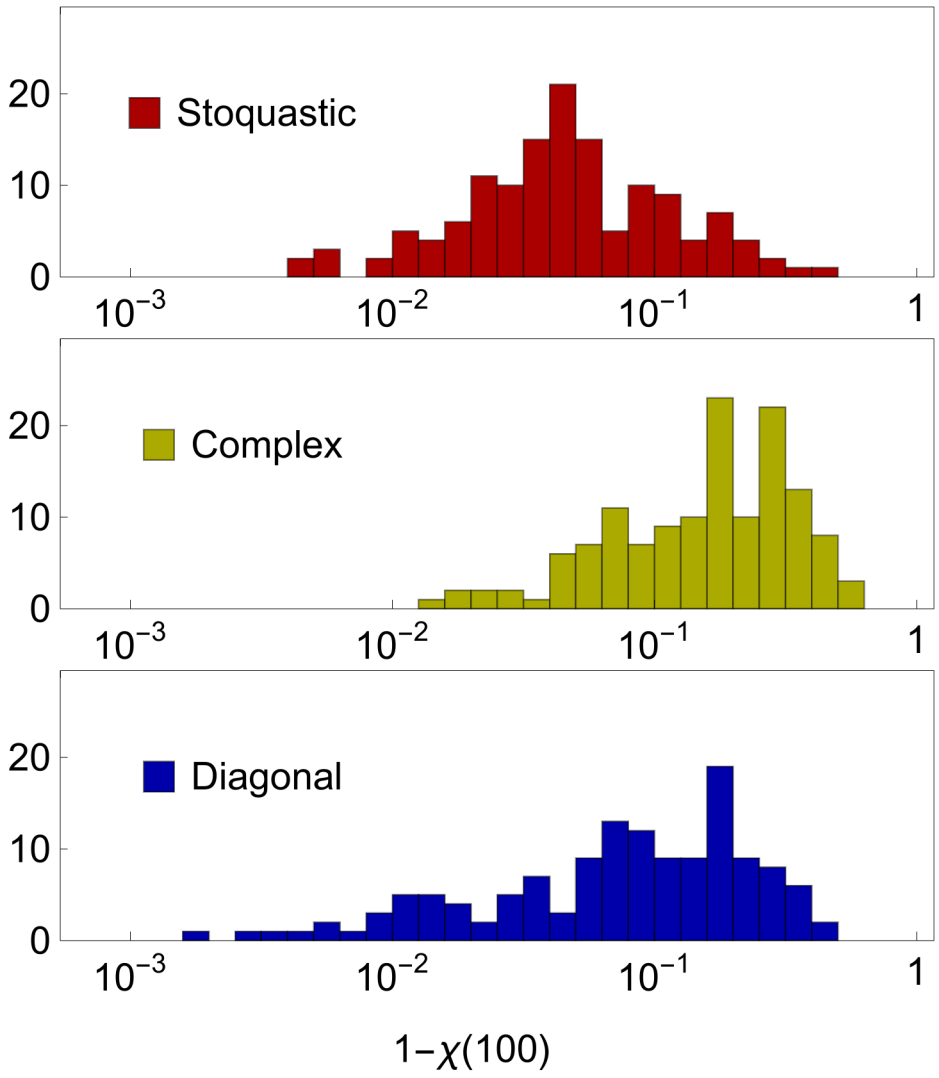


Figure 3.14: The effective success probabilities (given by 1 minus the geometric mean of the failure probabilities) obtained by running each of the 137 hard instances with 25 randomly sampled path changes.

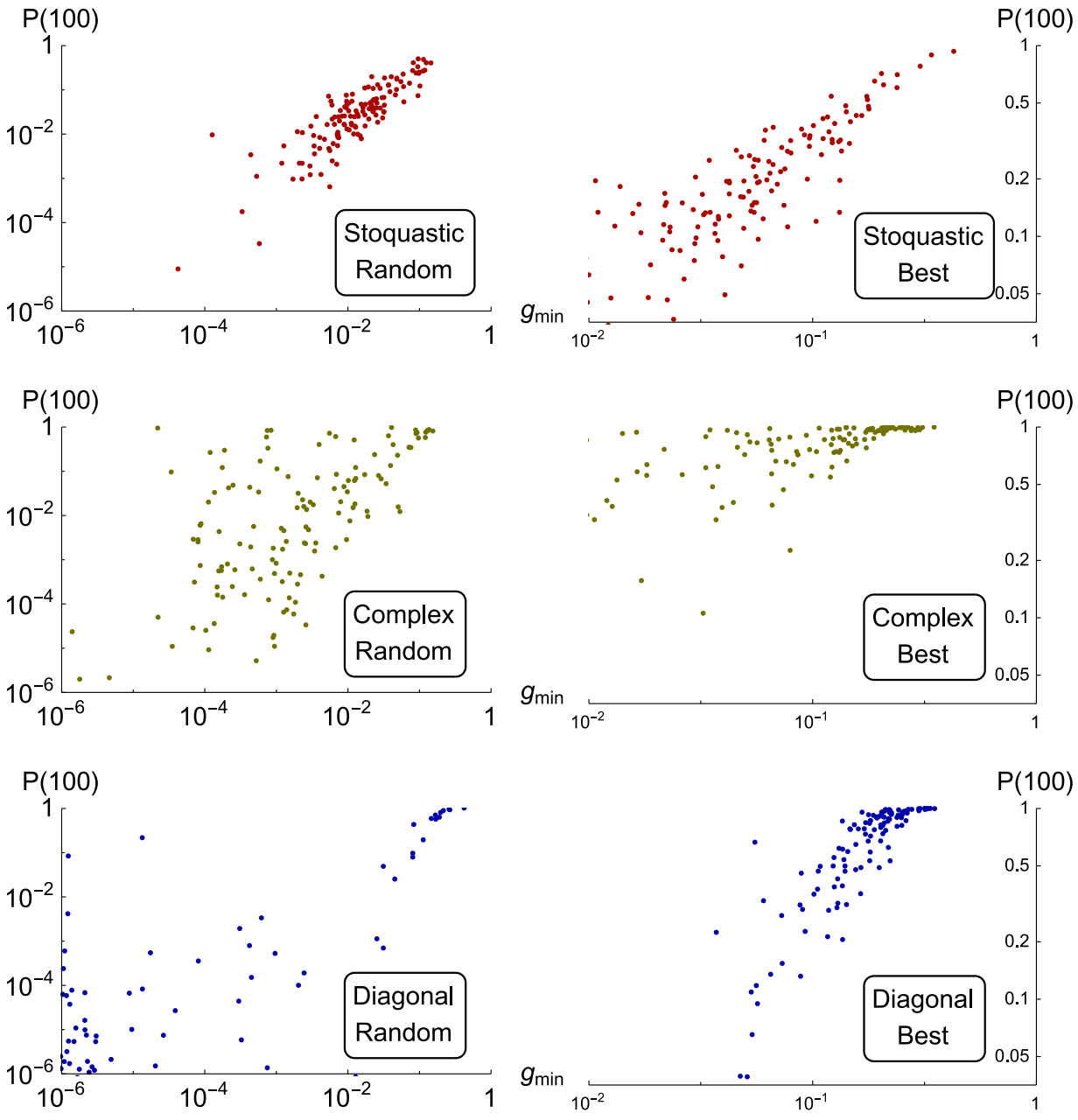


Figure 3.15: A comparison of success probabilities with the minimum spectral gap for several trials of path change. The plots in the left column contain one random path change trial for each instance. In the right column we plot the most successful path change trial for each instance. Note that the scales for the probabilities and the minimum gaps are different between the left column (random) and the right column (best).

3.5 Conclusion

The fact that our strategies improve the success probability for all 137 hard instances may be a consequence of those instances having the most room for improvement. The majority of instances we generated at 20 bits are far easier than the ones we selected. It may be the case that at higher bit number most instances have very low success probability when the traditional QAA is run for a time that scales polynomially in the number of bits. Therefore in future work it will be important to determine whether these strategies can increase the success probability of difficult instances beyond that of typical instances.

The impact of this large computational study, which used in total over $45,000$ hours of CPU time, is to motivate further exploration of non-adiabatic strategies and alternative Hamiltonian paths in the application of the QAA to random instances of optimization. To our knowledge this is the first time that non-adiabatic strategies have been applied to a large ensemble of randomly generated instances and shown to consistently improve the success probabilities compared to the traditional QAA. We hope that one day these strategies will be tested on a quantum computer running the Quantum Adiabatic Algorithm at high bit number where classical simulations are not available.

