Classical Approaches to Understanding Quantum Systems

Seung Woo Shin

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-191
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-191.html

December 1, 2016

Classical Approaches to Understanding Quantum Systems

by

Seung Woo Shin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Umesh V. Vazirani, Chair
Professor Satish Rao
Professor K. Birgitta Whaley

Spring 2016

The dissertation of Seung Woo Shin, titled Classical Approaches to Understanding Quantum Systems, is approved:

Chair _____ Date _____

_____ Date _____

_____ Date _____

University of California, Berkeley

Classical Approaches to Understanding Quantum Systems

Abstract

Classical Approaches to Understanding Quantum Systems

by

Seung Woo Shin

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Umesh V. Vazirani, Chair

While the exponential complexity of quantum systems is the basis of counterintuitive phenomena such as quantum computing, it also represents a fundamental challenge: how can we, classical beings, study, understand, and control quantum systems that are exponentially more powerful than ourselves? In this dissertation, we present classical methods to approach two such issues.

Firstly, we discuss the testing of quantum devices, particularly special-purpose quantum computers. We propose a simple classical model for quantum annealers, arguably the most intensely explored class of special-purpose quantum computers. The model provides a benchmark against which to compare the quantum annealer, in what may be called a "quantum Turing test," to determine whether the quantum annealer exhibits algorithmically significant quantum behavior. An application of the test reveals that the input-output behavior of the benchmark agrees with published data from the D-Wave One quantum annealer on random instances of its native problem on 108 qubits, and closely matches the reported performance of D-Wave 2X on special instances devised to exercise quantum tunneling. In other words, the machine does not pass the quantum Turing test with respect to these inputs. A more detailed analysis of the new classical model yields further algorithmic insights into the nature of quantum annealing.

Secondly, we show that commuting stoquastic quantum k-SAT, an interesting variant of the local Hamiltonian problem, is in **NP** for any $k = O(\log n)$. The result follows from a study of the computational complexity of tensor network nonzero testing, a fundamental problem in quantum Hamiltonian complexity. We show that the problem in its most general form is computationally very hard, i.e., not contained in the polynomial hierarchy unless the hierarchy collapses. On the other hand, we are able to identify two "easy" special cases of tensor network nonzero testing, namely nonnegative tensors and injective tensors, which may be useful in certain contexts. Indeed, our main result follows by exhibiting a direct connection between the special case of nonnegative tensor networks and commuting stoquastic quantum k-SAT.

To my family

Contents

Contents ii

List of Figures iv

1 Introduction 1
- 1.1 Testing of a quantum computer . . . 2
- 1.2 Tensor network nonzero testing . . . 4
- 1.3 Outline of the thesis . . . 5

2 Background 7
- 2.1 Quantum computing: the new limit of quantum mechanics . . . 7
- 2.2 Three principles of quantum mechanics . . . 10
- 2.3 Complexity of quantum systems: the local Hamiltonian problem . . . 14
- 2.4 Quantum Hamiltonian complexity: connections to condensed matter physics 17
- 2.5 Tensor networks . . . 20
  - 2.5.1 Basic concepts . . . 20
  - 2.5.2 Computational aspects . . . 27

3 A Turing Test for Quantum Annealers 30
- 3.1 Introduction . . . 30
  - 3.1.1 Related work . . . 32
  - 3.1.2 Preliminaries . . . 33
    - 3.1.2.1 Pauli matrices . . . 33
    - 3.1.2.2 Bloch sphere . . . 34
  - 3.1.3 Introduction to quantum annealing . . . 35
    - 3.1.3.1 Top-down approach to quantum computing . . . 35
    - 3.1.3.2 Adiabatic quantum computing . . . 37
    - 3.1.3.3 Quantum annealing . . . 38
- 3.2 Quantum Turing test . . . 41
- 3.3 Classical models for quantum annealers . . . 42
  - 3.3.1 Simulated annealing . . . 43
  - 3.3.2 Simulated quantum annealing . . . 45

3.3.3 Classical spin dynamics . . . 46
3.3.4 Our model . . . 46
3.4 Benchmarking the D-Wave machine . . . 49
3.4.1 Random instances of the D-Wave native problem . . . 52
3.4.2 Eight-qubit motif problem of Vinci et al. . . . 53
3.4.3 Sixteen-qubit motif problem of Boixo et al. . . . 60
3.5 Discussion . . . 65
3.5.1 Synchronized flipping of spins . . . 65
3.5.2 Deterministic behavior . . . 68
3.6 Conclusions . . . 69

4 Tensor Network Nonzero Testing **70**
4.1 Introduction . . . 70
4.1.1 Tensor networks in quantum Hamiltonian complexity . . . 70
4.1.2 Counting problems vs. decision problems . . . 71
4.1.3 Commuting local Hamiltonians . . . 73
4.2 Preliminaries . . . 75
4.2.1 Polynomial hierarchy . . . 75
4.2.2 Physical interpretations of tensor networks . . . 76
4.2.3 Commuting local Hamiltonians . . . 77
4.2.4 Quantum k-SAT . . . 77
4.2.5 Stoquastic Hamiltonians . . . 78
4.2.6 Tensor network nonzero testing . . . 78
4.3 Hardness of tensor network nonzero testing . . . 79
4.3.1 Generalized tensor network nonzero testing . . . 79
4.3.2 Tensor network nonzero testing . . . 80
4.4 Special cases of tensor network nonzero testing . . . 82
4.4.1 Nonnegative tensor networks . . . 82
4.4.2 Injective tensor networks . . . 83
4.5 Connections to quantum Hamiltonian complexity . . . 87
4.6 Conclusions . . . 88

5 Conclusions **90**

Bibliography **92**

List of Figures

2.1 Rewriting a quantum state in two different bases. For convenience in illustration, we are assuming that $|\psi\rangle, |u\rangle, |u^{\perp}\rangle$ are real vectors. 12

2.2 A schematic representation of a 3-local Hamiltonian $H = \sum_i H_i$. Each term H_i acts on at most three particles at a time. 15

2.3 An illustration of translation invariance on a 3×3 2D lattice. If each node represents a d-dimensional particle, A and B will be $d^2 \times d^2$ matrices. For each vertical edge (i, j), we take the tensor product of the matrix A that acts upon particles i and j and the identity matrix that acts upon all other particles to obtain $H_{i,j}$. For each horizontal edge (i, j), we similarly take the tensor product of the matrix B that acts upon particles i and j and the identity matrix on all other particles. Our translationally invariant 2-local Hamiltonian will simply be defined as $H = \sum_{(i,j)\in E} H_{i,j}$. 19

2.4 An example tensor network with two tensors of rank 3. Each tensor has two open edges and one closed edge, and it is assumed that every edge has a bond dimension of two. Labeling each edge with either 0 or 1, the input to each tensor is completely specified, so that it can output a complex number. The value of the tensor network at this labeling is simply defined as the product of these two complex numbers. 21

2.5 A pictorial illustration of how a tensor of rank 3 with bond dimension 2 on each edge can be interpreted as a 2^3-dimensional vector, a 2×2^2 matrix, or a $2^2 \times 2$ matrix. Namely, in this notation a tensor is interpreted as a linear map from a vector space whose basis vectors are indexed by its upper edges to another vector space whose basis vectors are indexed by its lower edges. 23

2.6 Tensor product of three vectors $|\psi_1\rangle$, $|\psi_2\rangle$, and $|\psi_3\rangle$. 24

2.7 Inner product of two vectors $|\psi_1\rangle$ and $|\psi_2\rangle$. Note that prior to the joining of the edges, one of the vectors (in this case $|\psi_2\rangle$) needs to be modified so that the modified vector's outputs are complex conjugates of the original vector's outputs. This is why the bottom vector on the right-hand side is labeled $\langle\psi_2|$ instead of $|\psi_2\rangle$. 24

2.8 Matrix multiplication of a 2×2^2 matrix A and a $2^2 \times 2$ matrix B, which yields a 2×2 matrix C. 25

2.9 Trace of a $2^4 \times 2^4$ matrix A. 26

2.10 Partial trace of a $2^4 \times 2^4$ matrix A over the third and fourth particles. 26
2.11 $\mathrm{Tr}(AB) = \langle \mathrm{flatten}(A), \mathrm{flatten}(B) \rangle$ for real matrices A and B. 27
2.12 $\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA) = \mathrm{Tr}(CAB)$. 27
2.13 Swallowing of a tensor network. 28
2.14 A matrix product state. 29
2.15 The tensor network corresponding to $\langle \psi | O | \psi \rangle$. 29

3.1 A Turing test. 31
3.2 The Bloch sphere representation of one-qubit state $|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$. 35
3.3 A schematic illustration of quantum tunneling vs. classical thermal fluctuations. 40
3.4 Classical 2D spins. Since the state of each individual qubit is fully described by the corresponding 2D vector, no entanglement is possible between these qubits. . 46
3.5 The annealing schedule of D-Wave One (figure adapted from the supplementary materials of [24]). In the actual implementation, the effective schedule may slightly vary from spin to spin (see the supplementary materials of [24] for details). 50
3.6 The "Chimera" interaction graph of D-Wave One (figure adapted from the supplementary materials of [24]). Due to issues in implementation, not all of the vertices on the Chimera graph represent working qubits on the device. The defective qubits are colored grey and are excluded from experiments. 51
3.7 Histogram of success probabilities of D-Wave One, our classical model, and simulated annealing (SA). Unlike simulated annealing, the D-Wave machine and our model exhibit a clear bimodal distribution. The histogram for simulated annealing (right panel) was borrowed from [24]. 54
3.8 Correlation between D-Wave and our model. Each simulation of our model consisted of 150,000 steps, following the annealing schedule of D-Wave One from Figure 3.5. The system temperature of $T = 0.22\mathrm{GHz} \approx 11\mathrm{mK}$ was used. The (Pearson's) correlation coefficient R between the D-Wave One and our model is about 0.91. 54
3.9 Correlation between simulated quantum annealing of [24] and our model. The correlation coefficient R is about 0.99. 55
3.10 The eight-qubit motif problem proposed in [116], which can be mapped to a single supernode of the D-Wave machine. All couplings are ferromagnetic, whereas there is a local z-field applied in the positive direction for the four "core" spins, and in the negative direction for the four "peripheral" spins. Formally, the final Hamiltonian is defined as $H_f = -\sum_i h_i \sigma_i^z - \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z$. The local field h_i is set to be 1 if i is a core spin, and -1 otherwise. The coupling strength $J_{ij} = 1$ for every edge $\{i, j\}$. Figure is borrowed from [116]. 55
3.11 The solid curves represent the annealing schedule of D-Wave Two. Dotted blue curves represent the effective annealing schedule for cases $\alpha = 0.2834$ and $\alpha = 0.1099$. The dotted black line represents the system temperature. Figure is borrowed from [116]. 56

3.12 Experimental and numerical results from [116]. DW2, ME, SA, SD, and SSSV represent D-Wave Two, quantum adiabatic Markovian master equation (exponential quantum simulation), simulated anneailng, classical spin dynamics, and our classical model respectively. P_{GS} denotes the probability of finding one of the seventeen ground states. Figure is borrowed from [116]. 57
3.13 Simulations of our classical model with Gaussian local noise. The model produces a behavior similar to that of the D-Wave machine or quantum adiabatic master equation from Figure 3.12. The model was simulated for 1,500 steps at the system temperature of $T = 0.22$GHz. Ten thousand runs were performed for each value of α. 58
3.14 Further simulations of our modified classical model reproduce various other signatures suggested in [116]. The top-left panel, which plots the trace-norm distance [84] between the simulated state at the end of the annealing process and the Gibbs state for the final Hamiltonian, is a good qualitative fit to the experimental data presented in Figure 14 of [116]. The top-right, bottom-left, and bottom-right panels are simulation results on larger instances of the problem with 12, 16, and 20 spins respectively (for details about the construction of these instances, see [116]), and are consistent with the experimental results from Figure 10 of [116]. 59
3.15 The sixteen-qubit motif problem proposed in [23], which can be mapped to two adjacent supernodes of the D-Wave machine. Figure is borrowed from [43]. . . . 60
3.16 The success probabilities of various algorithms as the annealing temperature T is varied. The data points represent the D-Wave machine, quantum simulations based on master equations (NIBA and Redfield), simulated quantum annealing (PIMC-QA), and our classical model (SVMC). Figure is borrowed from [23]. . . 61
3.17 A 945-qubit instance constructed in [43] using Boixo et al.'s 16-qubit motif problem as a building block. Black indicates a strong cluster spin and grey indicates a weak cluster spin. Couplings between neighboring strong clusters are chosen to be either ferromagnetic (blue) or antiferromagnetic (red) at random. Figure is borrowed from [43]. 62
3.18 A modified annealing schedule used in our quantum Turing test. 63
3.19 Performances of various algorithms on the instances of [43]. The data points for D-Wave 2X, simulated quantum annealing (SQA), and simulated annealing (SA) were taken from [43]. Each run of our classical model consisted of 2,000 steps and the system temperature of $T = 0.22\text{GHz} \approx 11\text{mK}$ was used. Once the success probability s is estimated for each instance, the time to find the optimal solution with 99% probability is calculated as (runtime for one run on a single core)/(# spins)$\cdot \frac{\log(1-0.99)}{\log(1-s)}$, where the factor 1/(# spins) accounts for the amount of parallelism inherent in the D-Wave machine. 64

3.20 The performance of our model on a single instance of the 16-qubit motif problem. The success probability clearly decreases with annealing temperature T, a behavior which was interpreted in [23] as a signature of quantum tunneling. Compare to the D-Wave data in Figure 3.16. 65
3.21 Role of transverse field. The right panel is a snapshot of a typical simulation run on the 108-qubit instances of [24], where the z-components of all 108 spins are plotted in decreasing order. 66
3.22 The first "branching point" of instance 13-55-29 at $t = 0.13$. There are two alternatives considered by the model at this point, i.e. $H(0.13)$ has two distinct local minima up to the two-fold symmetry of flipping all spins. Blue dots indicate the "difference" between these two alternatives, i.e. blue dots represent the spins on which the signs of z-components differ between the two alternatives. The Chimera graph figure was borrowed and modified from [24]. 67

4.1 The polynomial hierarchy. 73
4.2 The computational complexity of the commuting local Hamiltonian problem is not known in the general case. 74
4.3 When a tensor network is viewed as a quantum state or operator, open edges (dashed lines) are interpreted as corresponding to the physical particles. 76
4.4 (a) An example tensor network T. (b) The subnetwork of T induced by vertices $\{v_1, v_2\}$. Note that the edges (v_1, v_4) and (v_2, v_3) are treated as open edges in the subnetwork. Namely, $E_T^{\text{phys}}(S)$ contains the dashed open edges in (b), whereas $E_T^{\text{vir}}(S)$ contains the solid open edges. 83
4.5 Illustrating the proof of Theorem 4.6. 85
4.6 The tensor network T' in the proof of Theorem 4.7. In this example, $L = \{v_1\}$, $S_i = \{v_2, \ldots, v_{n-1}\}$, and $R = \{v_n\}$. 87
4.7 An example tensor network representation of Π. 88

Acknowledgments

First and foremost, I thank my advisor Umesh Vazirani. Looking back on the years I spent here at Berkeley, I can only vaguely trace the paths that I have trodden and Umesh guided me to tread. I came to Berkeley knowing that I wanted to learn quantum physics and quantum computing but knowing practically nothing about them. How could it be that all the things that happened happened, I do not know. One thing I remember well is that I enrolled in Umesh's quantum computing course in my first year and it was the most interesting course I had ever taken. And ever since, my life was only about chasing the small and immediate goals that lay before me. Therefore, if these small steps have somehow converged to a larger theme, I take no credit for that; it probably happened thanks to Umesh's careful guidance and by virtue of the environment. I take credit only for the mechanical work that was necessary to make the small steps.

I thank Zeph Landau and Sev Gharibian for showing me what it means to be a researcher. I am especially grateful to Zeph for teaching me the tensor network formalism, which became an invaluable part of my skill set. I thank Dorit Aharonov, Rahul Jain, Guy Kindler, Ashwin Nayak, and Miklos Santha for hosting me at their institutions at various times and for the wonderful experiences that only they could make possible. I thank the Berkeley theory group professors Christos Papadimitriou, Prasad Raghavendra, and Satish Rao for setting such high examples. I learned a lot watching their brilliant minds at work.

I thank Satish Rao and K. Birgitta Whaley for being on my dissertation committee and for their guidance in the writing of this thesis. I also acknowledge useful discussions with Tameem Albash, Sergio Boixo, Yichen Huang, Daniel A. Lidar, Daniel Nagaj, Troels Rønnow, Graeme Smith, John A. Smolin, and Matthias Troyer during research that was to become the body of this thesis.

Last but not least, I would like to acknowledge the past and present members of the Berkeley theory group – Nima Anari, Anand Bhaskar, Jonah Brown-Cohen, Siu Man Chan, Siu On Chan, Paul Christiano, James Cook, Anindya De, Rafael Frongillo, Fotis Iliopoulos, Varun Kanade, Jingcheng Liu, Peihan Miao, George Pierrakos, Alex Psomas, Aviad Rubinstein, Manuel Sabin, Aaron Schild, Tselil Schramm, Jarett Schwartz, Piyush Srivastava, Isabelle Stanton, Ning Tan, Greg Valiant, Thomas Vidick, Di Wang, Tom Watson, Ben Weitz, Chris Wilkens, and Sam Wong – for the invigorating discussions, inspiration, and most importantly the fun. I extend my special thanks to my only classmate Antonio Blanca for our shared struggles and escapades in the earlier days of PhD effort, and my fellow quantum students Urmila Mahadev, Anupam Prakash, and Guoming Wang for helping me activate the non-classical part of my brain.

Chapter 1

Introduction

The potential enormous computational power of quantum systems was recognized in as early as 1980's, when e.g. Feynman [51] observed that the number of parameters required to specify the quantum state of an n-particle system grows exponentially in n. This exponential nature of quantum systems has since motivated close collaborations between computer science and quantum physics and led to the birth of several new fields dedicated to studying these connections, such as quantum computing, quantum information science, quantum cryptography, and quantum Hamiltonian complexity. While being a crucial resource in these quantum technologies, this same exponential power also presents a formidable challenge; namely, how can we, classical beings, hope to study, understand, and control quantum systems which are much more complex and powerful than ourselves? To add to this difficulty, the laws of quantum mechanics severely limit the accessible information about the quantum state of a system to be linear in the number of particles [71]. Thus both science and engineering seem to require fundamentally new approaches in the quantum realm, where techniques we often take for granted, such as simulation or verification, become intractable.

In the literature, the challenge of dealing with high complexity quantum systems has been formulated prominently in two different contexts. Firstly, it has been observed that the testing of quantum computers poses a unique challenge, largely due to the gap in computational power between the quantum computer and the classical tester. Can the classical tester verify that the quantum computer is indeed performing the desired computation? Or, if the quantum computer is cheating, can the classical verifier catch it? Remarkably, a sequence of results [34, 1, 18, 52, 19, 92] shows that the framework of interactive proofs from complexity theory provides an effective approach to this problem. In particular, these techniques allow a fully classical verifier to command two entangled but non-communicating quantum computers [92], or a classical verifier with capacity for constant-size quantum computation to verify a fully universal quantum computer [34, 1]. Unfortunately, it is not clear whether these techniques can be adapted to work for a wider range of quantum systems that do not fit into these scenarios.

Secondly, the exponential nature of quantum systems is a major obstacle to understanding physical properties of matter. In quantum many-body physics, in which a given material is

often modeled using local Hamiltonians whose terms act on a constant number of neighboring particles, the ground states of such Hamiltonians determine the properties of the material at low temperatures. In view of the exponential description complexity of quantum states, this naturally leads one to consider the following question: when do the ground states of local Hamiltonians admit an efficient classical description? In full generality, the problem of finding the ground state of a local Hamiltonian is known to be **QMA**-hard [78], which effectively says that there is no general way to classically access the ground states of local Hamiltonians. On the other hand, recent advances in quantum Hamiltonian complexity allow us to circumvent some of these difficulties by exploiting structures of local Hamiltonians that naturally arise in physics – indeed, ground states of some such "physical" Hamiltonians are known to be efficiently computable (see e.g. [56] for a survey). While this gives us some hope that most physical systems may turn out to live in a small corner of the Hilbert space in which efficient computation is possible, many systems still remain outside of the reach of any classical approach and are waiting to be explored.

This thesis attempts to advance the state of the art with respect to these challenges by presenting two purely classical methods for understanding quantum systems. Firstly, we develop a classical benchmark for quantum annealing which can be used to test for quantum coherence or quantum speedup in a given implementation. Secondly, we investigate the possibility of classically verifying ground states of certain local Hamiltonian problems, via a fundamental problem called tensor network nonzero testing. We hope that the methods presented in this thesis will continue to play a role in future research on these topics.

1.1 Testing of a quantum computer

In general, the testing of quantum devices is a particularly vexing issue both because the laws of quantum mechanics severely limit the accessible information about the quantum state of a system [71], and because of the exponential gap in computational power between the quantum device and the classical tester. The nature of this challenge can perhaps be well summarized in the following analogy: suppose we had an extraterrestrial friend K, who claims he can travel near the speed of light. Every now and then, K begins to tell us stories about the exquisite cuisine on Sirius B, the sophisticated culture of Alpha Centauri, and so on and so forth. While we are generally amused by these stories, we wonder whether there is any way for us to verify the authenticity of his claims. Unfortunately, being mere human beings, we are utterly unable to travel with him to these places ourselves. Despite this disadvantage, can we still somehow ensure that K was indeed in Alpha Centauri last Thursday, or that he can actually travel at the speed he claims?

Of course, there are certain claims about quantum computers that can be tested using the existing theory of quantum computing. For instance, Shor's algorithm for factoring [102] is known to be able to factor numbers exponentially faster than the best classical algorithm. Since the solution to a factoring problem can be easily verified by multiplying the returned factors, factoring of very large numbers can be used to test whether a given

quantum computer is capable of an exponential quantum speedup. On the other hand, an implementation of Shor's algorithm would only test the machine with respect to one specific problem, which raises the natural question of whether there is a way to test it for more general problems. Remarkably, recent results on the testing of quantum computers [34, 1, 18, 52, 19, 92] provide a proof of principle that this is indeed possible for arbitrary **BQP** computation. A major open question in this line of research is whether a fully classical verifier can test a single quantum machine without resorting to the additional assumption of two-party entanglement or ability to perform constant-size quantum computation.

On the other hand, while such schemes would provide a clear-cut and unambiguous test for universal quantum computers, they tend to demand that the quantum computer that is being tested implements a rather strong model of quantum computation. For instance, even if the actual computation to be verified is very simple, the testing protocol might require the quantum computer to perform arbitrarily complex **BQP** computation. Therefore, these schemes do not address the testing of special-purpose quantum computation. This is a serious shortcoming, especially since general-purpose quantum computation presents an enormous engineering challenge which is not expected to be overcome in the next decade.

In fact, there has been more optimism about the possibility of building large-scale special-purpose quantum computers in recent years, partly due to these challenges in achieving general-purpose quantum computation. Instead of investing too much effort into improving and verifying the elementary components of a quantum computer, the engineering of special-purpose quantum computers often follows a "top-down" approach, which aims to first build *some* quantum system that achieves nontrivial quantum coherence and only then tries to find interesting computational problems that could be mapped to that system. In particular, quantum annealing, which can be thought of as a heuristic implementation of adiabatic quantum optimization [49], proposes a promising approach to finding the ground state of a given classical Hamiltonian, which is an **NP**-hard optimization problem.

The first contribution of this thesis is a classical benchmark for the quantum annealer, arguably the most intensely explored class of special-purpose quantum computers. In particular, we show that our classical benchmark provides a way to test these machines via exclusively classical methods. This may be viewed as an adaptation of the famous Turing test for artificial intelligence, which was designed to address the question "Can machines think?" The Turing test provides an elegant approach to this question by focusing solely on the black box behavior of the machine, thus avoiding many subtle issues around the concept of "thinking." It simply asks: is the black box behavior of the given machine indistinguishable from that of a human being? In our approach to quantum testing, which we call a "quantum Turing test," the input-output behavior of a given quantum annealer is compared to that of our classical benchmark. The machine is said to fail the test if their behaviors are nearly indistinguishable.

Since quantum annealing does not have a well-developed theory of quantum fault-tolerance nor a rigorous analysis of quantum speedup, it is critically important to test that these machines actually work as claimed. The challenge has become even more pressing starting with the announcement by the Canadian company D-Wave of its 108-qubit quantum annealer in

2011, its subsequent scaling and claims and counterclaims about the speedups achieved by these [25, 24, 93, 23, 43]. These recent developments have led such eminent institutions as NASA and Google to start researching into the technology, rendering the testing of quantum annealers as one of the most important subjects in experimental quantum computing today.

We demonstrate the effectiveness of our approach in testing both for quantum coherence and quantum speedup by applying it to the D-Wave machine. In contrast to previous comparisons of the machine to simulated annealing, the results of our quantum Turing tests do not show evidence of quantum coherence or quantum speedup in the D-Wave machine. A more detailed analysis of the new classical model yields further algorithmic insights into the subtle nature of quantum annealing.

1.2 Tensor network nonzero testing

A central problem in quantum computing, the local Hamiltonian problem asks whether the ground state of a given local Hamiltonian has small energy or not. It is famously complete for the complexity class **QMA** [78], which can be understood as the quantum analogue of **NP** – instead of asking for a classical witness that can be verified by a Turing machine in polynomial time, **QMA** asks for a quantum witness that can be verified by a quantum computer in polynomial time. Moreover, it is generally believed that **QMA** is a strict superset of **NP**, and indeed that there is no subexponential size classical witness for **QMA**-complete problems. This means that ground states of local Hamiltonians generally do not have an efficient classical description.

Hence, it is interesting and enlightening to identify "easy" special cases of the local Hamiltonian problem which can be placed in **NP** and thus allow for efficient classical verification of the ground state. We note that one such example is the classical constraint satisfaction problem, which can be obtained by restricting the local terms in a quantum Hamiltonian to be diagonal in the standard basis. A perhaps more interesting example is proposed in [32], which considers local Hamiltonian problems whose terms pairwise commute. Such problems lie between the constraint satisfaction problem and the local Hamiltonian problem, and may provide important insights into the nature of the gap between **NP** and **QMA**. While it has been shown that 2-local commuting Hamiltonians [32] and certain special cases of 3-local and 4-local commuting Hamiltonians [2, 96] are in **NP**, these results do not seem to generalize to k-local Hamiltonians with $k > 4$.

In this thesis, we show that an interesting variant of the commuting local Hamiltonian problem called commuting stoquastic quantum k-SAT is in **NP** for any $k = O(\log n)$. On the one hand, our results can be viewed as confirming the intuition that stoquastic Hamiltonians are free of the so-called "sign problem" [33] and therefore more classical. On the other hand, stoquastic Hamiltonians include highly nontrivial quantum systems such as the toric code [44].

This result follows from a more general complexity theoretic treatment of tensor network nonzero testing, a fundamental problem in quantum Hamiltonian complexity. Tensor net-

works, which provide a graphical way to understand various linear algebraic computations, have been increasingly popular in condensed matter physics as a data structure for encoding various quantum objects. In particular, many special classes of tensor networks [88, 113, 114] have been studied in the literature in order to describe ground states of local Hamiltonians. While computing the value of a given tensor network is known to be #**P**-complete [98], it has been pointed out that tensor networks with certain desirable structures can be efficiently computed on a classical computer (most notably [88, 114]). Indeed, identifying such efficiently computable classes of tensor networks is nowadays an important part of condensed matter physics research, such efficiency being essential to simulating and computing interesting physical properties of a given material. In this thesis, we instead focus our attention on the seemingly easier problem of testing whether a given tensor network encodes a nonzero value. If it turns out to be significantly easier than the computation of tensor networks, it could result in a new approach to dealing with such high complexity quantum Hamiltonians.

Towards this goal, we establish the following connection between tensor network nonzero testing and the commuting local Hamiltonian problem: if tensor network nonzero testing is in **NP**, the commuting local Hamiltonian problem is also in **NP**. Note that this implication could potentially be used to enable the classical verification of ground states of such Hamiltonians. Unfortunately, our results suggest that tensor network nonzero testing in its most general form is computationally very hard, i.e., not contained in the polynomial hierarchy unless the hierarchy collapses. On the other hand, we are able to identify two "easy" special cases of tensor network nonzero testing, namely nonnegative tensors and injective tensors, which may be useful in certain contexts. Indeed, we demonstrate this last point by translating the special case of nonnegative tensor networks (i.e. tensor networks whose entries are all nonnegative real numbers) in light of the above connection between tensor network nonzero testing and the commuting local Hamiltonian problem. We first show that nonzero testing of nonnegative tensor networks is in **NP**, which immediately implies that commuting stoquastic quantum k-SAT is also in **NP**. We expect that identifying other easy special cases of tensor network nonzero testing may lead to novel techniques for classical verification of other quantum problems.

1.3 Outline of the thesis

The results presented in this thesis are highly interdisciplinary, and as such, their context is as important as their content. To put them in proper context, a detailed exposition of the background is provided in Chapter 2. Those readers with prior exposure to these subjects are advised to proceed directly to the main results presented in the subsequent chapters.

In Chapter 3, we present a classical benchmark for quantum annealers and use it in a kind of quantum Turing test. We argue that although relying exclusively on classical simulation methods, our approach provides an effective way of testing whether a given machine achieves nontrivial quantum coherence and whether it achieves a quantum speedup. We demonstrate

the effectiveness of this approach by applying it to experimental data from the D-Wave machine.