Acknowledgements

We are very grateful to Christoph Paus for allowing us to use the CMS Tier 2 distributed computing cluster, and to William Detmold for allowing us to use his Lattice QCD cluster. We appreciate the assistance with using these resources that we received from Maxim Goncharov and Andrew Pochinsky.

Chapter 4

TUNNELING THROUGH ENERGY BARRIERS IN SIMULATED QUANTUM ANNEALING

In this chapter, which is based on [CD14], we analyze the performance of simulated quantum annealing (SQA) on an optimization problem for which simulated classical annealing (SA) is provably inefficient because of a high energy barrier. We present evidence that SQA can pass through this barrier to find the global minimum efficiently. This demonstrates the potential for SQA to inherit some of the advantages of quantum annealing (QA), since this problem has been previously shown to be efficiently solvable by quantum adiabatic optimization.

Simulated annealing (SA) is a popular classical algorithm based on a Markov chain Monte Carlo model of the thermal processes that take place in a system as it is cooled to low temperatures[KGV83]. Quantum annealing (QA) is a more recently proposed algorithm [FGGS00], intended to run on quantum hardware, which takes advantage of the tendency for quantum systems to remain in the ground state of a time-dependent Hamiltonian that transforms sufficiently slowly. A third physics-inspired optimization method is simulated quantum annealing (SQA), which uses a Quantum Monte Carlo method [San10] to sample the output distribution of a quantum annealing process on a classical computer.

One approach to understanding the strengths and weaknesses of these three optimization methods is to benchmark the performance of SA, QA, and SQA on large ensembles of random instances of NP-Hard discrete optimization problems. In [BRI$^+$14] the success probabilities of QA and SQA are found to be highly correlated across random instances of quadratic unconstrained binary optimization on the Chimera graph, while the distribution of success probabilities for SA on the same set of instances bears little resemblance to that of QA and SQA. In [MST02] SQA was found to be more efficient than SA at solving random instances

of 2D Ising spin glasses, but in [BST05] SQA performed worse than SA on hard instances of the Traveling Salesman Problem.

Rigorous comparisons of these optimization methods have also been performed on specific problem instances that can be treated analytically. In [HF13] several examples are given for which QA efficiently finds the minimum, but topological obstructions cause SQA to take exponential time to equilibrate. An example given in [FGG02] called the "Hamming weight with a spike" demonstrates that QA can be exponentially faster than SA, and in this work we examine the equilibration time of SQA for this particular instance to determine whether it inherits the quantum advantage of QA or the classical defficiency of SA.

4.1 Energy Function with a High Barrier

In order to show an exponential separation between SA and QA, the authors of [FGG02] introduce an objective function $f : \{0,1\}^n \to \mathbb{R}$ which is defined in terms of the Hamming weight h (the number of 1s in the bit string),

$$f(z) = \begin{cases} h(z) & : h(z) \neq n/4 \\ n & : h(z) = n/4 \end{cases} . \tag{4.1}$$

The function f is called the "Hamming weight with a spike." As the temperature in SA is lowered the probability of crossing the energy spike at Hamming weight $n/4$ is exponentially small in n, and so SA takes exponential time to find the true minimum of f. In contrast, the authors of [FGG02] are able to determine the scaling of the minimum energy gap of the QA Hamiltonian for this instance to be $g_{\min} = \mathcal{O}(n^{-1/2})$, which implies that QA can find the true minimum of f in polynomial time.

4.2 Methods and Results

The version of SQA we consider here is based on the path-integral quantum Monte Carlo (PI-QMC) method, which we will recall here and also explain in more detail in Section 5.3. The PI-QMC method is based on a quantum-to-classical mapping by which the QA partition

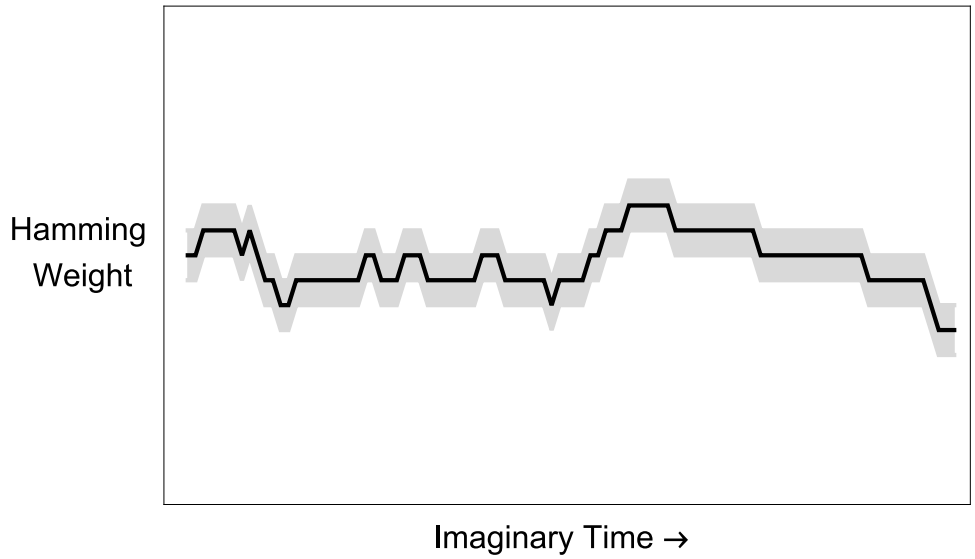


Figure 4.1: An illustration of a particular configuration $\mathbf{z} \in \Omega$ (in black) and its surrounding 1-neighborhood in the uniform norm (in gray).

function $Z_{QA} = tr(e^{-\beta H})$ at inverse temperature β is approximated by a classical partition function Z_C for an effective classical energy function E_C. The domain of E_C is the set $\Omega = \{(z_1, ..., z_L) : z_i \in \{0,1\}^n\}$ of length L sequences of n-bit strings. For a point $\mathbf{z} = (z_1, ..., z_L)$ in Ω we call the z_i "trotter slices along the imaginary-time direction", and we also use the notation $z_{i,j}$ to denote the j^{th} bit of the slice z_i. The integer parameter L determines the accuracy with which the classical partition function Z_C approximates the quantum partition function Z_{QA},

$$|Z_C - Z_{QA}| = \mathcal{O}(1/L). \tag{4.2}$$

The classical energy function is

$$\beta E_C(z_1, ..., z_L) = \sum_{i=1}^{L} \left(\frac{\beta}{L} f(z_i) + J \sum_{j=1}^{n} z_{i,j} z_{i+1,j} \right) \tag{4.3}$$

where $J = \frac{1}{2} \log \coth \left(\frac{\beta \Gamma}{L} \right)$ is the coupling strength along the imaginary-time direction. In our work we sample from the classical boltzmann distribution $\pi(\mathbf{z}) = \exp(-\beta E_C(\mathbf{z})/Z_C$ with

energy E_C using a markov chain consisting of local moves (flipping a single bit) and accepting a proposed move $\mathbf{z} \to \mathbf{z}'$ with a probability $P(\mathbf{z}, \mathbf{z}')$ given by the metropolis rule

$$P(\mathbf{z}, \mathbf{z}') = \min\left\{1, \frac{\pi(\mathbf{z}')}{\pi(\mathbf{z})}\right\} = \min\left\{1, e^{E_C(\mathbf{z}')-E_C(\mathbf{z})}\right\} \tag{4.4}$$

Most applications of the Path-Integral Monte Carlo method also implement non-local "worldline updates", which replace an entire imaginary-time trajectory $\{z_{1,k}, ..., z_{L,k}\}$ of a single qubit k in one step using a heat-bath acceptance probability, in order to speed up the convergence to the distribution π. A single worldline update changes the Hamming weight in each trotter slice of a configuration $\mathbf{z} \in \Omega$ by at most ± 1, so the set of all configurations which can be obtained after one such update is illustrated as the gray neighborhood in figure 4.1. Adding wordline updates to the dynamics speeds up the equilibration of the system, but for our interests it suffices to show that local updates equilibrate in polynomial time.

As our measure of convergence time for SQA we consider the minimum number of sweeps τ_s needed to sample the true minimum of f with a reasonably high probability. Each sweep consists of a systematic scan of single site updates that procedes through the bits within the 1^{st} trotter slice, then the 2^{nd} trotter slice, and so on.