In Chapter 4, we show that commuting stoquastic quantum k-SAT is in **NP** for any $k = O(\log n)$. We note that this makes it the first variant of the commuting local Hamiltonian which is shown to be in **NP** for all constant k. The result follows from a study of the computational complexity of the problem of testing whether a given tensor network is nonzero. While the problem in full generality unfortunately turns out to be very hard, we show that certain special cases are significantly easier and immediately imply the above result about commuting stoquastic quantum k-SAT.

Chapter 2

Background

2.1 Quantum computing: the new limit of quantum mechanics

Since quantum computing first emerged as a field in the 90's, it has attracted a great deal of attention from the academics and the masses alike. Among the many alternative approaches to computing [119, 20, 47, 105, 7, 35, 86], quantum computing stands out as the most surprising because of its potential to violate the Extended Church-Turing Thesis, a central conjecture in the theory of computing. The thesis posits that every model of computation can be efficiently simulated by a probabilistic Turing machine, with only a polynomial overhead in the time and space complexity. Early research in the theory of quantum computing produced a sequence of results that seemingly contradicted the conjecture [21, 103, 102, 64], culminating in Shor's algorithm for prime factoring [102], which has an exponential advantage over the best known classical algorithm for the same problem.

On the other hand, the experimental realization of a quantum computer has proved to be an extremely challenging task over the last two decades, and even the most optimistic estimates for realization of this goal still place it at least a decade away [58]. The main technical hurdle in implementing a quantum computer is presented by the intrinsic volatility of quantum information. While quantum systems are in general allowed to exist in a superposition of many classical states, they are also forced to "collapse" into exactly one of those classical states at the time of a measurement. To make things worse, once the system gets measured, the information about the original superposition is irreversibly destroyed. Interestingly, this is somehow related to the fact that we never seem to observe a quantum mechanical phenomenon in our everyday life; it turns out that it is as though every system, unless specially protected, is getting measured at every instant by its environment and any quantum coherence there may be is thus destroyed. This phenomenon, which is the main obstacle in experimental quantum computing, is called decoherence.

The most fundamental task in experimental quantum computing is perhaps that of building a robust quantum bit, or a **qubit** – a unit memory cell that could store one bit of

quantum information. The great technological advances in the last two decades provided us with many different ways of implementing a qubit, such as ion trap qubits, superconducting qubits, NMR qubits, optical qubits, topological qubits, etc. While all of these technologies fall short of achieving scalable quantum computation as yet, they offer different sets of advantages and disadvantages. For instance, superconducting qubits can implement elementary gate operations on the order of nanoseconds, but they also take only a few microseconds to decohere and lose the stored quantum information. On the other hand, ion traps can take as long as 50 seconds to decohere [66], but have scaling issues for larger numbers of qubits.

In general, techniques such as quantum error correction [84, 44, 39] are sought in order to achieve fault-tolerance for practical systems, because it will often be infeasible to finish the desired computation within microseconds or even 50 seconds. Quantum error correction codes such as the surface code (see e.g. [53]) can in principle protect an arbitrary amount of quantum information for an arbitrarily long time, but cannot be implemented at the present time because of the unmet quality requirements that these codes impose on the physical qubits used to implement them. Building a qubit that meets these requirements is the key to scalable quantum computing, which many researchers are hoping to achieve over the next decade.

In the meantime, even in the absence of an experimental realization of a quantum computer, the theory of quantum computing has already made a successful landing in computational complexity theory and cryptography. One reason that quantum computing could have such a big impact in these theoretical fields is perhaps that the most famous problem that it solves efficiently, namely factoring, is enormously important in the theory of computing. Firstly, it is one of the few problems which are neither known to be in **P** nor known to be **NP**-complete. Moreover, the conjectured hardness of factoring is the very basis of the RSA cryptosystem, which is one of the most widespread cryptosystems today. Conversely, this means that a quantum computer will be able to easily break into all systems that are secured by the RSA cryptosystem, which also motivates the study of quantum-resistant cryptography. On the other hand, factoring is certainly not the only problem for which the quantum computer will be useful. For instance, Grover's quantum algorithm for unstructured search [64] immediately yields a quadratic speedup over classical brute-force algorithms for NP-complete problems. This means that quantum computers will be able to solve satisfiability in $O(2^{n/2})$ time, violating the strong exponential time hypothesis, another important conjecture in complexity theory.

Whence comes this enormous computing power of quantum computers? The superposition principle of quantum mechanics states that if a system is capable of being in any one state from a given set of states (e.g. the electron in a hydrogen atom can be either in the ground state or the first excited state), then it is also capable of being in any superposition of those states (e.g. superposition of the groundstate and the first excited state). Interpreting the possible classical states of the system as orthogonal vectors in a vector space, we find that possible states of a quantum system form a complex vector space (i.e. Hilbert space) of dimension equal to the number of possible classical states of the system. In the case of a quantum n-bit system, its possible states will form a Hilbert space of dimension 2^n, because

the classical n-bit system have 2^n possible states.

This means that we can, for example, prepare the quantum system of n bits in the uniform superposition of all 2^n possible classical states. Moreover, if we now apply quantum versions of binary logic gates on this state, the effect is as though we are simultaneously applying the given logic circuit on all 2^n classical states. This inherent parallelism is certainly a source of the computational advantage of quantum systems, but it alone cannot achieve any useful speedup because of the limitations posed by the measurement principle of quantum mechanics.

Roughly speaking, the measurement principle states that when a quantum system gets measured, the system randomly collapses into exactly one of the possible classical states, with probability proportional to the magnitude squared of the coefficient of that classical state in the superposition. Hence, we realize that our scheme above of using the uniform superposition of all 2^n classical states fails to be effective as is. Even though the inherent parallelism of quantum systems allows us to perform computation on all 2^n classical inputs simultaneously, we are eventually only able to read out only one of the 2^n outputs, so the situation is not very different from sampling a random input and performing computation on it. The success of quantum algorithms, such as Shor's factoring algorithm, depends crucially on the fact that the coefficients in a quantum superposition are complex numbers, which allows cancellations to take place through constructive and destructive interferences. For certain problems, such as factoring, we are able to devise a quantum circuit to ensure that the coefficients of the undesired outputs are canceled out, whereas the coefficients of the desired outputs are amplified. This allows us to extract information from our quantum system much more efficiently, and in some cases results in a significant speedup over classical computation.

If one were to single out one quantum mechanical phenomenon that makes quantum computation possible, it would be entanglement. A distinguishing feature of quantum mechanics, entanglement refers to the fact that certain superposition states of a many-part quantum system do not allow for individual state description of its parts. This is a radical violation of the central tenet of classical mechanics (and of our intuition), which posits that the universe is a collection of particles and the description of individual particles constitutes the description of the universe. In contrast, entangled quantum systems exist only as a whole, so to speak, and it is impossible to speak of the individual particles without also referring to the whole. Entanglement is the source of such surprising technologies as quantum computation, quantum teleportation, quantum cryptography, etc., and it is not difficult to show that none of these is possible with only unentangled quantum states.

Moreover, while being one of the most interesting features of quantum mechanics, entanglement remains also the least understood part of it. Although it has been extensively verified that entanglement can occur in a small number of particles (see e.g. [54]), if we scale the system up to more than a hundred particles, we immediately enter a regime which has almost entirely evaded the grasp of experimental physics (the only exception being [80], to our best knowledge). On the other hand, this high-entanglement regime is precisely the one in which quantum computers are expected to function. Aharonov and Vazirani [38] suggest

that quantum computation should therefore be viewed also as a test of quantum mechanics in "the limit of high (computational) complexity."

In fact, quantum computation has always had a close relationship with experimental physics. One of the first motivations for quantum computation came from physics when e.g. Feynman observed in 1981 [51] that Turing machines will not be able to simulate quantum physics efficiently. This poses a significant challenge to the physicist, as it means that the standard scientific method of hypothesizing, experimenting, and verifying can no longer be applied – the computational complexity of our hypothesis is so large that even the best computers in the world cannot handle it. The physicist's need for a quantum computer is even more pronounced today, when developing classical algorithms that can efficiently simulate special cases of many-body quantum systems is an essential part of physics research. While classical heuristics such as Density Matrix Renormalization Group [118] have enjoyed great success for certain classes of quantum systems, even a small quantum computer consisting of a few hundred qubits may already make possible many simulations that were inaccessible on a classical computer.

On the one hand, quantum physics is complex and intractible. On the other hand, as Aharonov and Vazirani [38] point out, the part of quantum mechanics that has been experimentally verified so far is all of low complexity. Hence, quantum computation should perhaps be thought of as a fundamental physics experiment which aims to test, for the first time, the high complexity aspect of quantum mechanics. Will the theory of quantum mechanics stand this test? Either way, the resolution of this question will result in a major breakthrough in at least one of two fields. If the answer is yes, we will have built a quantum computer, the machine that promises enormous speedups over today's computers. Otherwise, our understanding of quantum physics will have to be significantly revised.

2.2 Three principles of quantum mechanics

For those readers without prior exposure to quantum mechanics, we briefly introduce the most basic principles of quantum mechanics, along with the notation that will be followed throughout this thesis. Those who wish to get a deeper understanding of quantum mechanics and quantum computation are invited to consult standard textbooks and online courses [84, 95, 63, 111].

1. **Principle of Superposition**: If a system can be in one of n classical states, it can also be in a superposition of those states.

   In the standard formulation of quantum mechanics, the n possible classical states of the given system are represented by an orthogonal basis of a complex vector space (Hilbert space) of dimension n. This basis is called the standard basis, and its constituent vectors are denoted by $|0\rangle, |1\rangle, \ldots, |n-1\rangle$.

   A superposition of classical states can then be represented by a unit vector in this Hilbert space, which may have more than one nonzero classical component. Most

generally, an arbitrary state $|\psi\rangle$ of an n-dimensional quantum system can be written in the following form:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{n-1}|n-1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{pmatrix}.$$

Here, α_i's are complex numbers and we assume that $|\psi\rangle$ is normalized, i.e. $\sum_i |\alpha_i|^2 = 1$. We note that in quantum mechanics, the global complex phase of a state is considered an irrelevant quantity, i.e., if $|\psi\rangle = e^{i\theta}|\phi\rangle$ for some $\theta \in \mathbb{R}$, the two states $|\psi\rangle$ and $|\phi\rangle$ are considered to be physically identical.

The direct quantum analogue of the classical bit is the **qubit**, which is nothing but a two-dimensional quantum system. In contrast to the classical bit which has only two possible states, the qubit has infinitely many possible states, as its state can be described by any one of the infinitely many unit vectors in the two-dimensional Hilbert space.

The state of a composite quantum system is defined as the tensor product of the states of its constituent parts. For instance, suppose we have a quantum system consisting of two qubits, whose states are $a|0\rangle + b|1\rangle$ and $c|0\rangle + d|1\rangle$. The state of the whole system is then given by

$$(a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) = ac|0\rangle \otimes |0\rangle + ad|0\rangle \otimes |1\rangle + bc|1\rangle \otimes |0\rangle + bd|1\rangle \otimes |1\rangle = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix},$$

where the standard basis of the two-qubit system consists of $|0\rangle \otimes |0\rangle, \ldots, |1\rangle \otimes |1\rangle$, which are often abbreviated $|00\rangle, \ldots, |11\rangle$. Note that an n-qubit system will be of dimension 2^n and its standard basis will consist of all possible classical states of an n-bit system, i.e. $|\overbrace{00\cdots0}^{n}\rangle, \ldots, |\overbrace{11\cdots1}^{n}\rangle$.

Here, it is important to note that while the state of a composite system can always be determined from the states of its constituent parts, the reverse procedure is not always possible. For example, it is easy exercise to check that there are no one-qubit states $a|0\rangle + b|1\rangle$ and $c|0\rangle + d|1\rangle$ such that $(a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. This means that when a two-qubit system is in the state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, there is no way to describe the state of each individual qubit separately. The two qubits are **entangled**, and it can be said that at this point they exist only as a whole. On the other hand, states which can be described as a tensor product of the individual states of its constituent parts are called **product states**. For example, $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle = |0\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$ is a product state.

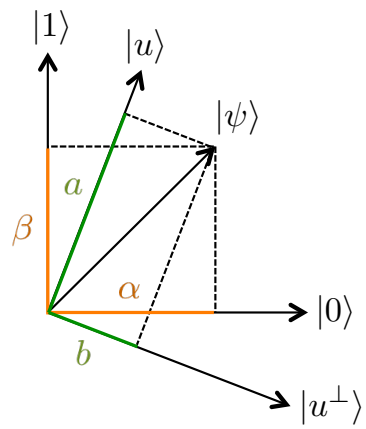

Figure 2.1: Rewriting a quantum state in two different bases. For convenience in illustration, we are assuming that $|\psi\rangle, |u\rangle, |u^{\perp}\rangle$ are real vectors.

The phenomenon of entanglement plays a crucial role in quantum computation. A main reason that we are unable to simulate general quantum systems on a classical computer is that we require exponential space to even describe the state of a quantum system. For example, we needed to store 2^n complex numbers to describe the state of an n-qubit system, a drastic increase from n bits of memory required by a classical n-bit system. In stark contrast, an unentangled n-qubit state only requires linear space to describe, as it suffices to specify two complex numbers for each qubit. This phenomenon of entanglement is also central to the themes of this thesis and will be repeatedly touched upon in different contexts.

Last but not least, we emphasize that the quantum superposition signifies neither the indefiniteness of the state of the system nor our lack of knowledge of it. On the contrary, a quantum superposition is a definite state of the system which is completely known to us. In that sense, the randomness of the quantum superposition is fundamentally different from that of a classical ensemble.

2. **Principle of Measurement**: When a system gets measured, it randomly collapses into exactly one of the possible classical states with probability equal to the magnitude squared of the complex coefficient of each classical state.

   Thus, were we to measure a one-qubit system in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the outcome of the measurement will be 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$. Moreover, the state of the system has now collapsed into $|0\rangle$ or $|1\rangle$, depending on which outcome we saw in the measurement.

   In principle, a quantum measurement can be made in an arbitrary orthogonal basis, and there will be an orthogonal basis corresponding to each physical quantity that

we can measure on the system, such as position, momentum, energy, spin, etc. The possible outcomes of the measurement can be understood as corresponding to the vectors in the given orthogonal basis. E.g., in the previous example, the basis of choice was the standard basis, so the potential outcomes were either 0 or 1, corresponding to $|0\rangle$ and $|1\rangle$ respectively. The probability of seeing each outcome is equal to the projection squared of the state of the system on the corresponding basis vector. Hence, if we were to measure the one-qubit system in the previous example in an arbitrary orthogonal basis $\{|u\rangle, |u^\perp\rangle\}$, we would first rewrite the given state in this basis so that $|\psi\rangle = a|u\rangle + b|u^\perp\rangle$, and find that the probability of seeing the outcome u and $u^\perp$ is equal to $|a|^2$ and $|b|^2$ respectively. Similarly to before, after the measurement, our system will have collapsed to the basis vector corresponding to the outcome that we saw in the measurement.