We decreased the transverse field geometrically along a schedule $\Gamma_0, ..., \Gamma_m$ with $\Gamma_0 = 1$ and $\Gamma_{i+1} = 0.7\Gamma_i$, until reaching a final value of $\Gamma_m \approx 10^{-12}$. We consider system sizes up to $n = 1400$, and use an inverse temperature of $\beta = 32$ to ensure that the true minimum of f has overwhelmingly high probability in the stationary distribution π when $\Gamma = \Gamma_m$.

In many applications of SQA the trotter number L is taken to be a constant that is independent of the number of quantum spins n. In this work, however, we find it is necessary that L should scale at least linearly with n, so that the probability $\mathcal{O}(\exp(-\frac{\beta n}{L}))$ of accepting a bit flip which increases the number of trotter slices with Hamming weight $n/4$ is not exponentially small in n. If L is taken to be a constant then the equilibration time will become exponential at sufficiently large values of n, indicating a crossover from quantum to classical behavior.

Figure 4.2 contains the main results of this work. We find that $\tau_s \approx \mathcal{O}(n^z)$, with the

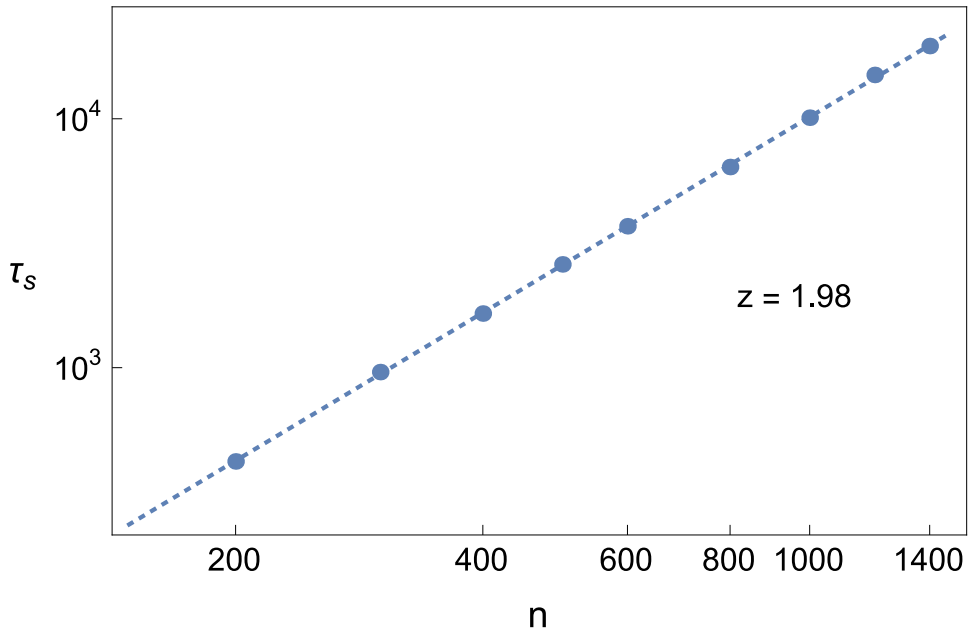


Figure 4.2: A log-log plot of the convergence time as measured by the number of sweeps τ_s (performed at each transverse field value Γ_i) vs n.

dynamical critical exponent $\mathtt{z} \approx 1.98$. Since the ferromagnetic coupling in equation **??** diverges as $J = \mathcal{O}(\Gamma^{-1})$ as $\Gamma \to 0$, we can compare this with the exact value $\mathtt{z} = 2$ for critical slowdown of the single site dynamics in the $\beta \to \infty$ phase transition of the 1D kinetic Ising model [Pri05]. This is the same scaling of the convergence time as we would find without the high barrier at Hamming weight $n/4$, since then we would have n uncoupled 1D Ising models with $L = \mathcal{O}(n)$ sites each (recall that we defined τ_s to count the number of sweeps, which accounts for the extra factor of n).

4.3 Conclusion

In this chapter we have presented evidence that SQA can be exponentially faster than SA in minimizing an energy function with a high barrier. We further find that the equilibration time of SQA on this instance is not inhibited by the barrier at all, provided that the discretization in the imaginary-time direction is sufficiently fine. Directions for future work include finding an analytic proof of this exponential separation in algorithmic performance, as well as searching for random instances of optimization problems which have similar features in their energy landscape to further understand the conditions which make SQA useful.

Chapter 5

RAPIDLY MIXING MONTE CARLO FOR 1D TRANSVERSE ISING MODELS

In this chapter we present a polynomial-time randomized classical algorithm for approximating the partition function of generalized 1D transverse Ising models (TIM) at any temperatures which is independent of the system size. The algorithm relies on the rapid mixing of a Markov chain which formalizes the path-integral quantum Monte Carlo (PI-QMC) method.

The primary challenge in using the quantum Monte Carlo (QMC) methods of section 1.4 for a provably efficient simulation algorithm is to show that the underlying random walk is rapidly mixing, i.e. that it equilibrates in polynomial time. In [BT09] a diffusion QMC method for the adiabatic evolution of frustration-free stoquastic Hamiltonians was shown to be rapidly mixing whenever the spectral gap of the Hamiltonian is lower bounded by an inverse polynomial in the system size. More recently, a path-integral QMC for ferrogmagnetic TIM on arbitrary graphs was shown to be rapid mixing [Bra14] by a direct reduction to a rapidly mixing random walk for classical ferromagnetic Ising models [JS93]. Another line of work treating the continuous-time nonequilibrium dynamics of path-integral QMC has established optimal temporal mixing (equilibration in time $\mathcal{O}(n \log n)$ for ferromagnetic transverse Ising models at high temperature [CP10], and for ferromagnetic Ising models on trees [MW11].

Generalized TIM differ from ferromagnetic TIM by allowing site-dependent real-valued couplings between neighboring spins as well as arbitrary real-valued local fields. The disorder and frustration that can arise from conflicting local terms can make these systems more difficult to analyze than the ferromagnetic and frustration-free cases described above where QMC is provably efficient. Generalized TIM are commonly considered for proposals of quan-

tum annealing because they can have ground states that encode the solution to NP hard classical problems[FGGS00]. Another justification for the focus on these models is the fact that any instance of the stoquastic local Hamiltonian problem (estimating the ground state energy) can be reduced to an instance of the local Hamiltonian problem for a generalized TIM [BH14].

The present work takes the view that the existence of an efficient simulation method can be of intrinsic interest, independent of the physical properties which one might use that simulation algorithm to compute. The path-integral QMC we analyze is closely related to the version of the algorithm that is widely used in practice, and therefore this work belongs to the tradition in Hamiltonian complexity of characterizing the complexity of sub-classes of quantum systems by providing a rigorous basis to heuristic approximation algorithms which were developed by the physics community.

5.1 Statement of results

We consider 1D transverse Ising models of the form

$$\mathcal{H} = -\Gamma \sum_{j=1}^{n} \sigma_j^x + \sum_{j=1}^{n} (K_j \sigma_j^z \sigma_{j+1}^z + b_j \sigma_j^z) \quad , \quad K_j, b_j \in [-1, 1] \text{ and } \Gamma \geq 0. \tag{5.1}$$

Note that the condition $\Gamma \geq 0$ implies that these models are "free of the sign problem", which is a necessary condition for the quantum Monte Carlo method we will apply. Our main result is an algorithm for the approximation of the partition function $\mathcal{Z} := \mathrm{tr}\left(e^{-\beta \mathcal{H}}\right)$ for models of the form (5.1) which runs in time that scales polynomially with the number of spin sites n.

Theorem 1. *For all $\beta > 0$ there is an algorithm which takes as input an n-qubit Hamiltonian H as in (5.1) and can output an estimate $\tilde{Z}_\beta$ of the partition function which satisfies*

$$|\tilde{\mathcal{Z}}_\beta - \mathcal{Z}_\beta| \leq \delta_{mult} \mathcal{Z}_\beta + \delta_{add}$$

with probability $\geq 1 - \delta_{fail}$. The algorithm runs in time $\mathrm{poly}(n, e^\beta, 1/\delta_{mult}, \log(1/\delta_{add}), \log(1/\delta_{fail}))$.