To formalize the notion of a measurement, we define what is called an observable in quantum mechanics. The observable specifies the orthogonal basis to be used in the measurement, along with real numbers which are to be interpreted as actual measurement outcomes, e.g. the numbers we read off of our measurement apparatus. As an example, $\{(|0\rangle, 1), (|1\rangle, -1)\}$ would be an observable for a one-qubit system that corresponds to a standard basis measurement. Note that there are many observables that correspond to the same orthogonal basis, due to the freedom of choice of accompanying real numbers. For purposes of computing the probability of each outcome, these real numbers have no relevance. On the other hand, these real numbers will often have important physical interpretation, e.g., in an energy observable they would tell us what the energy of the system is depending on the measurement outcome.

It is convenient to think of the orthogonal basis vectors and the real numbers specified by an observable as eigenvectors and eigenvalues of a matrix. Since this matrix has orthonormal eigenvectors and real eigenvalues, it will be Hermitian (i.e. $A = A^\dagger$). For example, the above-mentioned one-qubit observable can be written as the Hermitian matrix

$$1 \cdot |0\rangle\langle 0| + (-1) \cdot |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

where $\langle u|$ denotes the conjugate tranpose of $|u\rangle$ and hence $|u\rangle\langle u|$ denotes the rank-1 projection onto the vector $|u\rangle$. Most generally, we can think of any given $n \times n$ Hermitian matrix as an observable for an n-dimensional quantum system, and vice versa. What happens if there are repeated eigenvalues in the given Hermitian matrix? The answer is simple; the probability of seeing a repeated eigenvalue as the outcome of the measurement will be equal to the projection squared of the state of the system onto the corresponding *eigenspace*, and the state of the system after seeing this outcome will be the projection itself, appropriately renormalized.

The measurement principle also clarifies why a quantum superposition state reflects neither the indefinite nature of the system nor our lack of knowledge about the state of the system. While the state $|0\rangle$ appears to be a most definite state in the classical

viewpoint, if we measure this state in the $\{\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\}$ basis, the measurement outcome will be as indefinite as it can be – it will be uniformly random. On the other hand, while the state $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ appears to be a most indefinite state in the classical viewpoint, it is a completely definite state with respect to the above measurement. This shows that in quantum mechanics, a state which is most definite with respect to one observable can be highly indefinite with respect to another observable and vice versa, which is the very content of Heisenberg's famous uncertainty principle.

3. **Principle of Unitary Evolution**: Besides measurements, the time evolution of a quantum system is described by a unitary rotation.

   Most precisely, the time evolution of a quantum system is governed by the famous Schrödinger's equation, which depends on the energy observable H of the given system. It is outside of the scope of this thesis to discuss the details of the time evolution of a quantum system. However, we will point out that any time evolution described by Schrödinger's equation can be represented by a unitary matrix (i.e. $UU^\dagger = U^\dagger U = I$) acting on the Hilbert space in which the quantum state resides. Intuitively, this means that quantum systems evolve by a rigid-body rotation of the Hilbert space which preserves inner products between vectors. Moreover, the theory of universal gate sets in quantum computation [84] shows that in principle any given unitary rotation can be implemented using Schrödinger's equation by appropriately engineering the energy observable of the system.

2.3 Complexity of quantum systems: the local Hamiltonian problem

The energy observable, which determines the time evolution of a quantum system through Schrödinger's equation, is known as the **Hamiltonian** and is usually denoted by the letter H. In this section, we consider Hamiltonians of the n-qubit quantum system, which are central objects in the theory of quantum computing.

While we explained in the previous section that observables of the N-dimensional quantum system are described by $N \times N$ Hermitian matrices, when N is very large, it becomes infeasible to think about all such matrices. For instance, the dimensionality of the n-qubit quantum system is $N = 2^n$, which already exceeds estimates for the number of elementary particles in the observable universe at $n = 500$ [87]. Such observables are not only impossible to compute with, but also unlikely to be ever implemented. Therefore, it is reasonable to restrict our attention to a subclass of Hermitian matrices which admit a more succinct description.

A very useful and natural restriction that we could impose on our Hamiltonians is that of locality. This means that the energy of a given physical system will depend entirely upon local features of the state of the system. For example, imagine that we have n qubits placed

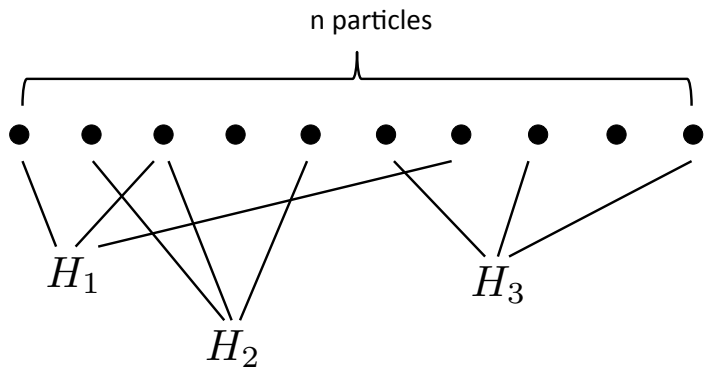

Figure 2.2: A schematic representation of a 3-local Hamiltonian $H = \sum_i H_i$. Each term H_i acts on at most three particles at a time.

on a one-dimensional chain, each of which is allowed to have a spin of either $+1$ or -1. However, these qubits interact with each other in such a way that neighboring qubits want to be aligned with each other, e.g., if a neighbor of a given qubit has the spin of $+1$, it should be energetically favorable for the given qubit to also have the spin of $+1$. We can succinctly represent the Hamiltonian H of such a system as a sum of small observables H_i's, each of which only measures two qubits i and $i+1$ and assigns an energy penalty of c if and only if they are misaligned. For instance, the two-qubit observable that achieves this can easily be written as $c|01\rangle\langle 01| + c|10\rangle\langle 10|$. Then, H_i for $1 \leq i \leq n-1$ can be obtained by taking the tensor product

$$H_i = I^1 \otimes \cdots \otimes I^{i-1} \otimes (c|01\rangle\langle 01| + c|10\rangle\langle 10|)^{i,i+1} \otimes I^{i+2} \otimes \cdots \otimes I^n,$$

where the superscripts denote which qubits are acted on by each matrix. Hence, while H_i is still technically a $2^n \times 2^n$ matrix, it acts nontrivially upon only two qubits at a time and can be succinctly described by specifying its action on those two qubits. Then, the system Hamiltonian H will simply be defined as $H = \sum_{i=1}^{n-1} H_i$.

More generally, we can define k-local Hamiltonians as follows.

Definition 2.1. A Hamiltonian H for an n-particle quantum system is said to be k**-local** if it can be written as $H = \sum_{i=1}^{m} H_i$ where each H_i acts nontrivially upon at most k particles at a time.

Note that in contrast to general Hamiltonians, k-local Hamiltonians require only polynomial space to describe. This is because there are only $\binom{n}{k} = O(n^k)$ ways to choose k particles out of n particles, and each term that acts on k particles at a time is fully specified by a $d^k \times d^k$ complex matrix, where d is the dimensionality of each individual particle (e.g. for qubits, $d = 2$). Moreover, Hamiltonians that naturally arise in physical systems tend to be well modeled by k-local Hamiltonians for a suitable choice of constant k.

To the computer scientist, the structure of the k-local Hamiltonian is a familiar one. Namely, it is straightforward to draw a parallel between the notion of locality in constraint satisfaction problems such as MAX-3-SAT and the notion of locality in local Hamiltonians. For instance, in the MAX-3-SAT problem we are given m Boolean clauses that each concern up to three variables and are asked to find a Boolean assignment to the n variables that minimizes the number of violated clauses. Similarly, we can imagine being given a 3-local Hamiltonian, i.e., m local observables that each concern up to three qubits, and being asked to find a quantum state on all n qubits that minimizes total energy. In fact, any MAX-3-SAT instance can be translated into an energy minimization problem on a 3-local Hamiltonian as follows. First, note that for each clause i in a MAX-3-SAT instance, there is exactly one assignment to its three variables that violates the clause. For example, if the given clause was $x_1 \vee \neg x_2 \vee x_5$, the only assignment to x_1, x_2, and x_5 that violates the clause, out of the eight possible assignments, would be $x_1 = 0, x_2 = 1, x_5 = 0$. Hence, we can naturally replace this clause with a 3-qubit observable $|010\rangle\langle 010|^{1,2,5}$ that acts on qubits 1, 2, and 5. Taking the tensor product of this matrix with identity matrices on all other qubits, we obtain our 3-local Hamiltonian term H_i. Now we define our final Hamiltonian to be $H = \sum_{i=1}^{m} H_i$, so that the minimum energy of the system is given by $E_0 = \min_{|\psi\rangle} \langle\psi|H|\psi\rangle$. Since it is clear by construction that the standard basis is the simultaneous eigenbasis of every H_i, at least one standard basis vector $|x\rangle$ must be an eigenvector for the eigenvalue E_0. Moreover, $\langle x|H|x\rangle$ is simply equal to the number of MAX-3-SAT clauses violated by the assignment x. We conclude that finding the minimum energy of H is equivalent to finding the solution to the given MAX-3-SAT instance.

The obvious generalization of the above reduction shows that any instance of MAX-k-CSP is in fact also an instance of energy minimization of a k-local Hamiltonian. The latter problem is simply known as the k-local Hamiltonian problem in quantum computing literature, and is formally defined as follows:

Definition 2.2. The k-local Hamiltonian problem is defined as the following promise problem:

- **Input**: A k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ on n particles such that $m = \text{poly}(n)$ and $||H_i|| \leq 1$ (where $||\cdot||$ is the operator norm), numbers a and b such that $b - a > 1/\text{poly}(n)$.
- **Output**:
  - YES if the smallest eigenvalue (minimum energy) of H is at most a,
  - NO if the smallest eigenvalue of H is at least b.

With the choice of $a = 0$ and $b = 1$, the above reduction from MAX-3-SAT immediately implies the following:

Theorem 2.1. The k-local Hamiltonian problem is **NP**-hard.

However, even at a first glance it seems that this problem should indeed be much harder than **NP**-complete problems. To see this, we recall that YES instances of **NP**-complete problems must have polynomial witnesses which can be efficiently verified by a Turing machine. Do local Hamiltonian problems admit any such witness? Since the local Hamiltonian problem seeks to decide whether there exists a state whose energy is at most a, a natural candidate for a witness of this fact is the state itself. This is rather troublesome, however. As we pointed out earlier, a quantum state of an n-qubit system requires exponential space to even describe, let alone verifying it or finding it. Could there be a different kind of witness that can be efficiently described and verified? While the answer to this question is not known, the general consensus seems to be that the answer is no.

On the other hand, we remark that the above choice of witness is already sufficiently good for quantum computers. If we were working with a quantum computer, a witness that requires n qubits to describe is in fact already an efficient witness. Moreover, it turns out that there is a procedure by which the quantum computer can verify that this witness state indeed has small energy [4, 78]. This immediately places the k-local Hamiltonian problem in a complexity class called **QMA**, which can be thought of the quantum analogue of **NP**; instead of requiring a polynomial witness that can be efficiently verified by a Turing machine, **QMA** requires a polynomial quantum witness (i.e. the number of qubits is polynomial) that can be efficiently verified by a quantum computer.

In a remarkable result, Kitaev proves that the k-local Hamiltonian problem is in fact also complete for the class **QMA** [78, 4, 56], which makes it one of the most important problems in the quantum computing literature. The k-local Hamiltonian problem to the theory of quantum computing is what satisfiability is to the theory of computing.

2.4 Quantum Hamiltonian complexity: connections to condensed matter physics

Since local Hamiltonians are objects that originated in physics, they also tend to play an important role in fundamental physics research. At the highest level, Hamiltonians are observables that allow us to measure the energy of a given state. Of course, a measurement of a quantum system generally yields a random outcome, so it is difficult in general to speak about the energy of the given state itself. On the other hand, it is often useful to think about e.g. its expectation. In our notation, the expectation of the energy of a given state $|\psi\rangle$ can simply be computed as:

$$\mathbb{E}_{|\psi\rangle}[H] = \langle\psi|H|\psi\rangle =: \langle H\rangle.$$

This is because if we write $H = \sum_i E_i|\phi_i\rangle\langle\phi_i|$ where E_i's and $|\phi_i\rangle$'s are the eigenvalues and eigenvectors of H, then

$$\mathbb{E}_{|\psi\rangle}[H] = \sum_i E_i|\langle\psi|\phi_i\rangle|^2 = \sum_i E_i\langle\psi|\phi_i\rangle\langle\phi_i|\psi\rangle = \langle\psi|\left(\sum_i E_i|\phi_i\rangle\langle\phi_i|\right)|\psi\rangle = \langle\psi|H|\psi\rangle,$$

where $\langle x|y\rangle$ is simply a shorthand for $\langle x| \cdot |y\rangle$. Moreover, if H happens to be a k-local Hamiltonian, i.e., $H = \sum_{i=1}^{m} H_i$ where each H_i acts nontrivially on at most k particles at a time, then

$$\langle H\rangle = \langle\psi|H|\psi\rangle = \langle\psi| \left(\sum_{i=1}^{m} H_i\right) |\psi\rangle = \sum_{i=1}^{m} \langle\psi|H_i|\psi\rangle.$$

Hence, if $|\psi\rangle$ were a product state, this quantity can be efficiently computed. To compute each term $\langle\psi|H_i|\psi\rangle$, we only need to sandwich a $d^k \times d^k$ matrix by two vectors of size d^k. Since both d and k are constant, the overall time complexity is linear in the number of terms in the given Hamiltonian.

The locality constraint in a Hamiltonian is particularly well-motivated in condensed matter physics, which is the study of condensed phases of matter such as solids, liquids, Bose-Einstein condensates, etc. When a given material is in a condensed phase, it is often natural to model the material as a 1D, 2D, or 3D lattice on which each node models a particle and each edge models interaction between neighboring particles. In such a model, longer-range interactions are considered negligible and dropped from consideration. Therefore, the resulting local Hamiltonian will, in addition to the CSP-like locality introduced in the previous section, also retain a kind of geometric locality imposed by this lattice structure. Namely, the particles in the system are thought of as residing on the nodes of a lattice of some dimensionality, and the system is assumed to be governed by a two-local Hamiltonian whose individual terms act nontrivially upon at most two *neighboring* particles on this lattice. Of course, in certain cases it will be appropriate to generalize this notion so that interaction is allowed up to some fixed distance, yielding k-local Hamiltonians.

Another assumption on the structure of local Hamiltonians that is frequently introduced by the condensed matter physicist is that of translational invariance. If the material that is being studied were homogeneous, it would be natural to expect that the interaction between its particles is uniform across the whole material. For local Hamiltonians defined on a lattice, this assumption can conveniently be formulated by requiring that the Hamiltonian is invariant under spatial translation. While finite lattices obviously cannot be translationally invariant in the strictest sense, a suitable notion of translational invariance can still be defined by imposing boundary conditions or by assuming that the lattice wraps around at its boundaries (e.g. we may think of a 2D lattice as a torus). For instance, translational invariance for 2-local Hamiltonians on a 2D lattice may be defined as the condition that requires all the vertical edges to have the same matrix form and all the horizontal edges to have the same matrix form, as in Figure 2.3.