5.2 Algorithm Overview

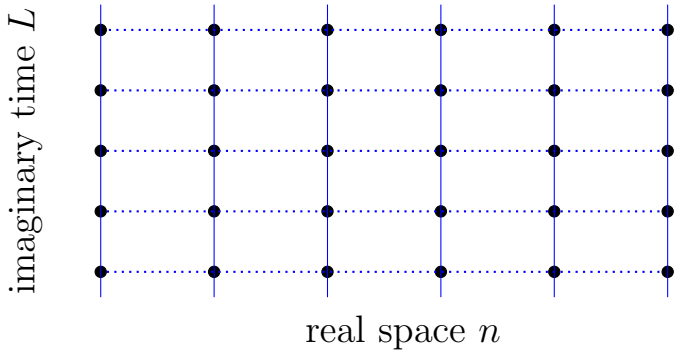
The path-integral quantum Monte Carlo (PI-QMC) method uses Suzuki's quantum-to-classical mapping [SMK77] to map a 1D qubit Hamiltonian as in (5.1) to a system of $\{\pm 1\}$ classical spins on a 2D lattice $\Lambda = \{1, ..., n\} \times \{1, ..., L\}$. The mapping introduces an energy function $E_\beta : \Omega \to \mathbb{R}^+$ on the set of classical spin configurations $\Omega = \{(z_1, ..., z_L) : z_i \in \{-1, 1\}^n\}$ that is designed to satisfy

$$e^{-E_\beta(z_1,...,z_L)} \approx \prod_{i=1}^{L} \langle z_i | e^{-\beta \mathcal{H}/L} | z_{i+1} \rangle \ , \tag{5.2}$$

which leads to relations between a Boltzmann-like distribution π on (Λ, E) and the quantum thermal density matrix ρ, as well as between the normalizing constant Z of π and the quantum partition function $\mathcal{Z}$,

$$\pi(z_1, ..., z_L) = \frac{e^{-E_\beta(z_1,...,z_L)}}{Z} \quad , \quad Z = \sum_{z_1,...,z_L} e^{-E_\beta(z_1,...,z_L)} \quad \overset{\text{dual}}{\Longleftrightarrow} \quad \rho = \frac{e^{-\beta\mathcal{H}}}{\mathcal{Z}}. \tag{5.3}$$

A derivation of this quantum-to-classical mapping is given in Section 5.3.2, but for the sake of this overview it suffices to describe the geometry and the interactions in the classical spin lattice Λ. For a configuration $z = (z_1, ..., z_L) \in \Omega$ we refer to each of the z_i as "real-space slices." The index ranging from $1, ..., L$ is commonly called the "imaginary-time direction", since $e^{-\beta\mathcal{H}}$ can be thought of as Schrödinger time evolution for an imaginary time $\tau := -i\beta$. For $j \in \{1, ..., n\}$ we will call the sequence which is formed by taking the j-th bit of each real-space slice the "world-line of the j-th qubit."



The couplings between the classical spins are related to the matrix elements of $e^{-\beta\mathcal{H}/L}$. If two neighboring real-space slices z_i, z_{i+1} are equal then the corresponding term in (5.2) is

$$\langle z_i|e^{-\beta\mathcal{H}/L}|z_i\rangle = \exp\left[-\frac{\beta}{L}\sum_{j=1}^{n}(K_j z_{i,j-1}z_{i,j} + b_i z_{i,j})\right] \tag{5.4}$$

which shows that each bond in the real-space direction contributes an amount of energy that is $\mathcal{O}(\beta/L)$. Since $L \gg \beta\|\mathcal{H}\|$, the bonds along this direction are weak when we take $L = \text{poly}(n)$ and $\beta = \mathcal{O}(\log n)$.

In contrast, when two neighboring real-space slices z_i, z_{i+1} are not equal then the contribution $\langle z_i|e^{-\beta\mathcal{H}/L}|z_{i+1}\rangle$ to the classical energy function arises from the off-diagonal matrix elements of $\mathcal{H}$. The couplings along this direction are ferromagnetic and diverging like $\log\Gamma^{-1}$ as $\Gamma \to 0$. This corresponds to a classical measure π in which $z_i = z_{i+1}$ for most i. Indeed even if $L \to \infty$ the expected number of i where $z_i \neq z_{i+1}$ will remain bounded.

To sample from π and approximate Z we use standard Markov chain Monte Carlo (MCMC) methods, which are detailed in Section 5.3.4 and briefly reviewed here in order to set the notation. We view Ω as a set of vertices of a graph, which are taken together with a set of edges $E(\Omega)$ with weights that give transition probabilities for a random walk. The transition probabilities are chosen to be reversible with respect to π (i.e. they satisfy detailed balance), and this together with a few additional assumptions ensures that the random walk will eventually converge to π.

With the above in mind, the primary question is to determine the *mixing time* τ for this Markov chain, which is the minimum number of steps that the random walk needs to be take in order to sample from a distribution that is close to π. We will use the method of canonical paths due to Jerrum and Sinclair (reviewed in [Sin92]) in order to bound the mixing time. The idea of the method is that for every pair of vertices $x, y \in \Omega$ the random walk P needs to route an amount of *traffic* equal to $\pi(x)\pi(y)$ units of probability along some path (or set of paths) which connects x to y along edges in $E(\Omega)$. Define the *congestion* of an edge e to be the ratio of the total traffic through it (i.e. $\sum_{x,y:e\in\gamma_{x,y}}\pi(x)\pi(y)$) to the amount of probability flow across it at equilibrium, which equals $P(v',v)\pi(v)$ if $e = (v,v')$.

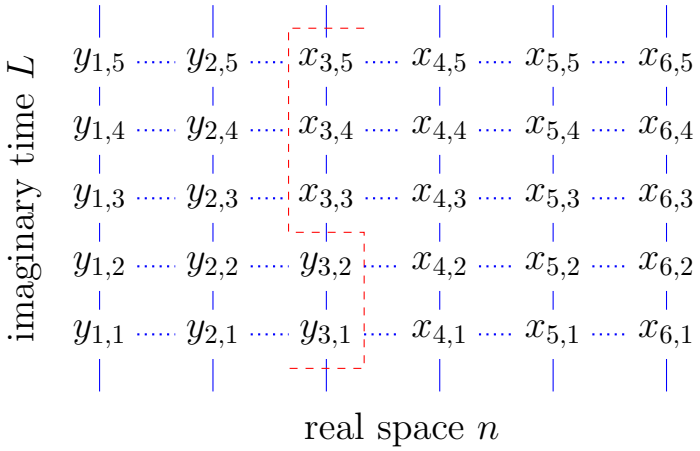


Figure 5.1: An intermediate point along a canonical path from x to y. We have finished updating qubits 1 and 2 and are in the middle of updating qubit 3. The red line indicates the boundary between the x and y configurations. In computing the energy this line is the only part which contributes terms not found in the energies of x or y.

If no edge is too congested then we can show that the chain is rapidly mixing.

Our problem now reduces to finding good sets of paths between every pair of configurations $x, y \in \Omega$. We will describe paths which replace the entries of x with entries of y one at a time. A crucial requirement is that the intermediate configurations do not have weight which is too much lower than that of x and y. Equivalently the energies of states along the path should not get too high. We will achieve this by scanning across each qubit's world-line in sequence, finishing one before starting the next.

An example of an intermediate step along this path is depicted in Fig. 5.1. The resulting configurations have energy comparable to a weighted average of the energies of x and y, except along the edges where the x sites border the y sites. These edges are depicted in Fig. 5.1 by a red line. Our path is chosen to take advantage of the particular energy function that appears in quantum Monte Carlo, and in particular the fact that horizontal bonds are weak

and vertical bonds are strong. Our path breaks at most 2 vertical bonds, each contributing energy $\mathcal{O}(\log(L/\beta\Gamma))$, and at most L horizontal bonds, each contributing energy $\mathcal{O}(1/L)$. The contribution of these broken bonds can raise the energy by at most a factor which grows logarithmically in the system size, and in section 5.4 we will see that this implies that the random walk mixes in polynomial time.

5.3 Preliminaries

5.3.1 Suzuki-Trotter Approximation

Let $\mathcal{H}$ be a TIM as in (5.1) and define the notation $A_i = -\beta\left(K_i\sigma_i^z\sigma_{i+1}^z + b_i\sigma_i^z\right)$ and $B_i = \beta\Gamma\sigma_i^x$, as well as $A = \sum_i A_i$ and $B = \sum_i B_i$. Note that $-\beta\mathcal{H} = A + B$. The idea of the Suzuki-Trotter approximation is to approximate $\mathcal{Z} = \operatorname{tr} e^{(A+B)}$ with $Z_L = \operatorname{tr}\left(e^{\frac{A}{L}}e^{\frac{B}{L}}\right)^L$. The fact that $\lim_{L\to\infty} Z_L = \mathcal{Z}$ is the content of the 1875 Lie product formula [LE75], but for the sake of a computational efficient algorithm we will consider L to be at most polynomially large in n, and for this we will make use of a lemma which follows from [Bra14].