An interesting question that naturally arises is whether the complexity theoretic results about the k-local Hamiltonian problem which we surveyed in the previous section extend to these apparently more restricted settings. Since we now seem to be working with a much smaller class of local Hamiltonians, could we hope that e.g. the 2-local Hamiltonian problem defined on a 2D lattice with the assumption of translational invariance would be efficiently solvable? If the answer is yes, it would mean that the condensed matter physicist can study the ground state (i.e. the lowest energy state) properties of the material on classical comput-

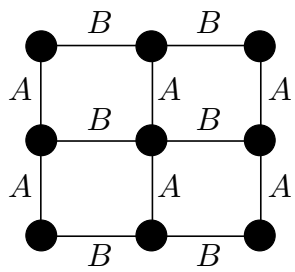

Figure 2.3: An illustration of translation invariance on a 3×3 2D lattice. If each node represents a d-dimensional particle, A and B will be $d^2 \times d^2$ matrices. For each vertical edge (i, j), we take the tensor product of the matrix A that acts upon particles i and j and the identity matrix that acts upon all other particles to obtain $H_{i,j}$. For each horizontal edge (i, j), we similarly take the tensor product of the matrix B that acts upon particles i and j and the identity matrix on all other particles. Our translationally invariant 2-local Hamiltonian will simply be defined as $H = \sum_{(i,j)\in E} H_{i,j}$.

ers. Unfortunately, a sequence of results from collaborative efforts of computer scientists and physicists [85, 22, 83, 76, 97, 6, 61, 65] vanquish any such hope. In particular, Aharonov, Gottesman, Irani, and Kempe [6] show that even the 2-local Hamiltonian problem defined on a 1D chain is **QMA**-complete for sufficiently large local dimension d, and Gottesman and Irani [61] show that the problem remains $\mathbf{QMA_{EXP}}$-complete (where $\mathbf{QMA_{EXP}}$ is the quantum analogue of **NEXP**) even with the assumption of translational invariance.

Of course, this does not imply that the ground states of local Hamiltonians are entirely inaccessible. While the above results do show that finding the ground state of a general Hamiltonian is intractable even with these additional assumptions, physicists have developed many special classes of Hamiltonians whose ground states can be computed or in some cases even analytically found. One such example is the class of *gapped* local Hamiltonians on a 1D chain. Namely, for a family of 1D local Hamiltonians $\{H_1, H_2, \ldots, H_n, \ldots, \}$ for which there is a constant gap between the smallest and the second smallest eigenvalues which does not decrease in n, a classical heuristic called Density Matrix Renormalization Group (DMRG) is known to be effective in computing the ground state [118].

We note that a basis of the DMRG algorithm is the longtime folklore in condensed matter physics which conjectures that the ground states of gapped local Hamiltonians have bounded entanglement [67], a proposition often referred to as the "area law." Moreover, it is well known that 1D states with bounded entanglement admit a polynomial description through a tool called matrix product states (MPS) [115, 88]. Hence, if the area law were true, we would be able to find the ground state of a given gapped 1D local Hamiltonian by searching for matrix product states that have low energy with respect to this Hamiltonian. DMRG is a method that achieves exactly this.

While DMRG has been hugely successful for 1D systems in practice for almost two

decades, the area law remained a conjecture for a long time until it was rather recently proved for 1D systems by Matt Hastings [67] and then significantly improved upon by Arad, Landau, and Vazirani [14]. Interestingly, the latter work uses different proof techniques from the former and heavily draws upon tools from theoretical computer science. In a follow-up work [79], a provably correct algorithm for computing the ground state of gapped 1D systems was presented for the first time, establishing that the problem of finding the ground energy is in **BPP** for this class of local Hamiltonians.

Very often, the condensed matter physicist is struggling with the fact that the object of their interest may be computationally intractable. Of course, the ground state, which may be thought of as the state of the system at zero temperature, is by no means the only object of their interest. They are interested in various quantities such as the state of the system at a finite temperature, expectations of local observables, time evolution of a given state, etc. In the absence of a quantum computer, their best bet is often to find scenarios in which there is a bound on the amount of entanglement in the system and develop classical methods that can exploit it. So there is a sense in which condensed matter physicists are doing computer science in disguise.

In this section, we have briefly summarized how the field of quantum computing and the field of condensed matter physics are meeting and exchanging ideas on their common ground which is the local Hamiltonian problem. This connection is increasingly being recognized as an emerging new field under the name of quantum Hamiltonian complexity. For a more comprehensive survey on this area, the reader is cordially referred to [56].

2.5 Tensor networks

In this section, we will introduce a mathematical tool called tensor networks, which play a central role in many of the aforementioned results in quantum Hamiltonian complexity. Tensor networks are particularly well suited for computing linear algebraic objects and can be thought of as a way of doing linear algebra with pictures.

2.5.1 Basic concepts

First, recall that a tensor is defined as a map from a fixed number of indices to complex numbers.

Definition 2.3. A **tensor** of rank k is a map from $[d_1] \times [d_2] \times \cdots \times [d_k]$ to $\mathbb{C}$, where $d_i \in \mathbb{Z}^+$ and $[x] := \{0, 1, \ldots, x-1\}$.

Then, we can formally define a tensor network as follows.

Definition 2.4. A **tensor network** T is a graph (V, E) where each vertex $v \in V$ is associated with a tensor $A_v : [d_{v,1}] \times [d_{v,2}] \times \cdots \times [d_{v,k}] \to \mathbb{C}$ whose rank is $k = \deg(v)$ and each edge $e \in E$ is associated with a positive integer d_e. Here, d_e is referred to as the

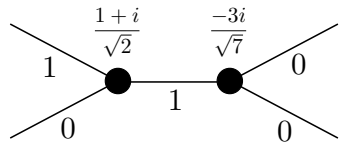

Figure 2.4: An example tensor network with two tensors of rank 3. Each tensor has two open edges and one closed edge, and it is assumed that every edge has a bond dimension of two. Labeling each edge with either 0 or 1, the input to each tensor is completely specified, so that it can output a complex number. The value of the tensor network at this labeling is simply defined as the product of these two complex numbers.

bond dimension of the edge e. Moreover, each vertex v is also associated with an ordering $\sigma_v : E(v) \to \{1, \ldots, k\}$ (where $E(v)$ denotes edges that are incident to v) on its k edges such that for all $e \in E(v)$, we have $d_{v,\sigma_v(e)} = d_e$.

In addition, this graph allows the following two unconventional features:

1. Two vertices can have arbitrarily many repeated edges.
2. An edge can be open-ended on one side, i.e., it can be incident to only one vertex. Such an edge is said to be **open** and denoted $(u, \cdot)$, where u is the single adjacent vertex of the edge. Edges that are incident to two vertices are said to be **closed**.

Definition 2.5. Let $T = (V, E)$ be a tensor network and $E = \{e_1, \ldots, e_m\}$. Given a labeling $(x_1, \ldots, x_m) \in [d_{e_1}] \times \cdots \times [d_{e_m}]$, the value of T at $(x_1, \ldots, x_m)$, denoted $T(x_1, \ldots, x_m)$, is defined as the product of all the complex numbers output by the tensors A_v in T when interpreting the labeling on each tensor's incident edges as input to the tensor.

While the above definitions appear rather technical, the crux of their content is only that each vertex on this graph can be interpreted as a tensor which takes a labeling on its incident edges as input. More precisely, if we were to label each edge $e \in E$ of a given tensor network by some integer in $[d_e]$, each vertex, taking these labels as input to their associated tensor, will output a complex number (see Figure 2.4). The value of the tensor network at the given labeling is defined as the product of these complex numbers.

In general, a tensor network with k open edges can itself be interpreted as a tensor of rank k via a process called contraction.

Definition 2.6. Let $T = (V, E)$ be a tensor network and let $E_O = \{e_1, \ldots, e_k\}$ and $E_C = \{e_{k+1}, e_{k+2}, \ldots, e_m\}$ denote the set of its open edges and closed edges respectively. Then we can define a tensor $A : [d_{e_1}] \times [d_{e_2}] \times \cdots \times [d_{e_k}] \to \mathbb{C}$ of rank k as follows:

$$A(x_1, \ldots, x_k) = \sum_{\substack{x_{k+1} \in [d_{e_{k+1}}] \\ x_{k+2} \in [d_{e_{k+2}}] \\ \cdots \\ x_m \in [d_{e_m}]}} T(x_1, \ldots, x_m).$$

Then, A is called the **contraction** of T.

In other words, the contraction of a tensor network T with k open edges is defined as a tensor of rank k whose output on any given input $x_1, \ldots, x_k$ is computed by

1. labeling the k open edges with $x_1, \ldots, x_k$, and then
2. taking the sum of the values of T at all possible labelings on its closed edges.

While these definitions of a tensor network and its contraction may seem rather unmotivated at a first glance, it turns out that these objects are capable of expressing almost every kind of computation that happens in linear algebra and quantum mechanics. To see this, we begin by observing that any given tensor can be interpreted as many different linear algebraic objects. For instance, consider a tensor $A : [2]^k \to \mathbb{C}$ of rank k. One obvious way of interpreting A is as a vector in a complex Hilbert space $\mathbb{C}^{2^k}$. Namely, we can interpret each input $(x_1, x_2, \ldots, x_k) \in [2]^k$ as representing the standard basis vector $|x_1 x_2 \cdots x_k\rangle$ in $\mathbb{C}^{2^k}$ and the corresponding output $A(x_1, \ldots, x_k)$ as the coefficient of that standard basis vector. In other words, we can think of A as representing the (potentially unnormalized) quantum state

$$|\psi_A\rangle = \sum_{x_1 x_2 \cdots x_k \in \{0,1\}} A(x_1, \ldots, x_k)|x_1 x_2 \cdots x_k\rangle = \begin{pmatrix} A(0, \ldots, 0, 0) \\ A(0, \ldots, 0, 1) \\ A(0, \ldots, 1, 0) \\ \vdots \\ A(1, \ldots, 1, 1) \end{pmatrix}.$$

On the other hand, A can also be viewed as a matrix of size $2^l \times 2^m$ for any $l, m \in \mathbb{Z}^+$ such that $l + m = k$, by interpreting $A(x_1, \ldots, x_k)$ as the matrix entry at the row indexed by $x_1, \ldots, x_l$ and the column indexed by $x_{l+1}, x_{l+2}, \ldots, x_{l+m}$. Namely,

$$M_A = \begin{pmatrix} A(\overbrace{0, \ldots, 0, 0}^{l}, \overbrace{0, \ldots, 0, 0}^{m}) & A(\overbrace{0, \ldots, 0, 0}^{l}, \overbrace{0, \ldots, 0, 1}^{m}) & \cdots & A(\overbrace{0, \ldots, 0, 0}^{l}, \overbrace{1, \ldots, 1, 1}^{m}) \\ A(\overbrace{0, \ldots, 0, 1}^{l}, \overbrace{0, \ldots, 0, 0}^{m}) & A(\overbrace{0, \ldots, 0, 1}^{l}, \overbrace{0, \ldots, 0, 1}^{m}) & \cdots & A(\overbrace{0, \ldots, 0, 1}^{l}, \overbrace{1, \ldots, 1, 1}^{m}) \\ A(\overbrace{0, \ldots, 1, 0}^{l}, \overbrace{0, \ldots, 0, 0}^{m}) & A(\overbrace{0, \ldots, 1, 0}^{l}, \overbrace{0, \ldots, 0, 1}^{m}) & \cdots & A(\overbrace{0, \ldots, 1, 0}^{l}, \overbrace{1, \ldots, 1, 1}^{m}) \\ \vdots & \vdots & \ddots & \vdots \\ A(\overbrace{1, \ldots, 1, 1}^{l}, \overbrace{0, \ldots, 0, 0}^{m}) & A(\overbrace{1, \ldots, 1, 1}^{l}, \overbrace{0, \ldots, 0, 1}^{m}) & \cdots & A(\overbrace{1, \ldots, 1, 1}^{l}, \overbrace{1, \ldots, 1, 1}^{m}) \end{pmatrix},$$

or equivalently, we can write $\langle x_1, x_2, \ldots, x_l | M_A | x_{l+1}, x_{l+2}, \ldots, x_{l+m}\rangle = A(x_1, \ldots, x_k)$.

Having seen these different linear algebraic interpretations of a tensor, we can even devise a kind of pictorial notation to indicate which of these interpretations we are adopting for each tensor, as in Figure 2.5. This kind of pictorial notation is in fact very useful because it allows us to perform various kinds of linear algebraic computations by merely juxtaposing the operand tensors and joining appropriate edges. Some examples include:

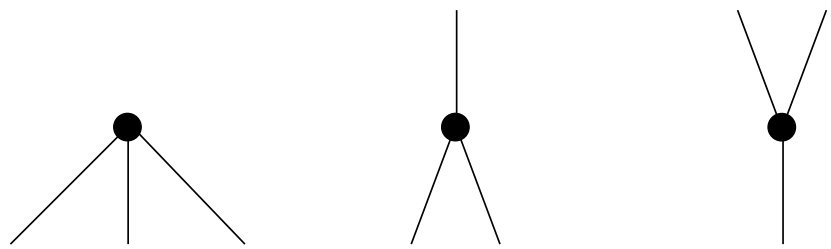

Figure 2.5: A pictorial illustration of how a tensor of rank 3 with bond dimension 2 on each edge can be interpreted as a 2^3-dimensional vector, a 2×2^2 matrix, or a $2^2 \times 2$ matrix. Namely, in this notation a tensor is interpreted as a linear map from a vector space whose basis vectors are indexed by its upper edges to another vector space whose basis vectors are indexed by its lower edges.

1. **Tensor product**

   One of the most basic operations that can be performed with tensor networks is that of tensor product. In fact, the tensor product of a sequence of two tensor networks $T_1, \ldots, T_m$ is given by a simple juxtaposition of them, i.e., the resulting tensor network includes all vertices and edges of $T_1, \ldots, T_m$ but there is no edge connecting T_i and T_j for $i \neq j$. For instance, Figure 2.6 shows how to take the tensor product of three one-qubit states $|\psi_1\rangle$, $|\psi_2\rangle$, and $|\psi_3\rangle$ to obtain a three-qubit state $|\psi\rangle$. The claim is that the contraction of the resulting tensor network yields a tensor that corresponds to the tensor product of the three given tensors. In fact, it is easy to verify that this is the case. On the one hand, if $|\psi_1\rangle = a_{1,0}|0\rangle + a_{1,1}|1\rangle$, $|\psi_2\rangle = a_{2,0}|0\rangle + a_{2,1}|1\rangle$, and $|\psi_3\rangle = a_{3,0}|0\rangle + a_{3,1}|1\rangle$, then by the rules of the tensor product we must have $|\psi\rangle = \sum_{i,j,k\in\{0,1\}} a_{1,i}a_{2,j}a_{3,k}|ijk\rangle$. On the other hand, if T_1, T_2, and T_3 are the tensor networks corresponding to $|\psi_1\rangle$, $|\psi_2\rangle$, and $|\psi_3\rangle$ respectively, by definition we have $T_i(x) = a_{i,x}$ for $i \in \{1, 2, 3\}$ and $x \in \{0, 1\}$. Now, if T is the tensor network obtained by juxtaposing T_1, T_2, and T_3, since T does not have any closed edge, its contraction A is defined as the tensor that maps its three inputs $i, j, k \in \{0, 1\}$ to the product $T_1(i)T_2(j)T_3(k) = a_{1,i}a_{2,j}a_{3,k}$, which is nothing but $|\psi\rangle$. Additionally, this shows that any product state of an n-qubit system can be described by a tensor network of n rank-one tensors each of which has an open edge of bond dimension two.

   It is straightforward to also verify that the tensor product of matrices can similarly be obtained as a simple juxtaposition of the tensor networks corresponding to each matrix.

2. **Inner product**

   We now show that the inner product of two vectors can also be performed using tensor

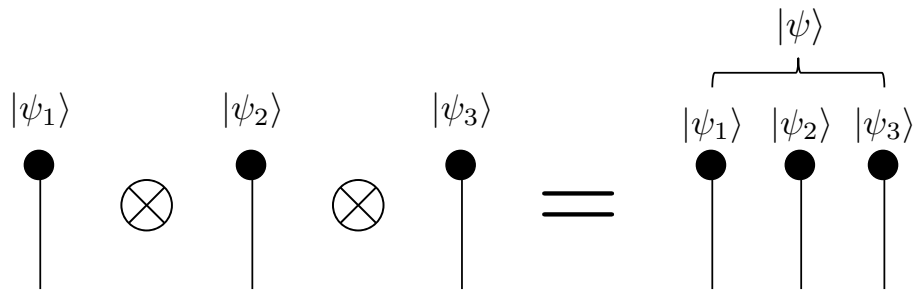

Figure 2.6: Tensor product of three vectors $|\psi_1\rangle$, $|\psi_2\rangle$, and $|\psi_3\rangle$.

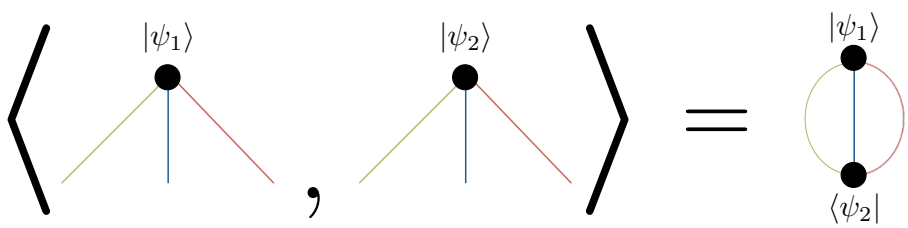

Figure 2.7: Inner product of two vectors $|\psi_1\rangle$ and $|\psi_2\rangle$. Note that prior to the joining of the edges, one of the vectors (in this case $|\psi_2\rangle$) needs to be modified so that the modified vector's outputs are complex conjugates of the original vector's outputs. This is why the bottom vector on the right-hand side is labeled $\langle\psi_2|$ instead of $|\psi_2\rangle$.

networks. Here, instead of merely juxtaposing the two given vectors, we also need to join the corresponding edges of the two vectors, as in Figure 2.7. To see this, suppose T_1 and T_2 are tensor networks corresponding to two vectors $|\psi_1\rangle = \sum_{x\in\{0,1\}^k} \alpha_x|x\rangle$ and $|\psi_2\rangle = \sum_{x\in\{0,1\}^k} \beta_x|x\rangle$ respectively and T is the tensor network obtained by performing the operation depicted in Figure 2.7 on T_1 and T_2. Since the resulting tensor T has no open edge, its contraction will yield a tensor of rank zero, i.e., a scalar. Moreover, this scalar is computed as

$$T = \sum_{i_1,\ldots,i_k\in\{0,1\}} T_1(i_1,\ldots,i_k)T_2(i_1,\ldots,i_k) = \sum_{i_1,\ldots,i_k\in\{0,1\}} \alpha_{i_1\cdots i_k}\beta^*_{i_1\cdots i_k} = \langle\psi_2|\psi_1\rangle,$$

which is exactly the inner product between $|\psi_1\rangle$ and $|\psi_2\rangle$.

3. **Matrix multiplication**

   Matrix multiplication is another operation which can easily be performed using tensor networks. Suppose we are given two tensor networks T_1 and T_2 corresponding to the two input matrices A and B. To obtain their product, we simply join the lower edges of T_1 to the corresponding upper edges of T_2 as depicted in Figure 2.8. For this to

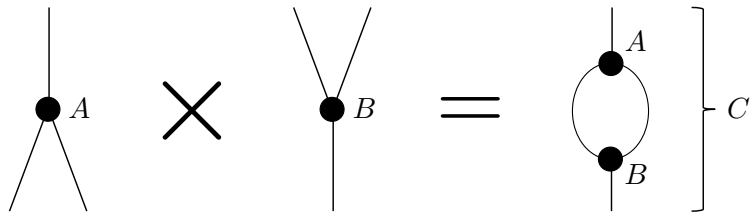

Figure 2.8: Matrix multiplication of a 2×2^2 matrix A and a $2^2 \times 2$ matrix B, which yields a 2×2 matrix C.

work, the number of lower edges of T_1 must be equal to the number of upper edges of T_2, and moreover the bond dimensions of the corresponding edges must exactly match. This condition is the equivalent of the linear algebraic condition that the number of columns of A must be equal to the number of rows of B. It is also straightforward to verify that this procedure yields the correct output AB. Assuming that T_1 is a tensor with n upper edges and m lower edges and T_2 is a tensor with m upper edges and k lower edges, and, for simplicity, that the bond dimension of every open edge is two, we note that the contraction of the tensor network T resulting from the above procedure is given by

$$\begin{aligned} T(i_1, \ldots, i_n, j_1, \ldots, j_k) &= \sum_{l_1,\ldots,l_m \in \{0,1\}} T_1(i_1, \ldots, i_n, l_1, \ldots, l_m) T_2(l_1, \ldots, l_m, j_1, \ldots, j_k) \\ &= \sum_{l_1,\ldots,l_m \in \{0,1\}} A_{i_1 \cdots i_n, l_1 \cdots l_m} B_{l_1 \cdots l_m, j_1 \cdots j_k} \\ &= (AB)_{i_1 \cdots i_n, j_1 \cdots j_k}. \end{aligned}$$

4. **Trace**

   Finally, we show how to compute the trace of a given square matrix A. The assumption here is that the tensor network T representing A has equal number of upper edges and lower edges, and that the bond dimensions of the corresponding upper and lower edges exactly match. Then, the trace of A can easily be obtained by joining each lower edge of T to its corresponding upper edge, as in Figure 2.9. Clearly, the resulting tensor network T' will have no open edge, and hence its contraction will yield a scalar which is computed as

$$T' = \sum_{i_1, i_2, \ldots, i_k \in \{0,1\}} T(i_1, \ldots, i_k, i_1, \ldots, i_k) = \sum_{i_1, i_2, \ldots, i_k \in \{0,1\}} A_{i_1 \cdots i_k, i_1 \cdots i_k} = \mathrm{Tr}(A),$$

   where we are assuming for simplicity that every open edge in T has bond dimension two and k denotes the number of upper edges in T.

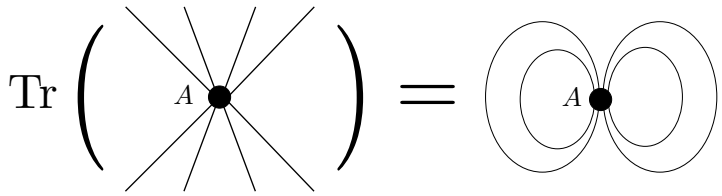

Figure 2.9: Trace of a $2^4 \times 2^4$ matrix A.

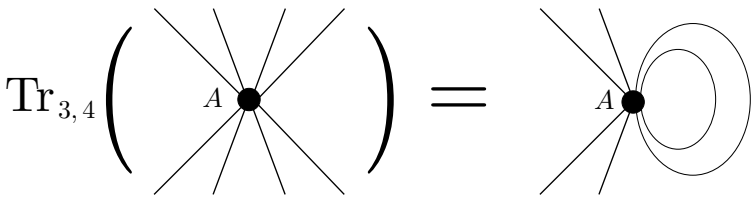

Figure 2.10: Partial trace of a $2^4 \times 2^4$ matrix A over the third and fourth particles.

We note that the partial trace of a matrix can similarly be computed, by only joining those edges that correspond to indices that are being traced out (see Figure 2.10).

Together, these primitives give us a way to express a wide range of linear algebraic computations as a tensor network. In particular, quantum mechanical objects such as states, operators, and observables can be expressed in such a way that the tensor product structure of the Hilbert space is well visualized, which is often helpful in recognizing important properties such as entanglement.

Another strength of the tensor network formalism is that while we may want to use the above primitives to initially construct a tensor network whose contraction encodes an object of our interest, once the tensor network is constructed, we are free to drop the existing interpretation of its component tensors and apply a different interpretation to get a new look at the desired object, or even treat it purely as a computational object. For example, as in Figure 2.11, given real matrices A and B we may construct a tensor network corresponding to $\mathrm{Tr}(AB)$ and then decide to interpret A and B as vectors instead. Since changing the pictorial representation of the vertices and edges does not affect the tensor network itself, we find that $\mathrm{Tr}(AB)$ must be equal to the inner product of the flattened versions of A and B (where flattening A means converting it into a vector by concatenating all its rows). Similarly, Figure 2.12 shows how the tensor network representation of $\mathrm{Tr}(ABC)$ immediately proves the well-known fact that $\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA) = \mathrm{Tr}(CAB)$.

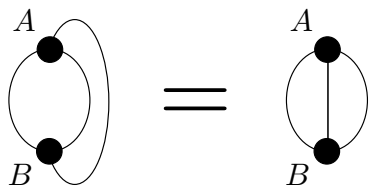

Figure 2.11: $\mathrm{Tr}(AB) = \langle \mathrm{flatten}(A), \mathrm{flatten}(B) \rangle$ for real matrices A and B.

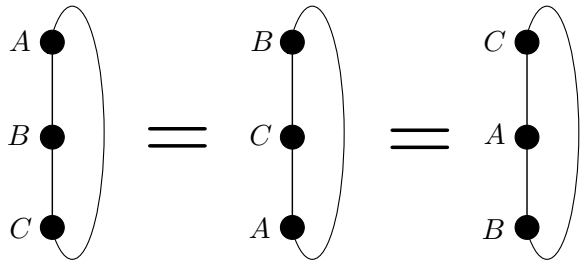

Figure 2.12: $\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA) = \mathrm{Tr}(CAB)$.

2.5.2 Computational aspects

A feature of the tensor network which is most useful for the computer scientist is that the geometric structure of a tensor network instantly reveals the feasibility of the encoded computation. This is particularly attractive in the context of quantum Hamiltonian complexity, where the tractability of linear algebraic objects is one of the most important issues.

We will presently outline a standard algorithm for contracting tensor networks, which is responsible for the above feature of the tensor network. The basis of this algorithm is the following simple observation. Let S be any subset of tensors in a given tensor network T, and denote by T_S the tensor network derived by taking all the tensors in S and their incident edges. If there are incident edges whose other end is outside of S, we make them open edges. The observation is that if we replace the occurrence of T_S in T by the contraction of T_S, the contraction of T does not change.

Hence, we can think of the following incremental algorithm for contracting a tensor network. Suppose we are given a tensor network T whose bond dimensions and ranks are bounded by a constant, along with some ordering $A_1, \ldots, A_n$ on its constituent tensors. We begin by initializing $S^{(1)} = A_1$ and $T^{(1)} = T$. Then, at step i, we contract $T^{(i)}_{\{S^{(i)}, A_{i+1}\}}$ to obtain a tensor $S^{(i+1)}$, and we replace the occurrence of $T^{(i)}_{\{S^{(i)}, A_{i+1}\}}$ in $T^{(i)}$ by $S^{(i+1)}$ to obtain a modified tensor network $T^{(i+1)}$. At termination of the algorithm, $T^{(n)}$ will contain a single tensor equal to the contraction of T. Following the presentation in [13], the algorithm can be visualized as a bubble swallowing tensors one by one, while maintaining at each step the contraction of the part of the tensor network which has already been swallowed (Figure

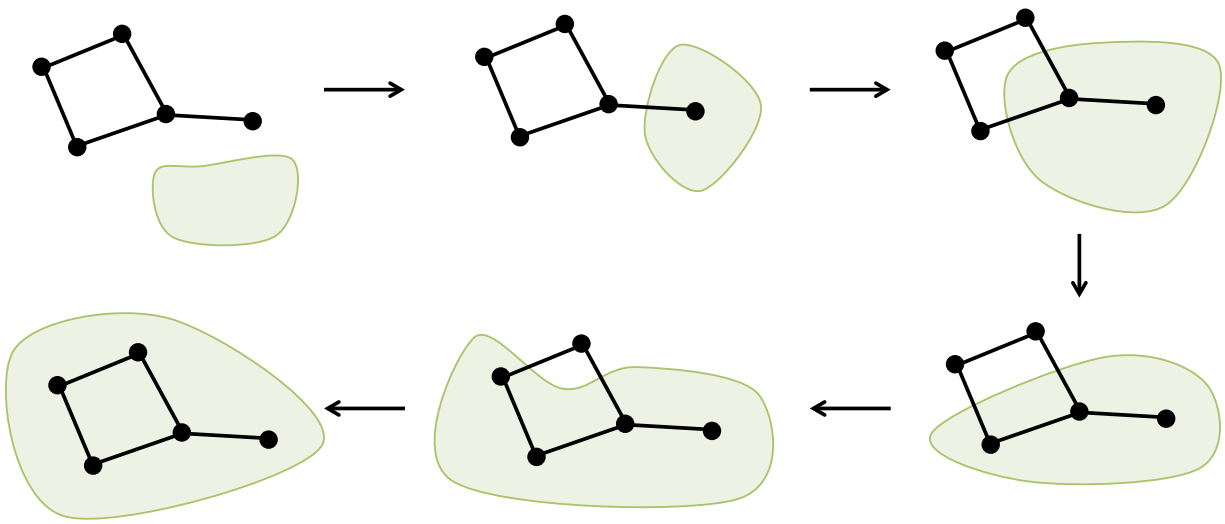

Figure 2.13: Swallowing of a tensor network.

2.13).