Lemma 1. *For any error tolerance $\delta > 0$, if $L \geq \max[4\beta\|\mathcal{H}\|, \sqrt{96(\beta\|\mathcal{H}\|)^3\delta^{-1}}]$ then*

$$(1-\delta)Z_{\beta,L} \leq \mathcal{Z}_\beta \leq (1+\delta)Z_{\beta,L} \tag{5.5}$$

According to the lemma it suffices to take $L = poly(n, \delta^{-1})$ in order for the Suzuki-Trotter approximation Z_L to the partition function to be within multiplicative error δ of the true quantum partition function $\mathcal{Z}$. Henceforth we will assume L satisfies the hypothesis of Lemma 1 and will drop the subscript L from the Suzuki-Trotter approximation Z.

5.3.2 Quantum-to-Classical Mapping

Here we will describe the well-known quantum-to-classical mapping [SMK77] which underlies the PI-QMC method. The first step is to expand Z by inserting several complete sets of

states,

$$Z = \sum_{z_1} \langle z_1 | \left(e^{\frac{A}{L}} e^{\frac{B}{L}} \right)^L | z_1 \rangle \tag{5.6}$$

$$= \sum_{z_1,\ldots,z_L} \prod_{i=1}^{L} \langle z_i | e^{\frac{A}{L}} e^{BL} | z_{i+1} \rangle \quad , \quad z_{L+1} := z_1 \tag{5.7}$$

$$= \sum_{z_1,\ldots,z_L} \prod_{i=1}^{L} e^{\frac{A(z_i)}{L}} \langle z_i | e^{\frac{B}{L}} | z_{i+1} \rangle. \tag{5.8}$$

To view this as an effective classical spin system, the goal is to express the summand of (5.8) in the form $e^{-E_\beta(z_1,\ldots,z_L)}$. The matrix elements of the off-diagonal part of the Hamiltonian factorize into a product of local terms,

$$\langle z_i | e^{\frac{B}{L}} | z_{i+1} \rangle = \langle z_i | e^{\frac{\beta\Gamma}{L} \sum_{j=1}^{n} \sigma_j^x} | z_{i+1} \rangle = \prod_{j=1}^{n} \langle z_j | e^{-\frac{\beta\Gamma}{L} \sigma_j^x} | z_{j+1} \rangle \tag{5.9}$$

and so defining a local operator $e^{h_j} = \langle z_{i-1,j} | e^{\frac{-\beta\Gamma\sigma_j^x}{L}} | z_{i,j} \rangle$ leads to

$$\prod_{i=1}^{L} e^{\frac{A(z_i)}{L}(z_i)} \langle z_i | e^{\frac{B}{L}} | z_{i+1} \rangle = e^{\sum_i \frac{A(z_i)}{L} + \sum_{j=1}^{n} h_j} \tag{5.10}$$

which shows that E_β can be expressed as a sum of local terms,

$$E_\beta = \sum_{i=1}^{L} \sum_{j=1}^{n} \left[\frac{\beta}{L} K_j z_{i,j} z_{i,j+1} + b_j z_{i,j} - h_j(z_{i,j}, z_{i+1,j}) \right]. \tag{5.11}$$

The sites of the classical lattice are $\Lambda = \{1, ..., n\} \times \{1, ..., L\}$, and the state space of the classical spin system is $\Omega = \{+1, -1\}^\Lambda$. For a spin configuration $\mathbf{z} \in \Omega$ we write $\mathbf{z} = (z_1, ..., z_L)$, where each z_i is an n-bit string called a real-space slice (these are also sometimes called Trotter slices), and we write $z_{i,j}$ to refer to the j-th bit of the real-space slice z_i.

Since the steps of the quantum-to-classical mapping are exact, the partition function for the Gibbs distribution of this classical spin system is equal to the Suzuki-Trotter approximation of the quantum partition function.

5.3.3 *Estimating the partition function*

In section 5.4 we will prove that for any tolerance $\delta > 0$ the PI-QMC method can efficiently output an element of Ω sampled from a distribution $\tilde{\pi}$ which is within $|\tilde{\pi} - \pi| \leq \delta$. Here we explain how these samples can be used to approximate the partition function with a standard telescoping product technique [BSVV05]. We express the partition function $Z(\beta)$ at inverse temperature β as a product of terms involving the partition function at higher temperatures,

$$Z(\beta) = Z(0)\frac{Z(\beta_1)}{Z(\beta_0)}...\frac{Z(\beta_k)}{Z(\beta_{k-1})} \tag{5.12}$$

where the *annealing schedule* $\beta_0 < \beta_1 < ... < \beta_k$ with $\beta_0 = 0$ and $\beta_k = \beta$ is chosen so that each ratio in the telescoping product is bounded. To estimate the ratios we will use the fact that

$$\left\langle \frac{w_{\beta_i}}{w_{\beta_{i-1}}} \right\rangle_{\pi_{\beta_{i-1}}} = \sum_{\mathbf{z}\in\Omega} \frac{w_{\beta_i}(\mathbf{z})}{w_{\beta_{i-1}}(\mathbf{z})}\pi_{\beta_{i-1}}(\mathbf{z}) = \frac{Z(\beta_i)}{Z(\beta_{i-1})} \tag{5.13}$$

At infinite temperature the quantum partition function is $\mathcal{Z}(0) = 2^n$. Further, a uniform schedule with $k = \mathcal{O}(\beta\|H\|\log(\beta\|H\|))$ ensures that

$$\frac{1}{e} \leq \frac{\mathcal{Z}(\beta_i)}{\mathcal{Z}(\beta_{i-1})} \leq 1 \tag{5.14}$$

and so together with lemma 1 this implies the Suzuki-Trotter approximation to the partition function satisfies

$$e^{-(1+2\delta)} \leq \frac{Z(\beta_i)}{Z(\beta_{i-1})} \leq e^{2\delta} \tag{5.15}$$

Note that at $\beta = 0$ the classical effective system (5.11) has infinite coupling strength between spins in the imaginary-time direction, and so the Markov chain we analyze in Section 5.4 is *not* rapidly mixing (it is not even ergodic). Therefore to compute the expectation $\left\langle \frac{w_{\beta_1}}{w_{\beta_0}} \right\rangle_{\pi_{\beta_0}}$ we will use the fact that π_{β_0} is equivalent to the uniform distribution on the subset $\Omega_{\text{classical}} \subseteq \Omega$ consisting of configurations that have no flips in the imaginary-time direction.

Finally, in order to bound the number of samples needed to compute the expectation values in the telescoping product we will make use of the Hoeffding bound.

Lemma 2 ([Hoe63]). *Let $X_1, \ldots, X_t$ be independent random variables satisfying $|X_i| \leq 1$ and $\mathbb{E}[X_i] = \bar{X}$. Then*

$$\mathbb{P}[\left|\frac{1}{t}\sum_{i=1}^{t} X_i - \bar{X}\right| \geq \delta] \leq 2e^{-t\delta^2/2} \tag{5.16}$$

Define $M(z) = e^{-2\delta} w_{\beta_i}(z)/w_{\beta_{i+1}}(z)$, so that $|M(z)| \leq 1$. Observe that $|\langle M\rangle_\pi - \langle M\rangle_{\tilde{\pi}}| \leq \|\pi - \tilde{\pi}\|_1 \|M\| \leq \delta$. We will estimate $\langle M\rangle_{\tilde{\pi}}$ by drawing t samples from $\tilde{\pi}$, which we call $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(t)}$. Our estimator will simply be the sample mean, namely

$$\hat{\bar{M}} := \frac{1}{t}\sum_{i=1}^{t} M(\mathbf{z}_i). \tag{5.17}$$

By Lemma 2 we have

$$\mathbb{P}[|\hat{\bar{M}} - \langle M\rangle_{\tilde{\pi}}| \geq \delta] \leq 2e^{-t\delta^2/2}. \tag{5.18}$$

We conclude that taking $t = O(1/\delta^2)$ samples yields an $O(\delta)$ approximation to $\langle M\rangle_{\tilde{\pi}}$ with high probability. Therefore a polynomial number of samples will suffice to estimate each term of the telescoping product, and it only remains to be seen that the samples can each be produced in polynomial time, which we will prove in the next section.

5.3.4 Markov Chains and Rapid Mixing

To sample from the Gibbs distribution π for the effective classical system (5.11) we will use a Markov chain which chooses a site in the classical lattice Λ uniformly at random and proposes to flip the bit at that site to make a transition $\mathbf{z} \to \mathbf{z}'$ with an acceptance probability $P(\mathbf{z}, \mathbf{z}')$ given by the metropolis rule. Since there are nL sites in the classical lattice, the transition probability between $\mathbf{z}$ and $\mathbf{z}'$ is given by

$$P(\mathbf{z}, \mathbf{z}') = \frac{1}{nL}\min\left\{1, \frac{\pi(\mathbf{z}')}{\pi(\mathbf{z})}\right\} = \frac{1}{nL}\ \min\left\{1, e^{E_\beta(\mathbf{z}') - E_\beta(\mathbf{z})}\right\}. \tag{5.19}$$

The matrix P is called a transition matrix, and since the state space of the random walk is a connected graph and the Metropolis rule satisfies the property of detailed balance, the transition matrix P with transitions satisfying (5.19) will have the Gibbs distribution π as an eigenvector corresponding to an eigenvalue with magnitude 1.

Let $\mathbf{z}$ be the state of the walk at time $t = 0$, and define $P^t(\mathbf{z}, \cdot)$ to be the distribution of the state of the walk at time t. Define the distance to the stationary distribution as

$$d_{\mathbf{z}}(t) = \max_{A \subseteq \Omega} |P^t(\mathbf{z}, A) - \pi(A)| = \frac{1}{2} \sum_{\mathbf{z}' \in \Omega} |P^t(\mathbf{z}, \mathbf{z}') - \pi(\mathbf{z}')|, \tag{5.20}$$

and define the mixing time $\tau(\epsilon)$ to be the worst-case time needed to be within distance ϵ of the stationary distribution

$$\tau(\epsilon) = \max_{\mathbf{z} \in \Omega} \min_{t} \{t : d_{\mathbf{z}}(t') \leq \epsilon \ \ \forall t \geq t'\}. \tag{5.21}$$

5.3.5 Canonical paths

Many tools exist for bounding mixing times, and in chapter 5 we will employ a tool called canonical paths which we now review. A path from $x \in \Omega$ to $y \in \Omega$ is a sequence $\gamma_{x,y} = (v_1, ..., v_k)$ of states with $v_1 = x, v_k = y$ and $P(v_i, v_i + 1) > 0$ for $i = 1, ..., k - 1$. A set $\mathcal{P} = \{\gamma_{x,y}\}$ containing a path that connects every pair of states is called a set of canonical paths. If $\ell = \max_{\gamma \in \mathcal{P}} |\gamma|$ is the maximum length of any path in $\mathcal{P}$ then the mixing time satisfies

$$\tau_{\text{mix}} \leq \mathcal{O}(\rho \, \ell \ln \pi_{\text{min}}^{-1}), \tag{5.22}$$

where the congestion ρ is defined as

$$\rho = \max_{(v,v') \in E(\Omega)} \frac{1}{\pi(v) P(v, v')} \sum_{\substack{\gamma_{x,y} \in \mathcal{P} \\ (v,v') \in \gamma_{x,y}}} \pi(x)\pi(y) \tag{5.23}$$

In order to compute the congestion we will use a standard technique called encoding. For an edge $e = (v, v')$, let $\mathcal{P}(e)$ be the set of paths in $\mathcal{P}$ which pass through e. An encoding is an assignment of an injective function $\eta_e : \mathcal{P}(e) \to \Omega$ to every edge $e \in E(\Omega)$. Suppose we have some $C > 0$ for which

$$\pi(x)\pi(y) \leq C\pi(v)\pi(\eta_e(x, y)) \quad \forall x, y \in \Omega, \tag{5.24}$$

then the congestion satisfies

$$\rho = \max_{(v,v')\in E(\Omega)} \frac{1}{\pi(v)P(v,v')} \sum_{\substack{\gamma_{x,y}\in\mathcal{P} \\ (v,v')\in\gamma_{x,y}}} \pi(x)\pi(y) \tag{5.25}$$

$$\leq \max_{(v,v')\in E(\Omega)} \frac{1}{\pi(v)P(v,v')} \sum_{\substack{\gamma_{x,y}\in\mathcal{P} \\ (v,v')\in\gamma_{x,y}}} C\pi(v)\pi(\eta_e(x,y)) \tag{5.26}$$

$$\leq C \max_{(v,v')\in E(\Omega)} \frac{1}{P(v,v')} \tag{5.27}$$

where (5.26) follows from property (5.24), and (5.27) follows from the injectivity property of the encoding.

5.4 Proof of rapid mixing

To prove rapid mixing for path-integral QMC applied to (5.1) we will use the canonical paths method of subsection 5.3.5. Let $\mathbf{x}, \mathbf{y} \in \Omega$ be spin configurations with $\mathbf{x} = (x_{1,1}, ..., x_{L,1}, ..., x_{1,n}..., x_{L,n})$ and $\mathbf{y} = (y_{1,1}, ..., y_{L,1}, ..., y_{1,n}..., y_{L,n})$. We will describe the canonical path from $\mathbf{x}$ to $\mathbf{y}$ in terms of n steps, each of which consists of L substeps that correspond to transitions of the markov chain. The r-th substep of the m-th step consists of flipping the bit in the r-th trotter slice of the m-th qubit, which corresponds to the edge $(\mathbf{v}, \mathbf{v}') \in E(\Omega)$ with

$$\mathbf{v} = (y_{1,1}, ..., y_{L,1}, ..., y_{L-r-1,m}, x_{L-r,m}, x_{L-r+1,m}..., x_{1,n}, ..., x_{L,n}) \tag{5.28}$$

$$\mathbf{v}' = (y_{1,1}, ..., y_{L,1}, ..., y_{L-r-1,m}, y_{L-r,m}, x_{L-r+1,m}..., x_{1,n}, ..., x_{L,n}). \tag{5.29}$$

The encoding for this path is

$$\eta_{(\mathbf{v},\mathbf{v}')}(\mathbf{x},\mathbf{y}) = (x_{1,1}, ..., x_{L,1}, ...x_{L-r-1,m}, y_{L-r,m}, y_{L-r+1,m}..., y_{1,n}, ..., x_{L,n}) \tag{5.30}$$

To see that $\gamma_{\mathbf{xy}}$ is uniquely determined by $\mathbf{v}$ together with $\boldsymbol{\eta} = \eta_{(\mathbf{v},\mathbf{v}')}(\mathbf{x},\mathbf{y})$, let $\boldsymbol{\sigma} = \mathbf{x} \cdot \mathbf{y}$ be the configuration corresponding to sitewise multiplication of the spins $\{\pm 1\}$ in $\mathbf{x}$ and $\mathbf{y}$, so that the path $\gamma_{\mathbf{xy}}$ described above begins from $\mathbf{x}$ and flips the spin at each site (i,j) for which $\sigma_{i,j} = -1$, in the order described above. The encoding is injective because $\boldsymbol{\sigma} = \mathbf{v} \cdot \boldsymbol{\eta}$.

The weight of a configuration $\mathbf{z} \in \Omega$ on a subset of the classical lattice $A \subseteq \Lambda$ is defined by the restricted classical energy function E^A_β,

$$\pi(\mathbf{z})_A = \frac{e^{-E^A_\beta(\mathbf{z})}}{Z} \quad , \quad E^A_\beta = E_\beta \restriction_A . \tag{5.31}$$

For any $1 \leq r \leq L$ and $1 \leq m \leq n$ define

$$[: r, : m] = (\{1, ..., L\} \times \{1, ..., m-1\}) \cup (\{1, ..., r\} \times \{m\}) \tag{5.32}$$

$$[r : , m :] = (\{r+1, ..., L\} \times \{m\}) \cup (\{1, ..., L\} \times \{m+1, ..., n\}) \tag{5.33}$$