Clearly, the time and space complexity of the algorithm depends crucially upon the size of the computation performed at each step. Since the computation performed at each step i is simply a contraction of a tensor network with two vertices $S^{(i)}$ and A_{i+1}, the size of this computation is bounded by the number of edges involved in it. While by definition the degree of A_{i+1} is bounded by a constant, the degree of $S^{(i)}$ can be arbitrarily large depending on the tensor network and the given ordering. In fact, this observation can be formulated into the following theorem.

Theorem 2.2. Let $\{T_1, T_2, \ldots, T_n, \ldots\}$ be a family of tensor networks with bond dimensions bounded by $\mathrm{poly}(n)$ and ranks bounded by $O(\log n)$. If the tensor networks T_n in the given family admit an ordering $A_1, \ldots, A_n$ on its constituent tensors such that for every i, $\{A_1, \ldots, A_i\}$ defines a cut across which there are at most $O(\log n)$ edges, then the contraction of tensor networks in the given family can be computed in polynomial time.

In fact, matrix product states [115, 88], which are heavily used in quantum Hamiltonian complexity to describe quantum states of a 1D system, are one example of families of tensor networks that allow for such efficient contraction. Figure 2.14 illustrates a general concept of the matrix product state, which looks largely similar to the product state of Figure 2.6 except that it has horizontal edges of polynomial bond dimension. These edges allow for the description of a small amount of entanglement, a significant improvement over product states. Moreover, the 1D structure of the matrix product state allows for the efficient computation of many interesting physical properties. For instance, Figure 2.15 shows how the expectation of a two-qubit local observable O on a matrix product state $|\psi\rangle$ can be expressed as a tensor network. Tensor networks of this shape can be efficiently contracted by Theorem 2.2, using

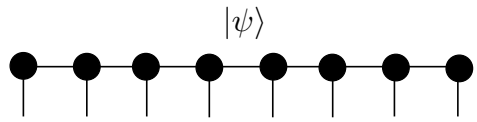

Figure 2.14: A matrix product state.

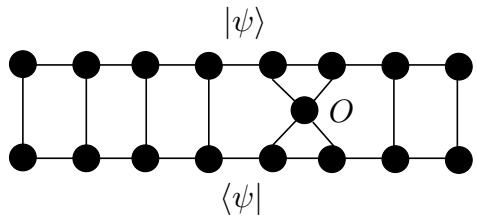

Figure 2.15: The tensor network corresponding to $\langle\psi|O|\psi\rangle$.

the ordering that sweeps the tensor network from left to right. Unfortunately, it is easy to see that these ideas do not generalize to the 2D setting, which is exactly why many important results in quantum Hamiltonian complexity (e.g. DMRG algorithm and area law) tend to hold only for 1D systems.

Chapter 3

A Turing Test for Quantum Annealers

The new form of the problem can be described in terms of a game which we call the "imitation game." It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A." The interrogator is allowed to put questions to A and B... We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?"

— Alan Turing [108]

This chapter is based on joint work with Graeme Smith, John A. Smolin, and Umesh Vazirani [101, 100].

3.1 Introduction

In 1950, Alan Turing authored a seminal paper on the topic of artificial intelligence [108], in which he introduced his famous test to distinguish machines from humans. Now simply named the "Turing test," the test consists in having an interrogator interact with two subjects A and B via a text-only channel, where one of A and B is the machine in test and the other is a human. At the end of the interrogation, the interrogator is to decide which one of the two subjects A and B is the machine. Turing concludes that if the interrogator is unable to distinguish the machine from the human, i.e. the failure rate of this game approaches $\frac{1}{2}$, we can conclude that the machine can indeed think.

The genius of the Turing test is that it conveniently avoids all philosophical and definitional problems about "thinking" by focusing strictly on the input-output behavior of

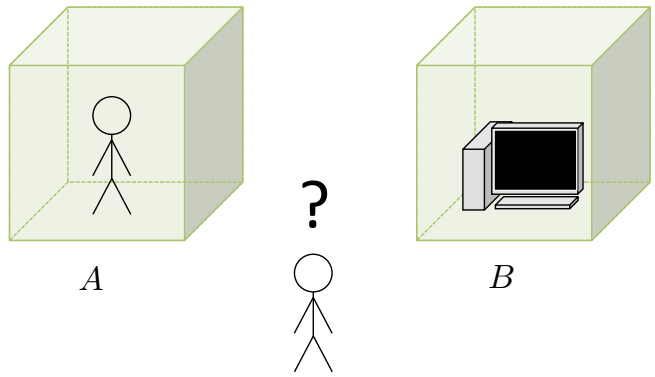

Figure 3.1: A Turing test.

the given machine. Ignoring any implementation details there may be, the test treats the machine simply as a black box and yet it provides a clear-cut criterion which is universally appealing. The Turing test teaches us that sometimes complete obliviousness to low-level details is the only reasonable way to approach a question.

Intriguingly, the problem of testing a quantum computer presents very similar issues to the ones presented by the problem of artificial intelligence. The main question to be addressed in the testing of a quantum computer is whether the given machine is indeed "quantum." Unfortunately it is difficult to arrive at a definition of "quantumness" which will be universally accepted, just as it is difficult to arrive at a universal definition of "thinking." For example, at the most basic level, there is even a sense in which every object in the universe is "quantum," because eventually its mechanics is governed by quantum physics.[1] Even if we try to refine our definition by requiring that the object has to make use of an exclusively quantum phenomenon which does not admit a classical description, we would still be in a situation in which we have to consider a laptop computer "quantum," because quantum mechanics is essential to the design and description of its central computing element, the transistor. On the other hand, a full-fledged universal quantum computer would be unambiguously "quantum," but such a criterion would leave out many interesting classes of quantum devices which are able to perform nonclassical computation while falling short of achieving universal quantum computation. Thus, the notion of "quantumness" seems to depend crucially upon the scale or level of abstraction and it is not always clear where the line should be drawn.

Even besides such definitional issues, the testing of a quantum computer poses more fundamental challenges. Although the description of the quantum state of a system scales exponentially in the size of the system, the laws of quantum mechanics severely limit the amount of information that can be accessed by measuring the system. For instance, Holevo's bound [71, 84] dictates that the amount of information that can be retrieved from an n-qubit

[1] In physics, classical mechanics is believed to be the macroscopic limit of quantum mechanics [46, 50].

system is bounded by n bits. This restriction, together with the mismatch in the computational power of the quantum computer and the classical tester, seems to constitute a substantial hindrance to finding a general strategy for the classical testing of quantum computers. More generally, as [38] suggests, these limitations may even be viewed as threatening the foundations of the scientific method itself. Namely, as our scientific theories such as quantum mechanics become computationally more complex and even intractable, it is getting increasingly difficult to compute the theoretical predictions to compare the experimental data to.

In this chapter, we introduce a kind of Turing test that can be used to test whether a given special-purpose quantum computer is indeed "quantum." In this test, the input-output behavior of a given machine is compared to that of a suitable classical model, where the machine and the classical model are both treated like black boxes as in the original Turing test. Such a test has two advantages. Firstly, by focusing solely on the input-output behavior of the given machine, we avoid any philosophical or definitional problem regarding what a "quantum" machine is, while providing an unambiguous criterion that can serve as a useful necessary condition for any such device to claim quantumness. Secondly, we manage to circumvent most of the fundamental challenges mentioned above, as the test consists only of physical experiments on the machine and simulations of a classical model.

For such a quantum Turing test to be effective, it is critically important to define and use a suitable classical model. In this thesis, we propose a simple classical model for quantum annealers, in which qubits are modeled by classical magnets coupled through nearest-neighbor interaction and subject to an external magnetic field. The finite temperature of the device is modeled by applying the Metropolis rule to randomly "kick" each magnet at each step. The detailed design is motivated as much by algorithmic considerations as physical ones, the goal being to preserve the algorithmic characteristics of the quantum annealer as far as possible while also introducing the extra assumption that quantum coherence in the machine is destroyed at each instant.

While we predict that such a suitable model can be defined on a broader range of special-purpose quantum computers, in this thesis we focus on devices that are based on quantum annealing, a heuristic algorithm designed to solve **NP**-hard optimization problems. In the absence of a well-developed theory of quantum fault-tolerance or a rigorous analysis of quantum speedup by such machines, the testing of quantum annealers is a critically important problem in experimental quantum computing today. We note that quantum annealing is as yet the only model of special-purpose quantum computation which is being implemented at a scale large enough to demand serious testing, intensely pursued by companies such as D-Wave Systems and Google. The newest model of the D-Wave quantum annealer, D-Wave 2X, has as many as 1097 working qubits.

3.1.1 Related work

We note that there is a sequence of remarkable results [1, 34, 92, 18, 52, 19] that show that the framework of interactive proofs from computational complexity theory together with

the remarkable properties of quantum entanglement provides a way, in principle, of testing general-purpose quantum computers. These results build upon the well studied theory of interactive proofs from complexity theory, in which the computationally bounded verifier seeks to verify a statement made by an all-powerful, but potentially malignant prover. To adapt this framework to our problem of testing quantum computers, we could replace the all-powerful prover in the interactive proof by a universal quantum computer and obtain a complexity class called Quantum Prover Interactive Proofs (**QPIP**), originally introduced in [1]. If it happens that **QPIP** = **BQP**, where **BQP** denotes the class of decision problems that can be solved by a quantum computer in polynomial time with bounded error, it would imply that the classical verifier can indeed verify any quantum computation via interactive proofs, as desired. While this question is still open, certain restricted versions of the theorem have been proved. For instance, it is known that the classical verifier who is capable of small-scale quantum computation involving only a constant number of qubits at a time can verify arbitrary quantum computation [1, 34], and that the classical verifier with access to two quantum provers which are entangled but are not allowed to communicate with each other can verify arbitrary quantum computation [92].

On the other hand, we observe that the above techniques tend to demand that the quantum computer that is being tested implements a rather strong model of quantum computation. For instance, even if the actual computation to be verified is very simple, the testing protocol might require the quantum computer to perform arbitrarily complex **BQP** computation. Therefore, these schemes do not address the testing of special-purpose quantum computation. This is a serious shortcoming, especially since general-purpose quantum computation presents an enormous engineering challenge which is not expected to be overcome in the next decade.

3.1.2 Preliminaries

In this section we introduce some elementary concepts and notations that will be used throughout the rest of this chapter.

3.1.2.1 Pauli matrices

In quantum mechanics, there are three very frequently used one-qubit observables known as Pauli matrices, which are defined as follows:

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

While these matrices have many interesting properties that would normally deserve a dedicated section, here we only cover a few properties that will be needed for our later exposition. When we refer to these matrices as local Hamiltonian terms in an n-qubit system, we will denote which qubit they are acting on by the means of subscripts, e.g. σ_1^x, σ_5^z, etc. Moreover,

for the sake of simplicity we will often allow the following abuse of notation:

$$\begin{aligned}
\sigma_i^x &:= I_1 \otimes \cdots \otimes I_{i-1} \otimes \sigma_i^x \otimes I_{i+1} \otimes \cdots \otimes I_n \\
\sigma_i^z &:= I_1 \otimes \cdots \otimes I_{i-1} \otimes \sigma_i^z \otimes I_{i+1} \otimes \cdots \otimes I_n \\
\sigma_i^z \sigma_j^z &:= I_1 \otimes \cdots \otimes I_{i-1} \otimes \sigma_i^z \otimes I_{i+1} \otimes \cdots \otimes I_{j-1} \otimes \sigma_j^z \otimes I_{j+1} \otimes \cdots \otimes I_n
\end{aligned}$$

Without further ado, we make the following observations:

1. The ground state of $-\sigma^x$ is the uniform superposition.

   This is easily verified by noting that the two eigenvectors of σ^x are $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$, with eigenvalues 1 and -1 respectively. Hence, the ground state of $-\sigma^x$ is $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, which is indeed the uniform superposition. More generally, the ground state of the n-qubit Hamiltonian $\sum_{i=1}^n \sigma_i^x$ is the n-qubit uniform superposition $\sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}|x\rangle$.

2. A $\pm\sigma^z$ term in the Hamiltonian biases the qubit towards $|1\rangle$ or $|0\rangle$.

   This is again easily verified by noting that the two eigenvectors of σ^z are $|0\rangle$ and $|1\rangle$, with eigenvalues 1 and -1 respectively. Hence, the ground state of σ^z is $|1\rangle$, whereas the ground state of $-\sigma^z$ is $|0\rangle$.

3. A $\pm\sigma^z \otimes \sigma^z$ term in the Hamiltonian "couples" two qubits.

   Since

   $$-\sigma^z \otimes \sigma^z = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix},$$

   this Hamiltonian term would prefer to "align" the two qubits by putting them in either the state $|00\rangle$ or $|11\rangle$. Similarly, a $\sigma^z \otimes \sigma^z$ term in the Hamiltonian would prefer to anti-align the two qubits by putting them in either $|01\rangle$ or $|10\rangle$.

3.1.2.2 Bloch sphere

Another useful formalism in quantum mechanics is the Bloch sphere, which provides a convenient representation of a qubit in terms of a spin pointing in some direction in three-dimensional space. Namely, since we can write any given one-qubit quantum state $|\psi\rangle$ as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

where $0 \leq \theta \leq \pi$ and $0 \leq \phi < 2\pi$,[2] we can easily map it onto the surface of a three-dimensional sphere and vice versa, as in Figure 3.2. Moreover, it is straightforward to check

[2] We can assume without loss of generality that the coefficient of $|0\rangle$ is real because the global phase of a quantum state is physically irrelevant.

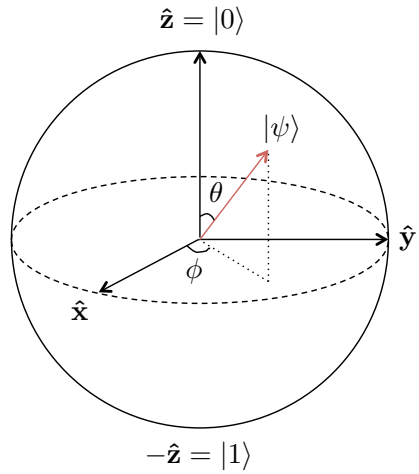

Figure 3.2: The Bloch sphere representation of one-qubit state $|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$.

that the three axes of this three-dimensional sphere correspond to the eigenvectors of the three Pauli matrices. Namely, the $+1$ eigenvectors of σ^x, σ^y, and σ^z correspond to $\hat{x}$, $\hat{y}$, and $\hat{z}$ on the Bloch sphere, and similarly the -1 eigenvectors of σ^x, σ^y, and σ^z correspond to $-\hat{x}$, $-\hat{y}$, and $-\hat{z}$.

The Bloch sphere provides a particularly useful geometrical interpretation of the qubit. For instance, since the standard basis states $|0\rangle$ and $|1\rangle$ are mapped to $\hat{z}$ and $-\hat{z}$ on the Bloch sphere respectively, we can think of the standard basis measurement as measuring whether the spin is pointing up or down on the Bloch sphere. The sign and magnitude of the z-component of the spin determines how probable it is to see each outcome, with the two outcomes being equiprobable when the spin is pointing at some point on the equator, e.g. $\hat{x}$.