Now we compute

$$\pi(\mathbf{v}) = \pi(\mathbf{y})_{[: r, : m]} \pi(\mathbf{x})_{[r : , m :]} \cdot e^{-B} \tag{5.34}$$

$$\pi(\boldsymbol{\eta}) = \pi(\mathbf{x})_{[: r, : m]} \pi(\mathbf{y})_{[r : , m :]} \cdot e^{-B'} \tag{5.35}$$

where

$$\begin{aligned} B &= J\left(y_{1,m} x_{L,m} + y_{r,m} x_{r+1,m}\right) + \sum_{i=r}^{L} \left(\frac{\beta}{L} K_m y_{i,m-1} x_{i,m} + J x_{i-1,m} x_{i,m} \right) \\ &+ \sum_{i=1}^{r} \left(\frac{\beta}{L} K_{m+1} y_{i,m} x_{i,m+1} + J y_{i-1,m} y_{i,m} \right) \end{aligned}$$

and

$$\begin{aligned} B' &= J\left(x_{1,m} y_{L,m} + x_{r,m} y_{r+1,m}\right) + \sum_{i=r}^{L} \left(\frac{\beta}{L} K_m x_{i,m-1} y_{i,m} + J y_{i-1,m} y_{i,m} \right) \\ &+ \sum_{i=1}^{r} \left(\frac{\beta}{L} K_{m+1} x_{i,m} y_{i,m+1} + J x_{i-1,m} x_{i,m} \right) \end{aligned}$$

we also need

$$\pi(\mathbf{x}) = \pi(\mathbf{x})_{[: r, : m]} \pi(\mathbf{x})_{[r : , m :]} \cdot e^{-B_X} \tag{5.36}$$

$$\pi(\mathbf{y}) = \pi(\mathbf{y})_{[: r, : m]} \pi(\mathbf{y})_{[r : , m :]} \cdot e^{-B_Y} \tag{5.37}$$

where

$$\begin{aligned} B_X &= J\left(x_{1,m} x_{L,m} + x_{r,m} x_{r+1,m}\right) + \sum_{i=r}^{L} \left(\frac{\beta}{L} K_m x_{i,m-1} x_{i,m} + J x_{i-1,m} x_{i,m} \right) \\ &+ \sum_{i=1}^{r} \left(\frac{\beta}{L} K_{m+1} x_{i,m} x_{i,m+1} + J x_{i-1,m} x_{i,m} \right) \end{aligned}$$

and similarly for B_Y. Now we have

$$\begin{aligned}\frac{\pi(\mathbf{x})\pi(\mathbf{y})}{\pi(\mathbf{v})\pi(\boldsymbol{\eta})} &= \frac{\left(\pi(\mathbf{x})_{[:\,r,:\,m]}\pi(\mathbf{x})_{[r:\,,m:\,]}\cdot e^{-B_X}\right)\left(\pi(\mathbf{y})_{[:\,r,:\,m]}\pi(\mathbf{y})_{[r:\,,m:\,]}\cdot e^{-B_Y}\right)}{\left(\pi(\mathbf{y})_{[:\,r,:\,m]}\pi(\mathbf{x})_{[r:\,,m:\,]}\cdot e^{-B}\right)\left(\pi(\mathbf{x})_{[:\,r,:\,m]}\pi(\mathbf{y})_{[r:\,,m:\,]}\cdot e^{-B'}\right)} \\ &= e^{B+B'-B_X-B_Y} \\ &\equiv C(X,Y)\end{aligned}$$

$$\begin{aligned}B+B'-B_X-B_Y &= J\left(y_{1,m}x_{L,m}+y_{r,m}x_{r+1,m}+x_{1,m}y_{L,m}+x_{r,m}y_{r+1,m}\right) \\ &- J\left(x_{1,m}x_{L,m}+x_{r,m}x_{r+1,m}+y_{1,m}y_{L,m}+y_{r,m}y_{r+1,m}\right)+ \\ &+ \frac{\beta}{L}\left(\sum_{i=1}^{r}K_{m+1}y_{i,m}x_{i,m+1}+\sum_{i=r}^{L}K_m y_{i,m-1}x_{i,m}\right) \\ &+ \frac{\beta}{L}\left(\sum_{i=1}^{r}K_{m+1}x_{i,m}y_{i,m+1}+\sum_{i=r}^{L}K_m x_{i,m-1}y_{i,m}\right) \\ &- \frac{\beta}{L}\left(\sum_{i=1}^{r}K_{m+1}x_{i,m}x_{i,m+1}+\sum_{i=r}^{L}K_m x_{i,m-1}x_{i,m}\right) \\ &- \frac{\beta}{L}\left(\sum_{i=1}^{r}K_{m+1}y_{i,m}y_{i,m+1}+\sum_{i=r}^{L}K_m y_{i,m-1}y_{i,m}\right) \\ &\leq 8J+4\beta\end{aligned}$$

therefore to satisfy (5.24) it suffices to take $C = e^{8J+4\beta}$. If the 1D chain has periodic boundary conditions connecting qubit 1 to qubit n, then the same analysis would show that $C = \exp^{8(J+\beta)}$ satisfies (5.24). By (5.25)-(5.27) the congestion satisfies

$$\rho = C \max_{(V,V')\in E(\Omega)} \frac{1}{P(V,V')} = e^{8(\beta+J)}\cdot 2nL\cdot e^{2\left(J+\frac{\beta}{L}\right)}$$

the factor of $2nL$ comes from the chain being lazy and the choice of random single-site updates (which means there is a $1/nL$ chance of proposing each particular transition). To compute the mixing time we also need $\ell = nL$ and $\ln \pi_{\min}^{-1} = \ln(Z\cdot e^{nLJ+n\beta}) \leq nL(J+\beta+\ln 2)$, where we've used the fact that $Z \leq 2^{nL}$. Putting this together with (5.22) we obtain the following bound on the mixing time,

$$\tau_{\text{mix}} \leq \mathcal{O}((nL)^3 e^{10J+8\beta+2\frac{\beta}{L}}(J+\beta+\ln 2)).$$

By definition (5.21) this shows that the PI-QMC method generates samples as was assumed in subsection 5.3.3, and this completes the proof of theorem 1.

5.5 Conclusion

In the previous sections we provided a new analysis of the convergence of the path-integral quantum Monte Carlo for the case of 1D generalized transverse Ising models, and we've shown that this method results in an efficient algorithm for approximating the partition function at any temperature which is independent of the system size. In addition to the practical use of such a simulation method for studying the thermal properties of these quantum systems, we also take the view that the computational complexity of simulating these systems is a physical property of intrinsic interest. In this light our result is seen as a new contribution to the topic of stoquastic complexity, which aims to distinguish the computational capabilities of these systems which lie at the border between quantum and classical computation.

We believe it is possible to extend the analysis here to more general families of 1D stoquastic Hamiltonians that go beyond the transverse Ising form. A significant challenge arises in these models when the local moves applied in this work no longer suffice to make the random walk ergodic (see [San10] for a discussion of this well-known issue in the case of the ferromagnetic Heisenberg model). In practice this issues are usually resolved in an ad hoc by designing non-local update rules which make use of specific properties of the Hamiltnonian of interest. In future work we will pursue a general solution to the ergodicity problem which adds a small fictitious transverse field term to these Hamiltonians which would not disturb the partition function to within the error that the algorithm attempts to achieve. To analyze the convergence of the random walk in this case, the system of canonical paths used in the proof of rapid mixing would need to account for the different energetic contributions of the fictitious transverse field and the terms which are genuinely present in the Hamiltonian of interest. In addition to simulating 1D stoquastic models, we are hopeful that this technique may also be useful for the analysis of the path-integral quantum Monte Carlo method to spin systems with higher dimensional interaction graphs.

BIBLIOGRAPHY

[AN02] Dorit Aharonov and Tomer Naveh. Quantum NP - a survey, 2002. arXiv:quant-ph/0210077.

[Bab85] L. Babai. *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*. ACM Press, 1985.

[Bar82] J. Barahona. *Phys A: Math. Gen.*, page 3241, 1982.

[Bax82] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics*. London Academic Press, 1982.

[BBT06] Sergey Bravyi, Arvid J. Bessen, and Barbara M. Terhal. Merlin-arthur games and stoquastic complexity. 2006.

[BDI75] R. Balian, J. M. Drouffe, and C. Itzykson. *Phys. Rev. D*, page 2098, 1975.

[BDOT08] Sergey Bravyi, David P. DiVincenzo, Roberto I. Oliveira, and Barbara M. Terhal. The complexity of stoquastic local hamiltonian problems. *Quant. Inf. Comp.*, 8(5):0361–0385, 2008.

[BF08] D. Bacon and S. T. Flammia. *Phys. Rev. Lett.*, page 120504, 2008.

[BF10] D. Bacon and S. T. Flammia. Adiabatic cluster state quantum computing. 2010. arXiv:0912.2098 [quant-ph].

[BH14] Sergey Bravyi and Matthew Hastings. On complexity of the quantum ising model. 2014. arXiv:1410.0703 [quant-ph].

[BL08] J. D. Biamonte and P. J. Love. *Phys. Rev. A.*, page 012352, 2008.

[Bra14] Sergey Bravyi. Monte carlo simulation of stoquastic hamiltonians. 2014. arXiv:1402.2295 [quant-ph].

[BRI+14] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer. Quantum annealing with more than one hundred qubits. *Nature Phys.*, page 218, 2014. arXiv:1304.4595 [quant-ph].

[BST05] D. Battaglia, G. E. Santoro, and E. Tosatti. Optimization by quantum annealing: Lessons from hard 3-sat cases. *Phys. Rev. E*, 2005. arXiv:cond-mat/0502468.

[BSVV05] I. Bezakova, D. Stefankovic, V. Vazirani, and E. Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM Journal of Computing*, 37(5):1429–1454, 2005.

[BT09] Sergey Bravyi and Barbara M. Terhal. Complexity of stoquastic frustration-free hamiltonians. *SIAM J. Comput.*, 39(4):1462–1485, 2009. arXiv:0806.1746 [quant-ph].

[CBB10] Elizabeth Crosson, Dave Bacon, and Kenneth R. Brown. Making classical ground state spin computing fault-tolerant. *Phys. Rev. E*, 82(3), 2010. arXiv:1006.4388 [cond-mat.stat-mech].

[CCF+11] T. Caneva, T. Calarco, R. Fazio, G. E. Santoro, and S. Montangero. Speeding up critical system dynamics through optimized evolution. *Phys. Rev. A*, 84(1):012312, 2011.

[CCMP05] C. Castelnovo, C. Chamon, C. Mudry, and P. Pujol. From quantum mechanics to classical statistical physics: generalized rokhsar-kivelson hamiltonians and the "stochastic matrix form" decomposition. *Annals of Physics*, pages 316–344, 2005.

[CD14] Elizabeth Crosson and Mingkai Deng. Tunneling through high energy barriers in simulated quantum annealing. 2014. arXiv:1401.7320 [quant-ph].

[CFL+14] Elizabeth Crosson, Edward Farhi, Cedric Yen-Yu Lin, Han-Hsuan Lin, and Peter Shor. Different strategies for optimization using the quantum adiabatic algorithm. 2014. arXiv:1401.7320 [quant-ph].

[Coo71] Cook. *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*. ACM Press, 1971.

[CP10] Alessandra Cipriani and Paolo Dai Pra. Decay of correlations for quantum spin systems with a transverse field: A dynamic approach. 2010. arXiv:1005.3547 [math.PR].

[Egg74] T. P. Eggarter. *Phys. Rev. B.*, page 2989, 1974.

[EKPS00] W. Evans, C. Kenyon, Y. Peres, and L. J. Schulman. *The Annals of Applied Probability*, page 410, 2000.

[FGG+01] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. *Science*, page 472, 2001.

[FGG02] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. Quantum adiabatic evolution algorithms with different paths. 2002. quant-ph/0208135.

[FGGS00] Edward Farhi, Jeffery Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. 2000. quant-ph/0001106.

[FGH+12] Edward Farhi, David Gosset, Itay Hen, A. W. Sandvik, Peter Shor, A. P. Young, and Francesco Zamponi. Performance of the quantum adiabatic algorithm on random instances of two optimization problems on regular hypergraphs. *Phys. Rev. A*, 86(5):052334, 2012. arXiv:1208.3757 [quant-ph].

[Fro12] G. Frobenius. Ueber matrizen aus nicht negativen elementen. *Sitzungsber. Knigl. Preuss. Akad. Wiss*, pages 456–477, 1912.

[GS86] Goldwasser and Sipser. *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*. ACM Press, 1986.

[HF13] M. B. Hastings and M. H. Freedman. Obstructions to classically simulating the quantum adiabatic algorithm, 2013. arXiv:1302.5733 [quant-ph].

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *JASA*, 58(301):13–30, Mar 1963.

[Hor86] R. Horn. *Topics in Matrix Analysis*. Cambridge University Press, 1986.

[Isi25] E. Ising. Beitrag zur theorie des ferromagnetismus. *Z. Phys.*, pages 253–258, 1925.

[JS93] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on Computing*, 22:1087–1116, 1993.

[KGV83] Kirkpatrick, Gelatt, and Vecchi. Optimization by simulated annealing. *Science*, pages 671–680, 1983.

[Kis85] L. B. Kish. *Optics News*, page 11, 1985.

[KN98] T. Kadowaki and H. Nishimori. Quantum annealing in the transverse ising model. *Phys. Rev. E*, 1998.

[KR06] J. Kempe and O. Regev. *SIAM Journal of Computing*, page 1070, 2006.

[KSV02] A. Yu Kitaev, A. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2002.

[LE75] S. Lie and F. Engel. Theorie der transformationsgruppen. *Leipzig*, 1875.

[Lid08] D. A. Lidar. *Phys. Rev. Lett.*, page 160506, 2008.

[LPS86] A. Lubotzky, R. Phillips, and O. Sarnak. ACM, 1986.

[LTP94] C. Lent, P. Tougaw, and W. Porod. *Proceedings of 1994 Conference on Physics and Computation*, 1994.

[LTPB93] C. Lent, P. Tougaw, W. Porod, and G. Bernstein. *Nanotechnology*, page 49, 1993.

[Lyo89] R. Lyons. *Comm. Math. Phys.*, page 337, 1989.

[Miz04] A. Mizel. *Phys. Rev. A.*, page 012304, 2004.

[MMC02] A. Mizel, M. W. Mitchell, and M. L. Cohen. *Phys. Rev. A.*, page 022315, 2002.

[MS08] I. L. Markov and Y. Shi. *SIAM Journal on Computing*, page 963, 2008.

[MST02] R. Martoňák, G. E. Santoro, and E. Tosatti. Quantum annealing by the path-integral monte carlo method: The two-dimensional random ising model. *Phys Rev B*, 2002.

[MW11] F. Martinelli and M. Wouts. Glauber dynamics for the quantum ising model in a transverse field on a regular tree. 2011. arXiv:1105.5970 [math.PR].

[NSK12] Daniel Nagaj, Rolando D. Somma, and Maria Kieferova. Quantum speedup by quantum annealing. *Phys. Rev. Lett.*, 109(5):050501, 2012. arXiv:1202.6257 [quant-ph].

[OBL09] O. Oreshkov, T. A. Brun, and D. A. Lidar. *Phys. Rev. Lett.*, page 070502, 2009.

[Ore09] O. Oreshkov. *Phys. Rev. Lett.*, page 090502, 2009.

[OT08] R. Oliveira and B. Terhal. *Quantum Inform. Compu.*, page 0900, 2008.

[Per07] O. Perron. Zur theorie der matrices. *Mathematische Annalen*, pages 248–263, 1907.

[Pip85] N. Pippenger. *Proc. of the 26th Annual Symposium on Foundations of Computer Science*, pages 30–38, 1985.

[Pri05] Vladimir Privman. *Nonequilibrium Statistical Mechanics in One Dimension*. Cambridge University Press, 2005.

[RK88] D. S. Rokhsar and S. A. Kivelson. *Phys. Rev. Lett*, 1988.

[San10] Anders W. Sandvik. Computational studies of quantum spin systems. *AIP Conf. Proc.*, 1297:135, 2010. arXiv: 1101.3281 [cond-mat.str-el].

[Sch78] T. J. Schaefer. *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978.

[Sha92] A. Shamir. *J. ACM*, page 869, 1992.

[Sho94] P. W. Shor. *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, 1994.

[Sin92] Alistair Sinclair. Improved bounds for mixing rates of markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1(4):351–370, 1992.

[Sip05] M. Sipser. Introduction to the theory of computation. 2005.

[SMK77] Masuo Suzuki, Seiji Miyashita, and Akira Kuroda. Monte carlo simulation of quantum spin systems. *Prog. Theor. Phys.*, 58(5):1377–1387, 1977.

[Tra84] B. A. Trakhtenbrot. *IEEE Annals of the History of Computing*, volume 6. 1984.

[Tur36] A. M. Turing. *Proc. Lond. Math. Soc. 2*, page 230, 1936.

[UFMI00] C. Ungarelli, S. Francaviglia, M. Macucci, and G. Iannaconne. *Journal of Applied Physics*, page 7320, 2000.

[von56] J. vonNeumann. *Automata Studies*. IEEE Computer Society, 1956.

[VTB09] C. R. Viteri, Y. Tomita, and K. R. Brown. *Phys. Rev. A*, page 042313, 2009.

[WL04] Y. Wang and M. Lieberman. *IEEE Transactions on Nanotechnology*, page 368, 2004.

[WS98] H. Wu and D. Sprung. *J. Appl. Phys*, page 4000, 1998.