However, for obvious reasons the Bloch sphere is not going to be as useful in describing the state of a qubit which is entangled with other qubits, although a certain limited description is still possible via yet another formalism called mixed states [84].

3.1.3 Introduction to quantum annealing

3.1.3.1 Top-down approach to quantum computing

Whereas building a large-scale general-purpose quantum computer is a monumental task, in recent years there is more optimism about the possibility of building large-scale special-purpose quantum computers. Here, it is important to make note of the fact that a special-purpose quantum computer is not merely a simpler version of a general-purpose quantum computer. On the contrary, models of special-purpose quantum computation often tend to employ principles that are fundamentally different from those employed in a model of

general-purpose quantum computation. To highlight these differences, we observe that the standard approach to building quantum computers can be thought of as the "bottom-up" approach in that it first requires building well-specified elementary components such as high-fidelity qubits and high-precision gate operations, and only then uses these lower-level components as building blocks to implement higher-level components such as quantum error correction, quantum algorithms, etc. The relatively slow progress in experimental quantum computing can be attributed to the technical difficulty of realizing each of these components at sufficiently high quality.

In contrast, engineering of a special-purpose quantum computer frequently follows what may be called the "top-down" approach to computing. In the quantum setting, the spirit of this approach would be translated as to first build *some* large-scale quantum system and then try to find interesting computational problems that could be mapped to that system. For better illustration, let us imagine for the moment that we are interested in the problem of determining the shapes of water waves in a water basin. One approach to solving this problem would be to first engineer a general-purpose classical computer which is Turing-complete and then program it to simulate the physics of the water basin. However, there is a sense in which this is a huge overkill with respect to the problem at hand. For instance, if determining the shapes of the water waves was the only problem that we were interested in solving, we can easily achieve our goal by having a real water basin along with a measuring instrument, as, for this particular problem, the water basin itself can be considered a highly effective special-purpose computer.

This kind of top-down approach is particularly attractive in quantum computing because it lets us bypass many difficult problems in quantum engineering, quantum error correction, fault-tolerance, etc., while still retaining the hope of achieving nontrivial quantum computation. In particular, it is easy to see that if we manage to build *any* large-scale quantum system that achieves nontrivial quantum coherence, it would immediately yield a special-purpose quantum computer that outperforms all classical computers on at least one problem, namely the simulation of that system itself. While such a problem may not always be of much practical relevance, we can imagine that it will at least be of enormous interest to physicists.

By far, the leading candidate for such a "top-down" approach is quantum annealing (see e.g. [75]), which is particularly promising in that it seeks to implement a quantum system that can encode a large class of practically relevant problems. On the highest level, quantum annealing can be thought of as a heuristic implementation of a quantum physical process called adiabatic dragging, which is known to be able to encode the problem of finding the ground state of a given classical Hamiltonian [49]. This class of problems is **NP**-hard [16] and therefore any technique for solving them quickly and reliably is of great interest. While there is no theoretical guarantee that even a successful, rigorous implementation of adiabatic dragging will actually yield the optimal solution to these problems, the hope is that quantum annealing may be able to utilize quantum effects such as tunneling [74] to yield a performance that is superior to classical computers. The basis of this optimism will be elaborated upon in Section 3.1.3.3.

In addition, we note that there has been considerable optimism about the possibility of building large-scale practical quantum annealers in recent years. The announcement by the Canadian company D-Wave Systems of its 108-qubit quantum annealer in 2011 led to intense excitement in the mainstream media (including a Time magazine cover dubbing it "the infinity machine") and the computer industry, and a lively debate in the academic community. This, and subsequent scaling and improvements [25, 24, 93, 23, 43], have highlighted both the importance and challenge in testing special-purpose quantum computers.

3.1.3.2 Adiabatic quantum computing

Before we introduce quantum annealing, it will be appropriate to first have a brief discussion of a model called adiabatic quantum computing, which served as a primary inspiration in the development of quantum annealing. An alternative approach to quantum computing, adiabatic quantum computing was first proposed by Farhi, Goldstone, Gutmann, and Sipser [49] as a way to achieve quantum speedup for combinatorial search problems such as satisfiability. Its main idea, which is very simple and elegant, can be summarized as follows. Suppose we have an n-bit combinatorial search problem that we would like to solve, e.g., satisfiability. To implement adiabatic quantum computing, we begin by first preparing an n-qubit quantum system in the ground state of some simple "initial" local Hamiltonian H_0, such as $H_0 = -\sum_{i=1}^{n} \sigma_i^x$. We note that the ground state of this choice of H_0 is simply the uniform superposition of all standard basis states and is relatively easy to prepare in the lab. Then, we "adiabatically drag" the system Hamiltonian from H_0 to another local Hamiltonian H_f, which encodes the computational problem of interest and is typically significantly more complex. This kind of adiabatic dragging can be implemented as a time-dependent local Hamiltonian of the following form:

$$H(t) = (1-t)H_0 + tH_f.$$

Here, we assume that the system temperature is maintained at zero for the whole duration of the process (hence the name adiabatic) and that t goes from 0 to 1. That is, if the actual duration of the dragging process is t_f, the system Hamiltonian at time $s \in [0, t_f]$ will be given by $H(s/t_f)$. Intuitively speaking, the idea of adiabatic quantum computing is that if we prepare our system in the ground state of the initial Hamiltonian and then morph the system Hamiltonian into another "target" Hamiltonian sufficiently slowly, the system will always remain in the ground state of the resulting time-varying Hamiltonian at each instant of the dragging process. In particular, at the end of the process the system will be found in the ground state of the final Hamiltonian H_f, which can be designed to encode the solution to a desired combinatorial optimization problem. For example, we saw in Section 2.3 that local Hamiltonians can encode solutions to arbitrary instances of constraint satisfaction problems, including satisfiability.

Moreover, the well-known adiabatic theorem [73] in quantum mechanics rigorously establishes the above intuition:

Theorem 3.1. Let $\Delta = \min_{0 \leq t \leq 1}\{\lambda_2(H(t)) - \lambda_1(H(t))\}$, where $\lambda_i(A)$ denotes the i-th smallest eigenvalue of A. If $t_f = \Omega(1/\Delta^3)$, the probability that the system will be found in the ground state of H_f at time $t = 1$ is bounded away from 0.

This means that the running time of the adiabatic algorithm as an exact algorithm should be thought of as proportional to $1/\Delta^3$. This inverse dependence on Δ has an intuitive explanation; since a small value of Δ implies the existence of suboptimal states which come very close to the optimal state at some point during this evolution, in such a case it would be logical to exercise more care by increasing our running time, so as to ensure that the system does not "jump" from the optimal state to a suboptimal state.

Unfortunately, there are known instances of certain combinatorial optimization problems, such as satisfiability, for which the spectral gap Δ is shown to scale inverse-exponentially [41, 42, 91]. This implies that the adiabatic algorithm also requires exponential time to solve those problems in the worst case. While other instances have been constructed on which the adiabatic algorithm provably outperforms simulated annealing [48, 91], there is some numerical evidence that Δ is exponentially small even in the generic case, i.e. for randomly chosen instances [11]. Perhaps these results are not too surprising, given that adiabatic quantum computing has been shown to be equivalent to the standard gate model of quantum computing, up to a polynomial overhead [5].

On the other hand, the adiabatic model of quantum computing may offer significant advantages over the standard gate model with respect to experimental realization. This is already evident in the fact that the adiabatic model is described in the language of physics, i.e. using Hamiltonians and continuous-time evolution, whereas the standard gate model is described in the language of computer science, i.e. using gates and circuits. In addition, the adiabatic model is known to possess some intrinsic robustness to certain kinds of noise [49, 36]. Nevertheless, the fact that the adiabatic model has no well-established theory of error correction (see [121]) will be a huge concern for anyone who wishes to implement it, as several studies [36, 55, 106, 12, 112] show that such intrinsic robustness is not sufficient to protect computation.

3.1.3.3 Quantum annealing

Quantum annealing, which is the main subject of this chapter, can be understood as a noisy implementation of adiabatic quantum computing. It is defined in terms of the same kind of Hamiltonian dragging as adiabatic quantum computing, but differs in that the system temperature is assumed to be finite (i.e. nonzero) and the spectral gap condition in the adiabatic theorem is not respected. Since its premises are not met, the adiabatic theorem is no longer able to provide any form of performance guarantee to the model.

Hence, quantum annealing is not to be considered next to exact algorithms typically studied by computer science theorists, which come with a rigorous guarantee to find the optimal or near-optimal solution within a prescribed running time. Rather, quantum annealing begins with the admission that there is no such theoretical guarantee, and focuses more on

obtaining a good field performance. In that regard, quantum annealing is a *heuristic* and therefore would more appropriately be compared with classical heuristic algorithms such as simulated annealing.

Interestingly, there is a sense in which comparison between quantum annealing and simulated annealing (unfamiliar readers are referred to the explanation in Section 3.3.1) is particularly more appropriate, for these two methods seem to be built upon roughly the same principle. Namely, both algorithms begin exploring the search space from a completely random state at time $t = 0$ and gradually increase the influence of the target Hamiltonian over time, in the hope that such gradual evolution will help the system escape local minima and have greater chances of finding the global minimum. The crucial difference is that in quantum annealing a random state means a quantum superposition over multiple states, whereas in simulated annealing it means a classical ensemble over states. Hence, in this sense quantum annealing can be viewed as a quantum analogue of simulated annealing.

The basis for optimism about quantum annealing's potential for a super-classical speedup lies in the possibility of a phenomenon called quantum tunneling, whereby quantum systems are able to get through barriers in the energy landscape notwithstanding an energy deficit (see [91] for a concrete example). This is in stark contrast to the classical case, where the system must acquire the amount of energy prescribed by the barrier if it were to ever surmount the barrier (see Figure 3.3). Hence, if one could ensure that a given implementation of quantum annealer is capable of inducing quantum tunneling at a sufficiently large scale, it would at least be a demonstration of a computational primitive which is inherently quantum and cannot be reproduced by any classical device. Even though the aforementioned results about adiabatic quantum computing may be suggesting that there will be no implication on the asymptotic time complexity of **NP**-hard optimization problems, it is reasonable to expect that such a novel primitive will eventually find a practical use case.

The possibility of a speedup by quantum tunneling presents two main questions: firstly, can quantum tunneling have a significant impact on the computation in zero-error models such as adiabatic quantum computing? Unfortunately, as we pointed out earlier, there are known instances of certain combinatorial optimization problems, such as satisfiability, for which the spectral gap Δ is shown to scale inverse-exponentially [41, 42, 91]. Hence, the adiabatic algorithm also requires exponential time to solve those problems in the worst case. While other instances have been constructed on which the adiabatic algorithm outperforms simulated annealing by exercising quantum tunneling [48, 91], there is some numerical evidence that Δ is exponentially small even in the generic case, i.e. for randomly chosen instances [11].

Secondly, there is also the question of whether quantum tunneling can occur in quantum annealers despite the fact it is implemented at a finite (i.e. nonzero) temperature. This means that the system is subject to thermal noise at all times and, in particular, can lose quantum coherence very rapidly. In the absence of a general scheme for error correction on quantum annealers, whether the quantum annealer can support large-scale quantum phenomena in spite of such rapid decoherence is a particularly big concern. While some studies [25, 45, 8] suggest that decoherence in a quantum annealer may take place in the eigenbasis of the

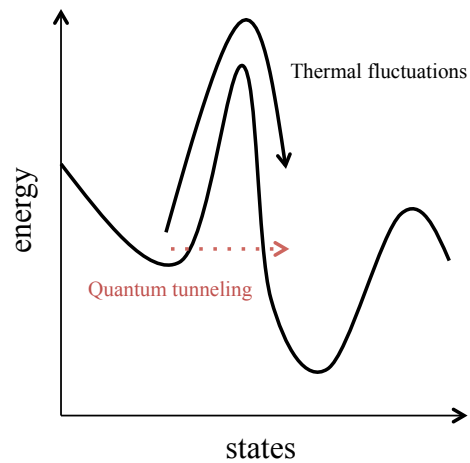

Figure 3.3: A schematic illustration of quantum tunneling vs. classical thermal fluctuations.

Hamiltonian and therefore be less detrimental, as yet there is no clear consensus on whether large-scale quantum effects can indeed survive.

On the other hand, we observe that the finite temperature of the quantum annealer is not necessarily a strictly adverse factor. On the contrary, there is even a sense in which it can be highly beneficial to the computation, because thermal fluctuations that happen at a finite temperature are in fact the very basis of classical annealing-based algorithms such as simulated annealing. In such algorithms, it is precisely these thermal fluctuations that provide the system with the necessary energy to surmount energy barriers surrounding local minima and give it the potential to settle into the global minimum. In fact, simulated annealing at zero temperature would simply correspond to the local search algorithm that only explores neighboring states which are better than the current one, and it is easy to see that such an algorithm would immediately get stuck in the first local minimum it finds, yielding a terrible performance.

Therefore, while being a spiritual successor of adiabatic quantum computing, quantum annealing should at the same time be clearly distinguished from adiabatic quantum computing. While both algorithms depend upon the possibility of quantum tunneling to achieve a super-classical speedup, quantum annealing is also intended to make active use of the above computational primitive based on classical thermal effects, and it must be made clear that quantum tunneling can effectively assist this classical primitive in a manner that will manifest a noticeable improvement in the performance of the algorithm.

All in all, two main questions that must be answered about any given implementation of a quantum annealer are as follows:

1. *Does the implementation support large-scale quantum tunneling?*

   An important condition for a quantum annealer to achieve super-classical performance is large-scale quantum tunneling. If the device is incapable of inducing quantum tun-

neling or it can only support quantum tunneling at a small scale (e.g. affecting only a constant number of spins at a time), it would likely imply that the computational primitives used by the machine are local and can be efficiently simulated classically.

2. *Does quantum tunneling benefit computation (i.e. yield a speedup over classical algorithms)?*

   Even if the quantum annealer does support large-scale quantum tunneling, it does not directly imply that it achieves a computational speedup. In order to claim a speedup, a concrete class of instances on which the machine outperforms the best classical algorithms must be demonstrated.

In this chapter, we seek to introduce a systematic approach to the above two questions, via the concept of a quantum Turing test.

3.2 Quantum Turing test

The heuristic nature of quantum annealing presents a unique challenge in the verification of its experimental realizations. We note that in the engineering of a general-purpose quantum computer, in which each individual component such as qubits, quantum gates, measuring equipment, etc., are carefully tested and specified, there is a sense of correctness provided by the accuracy of those specifications. Namely, once all the elementary components used to build a quantum computer are fully specified – e.g. in terms of the time it takes for a qubit to decohere, the fidelity of a gate operation, the failure rate of an error correction code, etc. – there is a sense in which we can expect the composition of those elementary components to also behave according to our predictions, provided our understanding of quantum mechanics is correct. On the other hand, such bottom-up approaches to verification are obviously not compatible with top-down computational models such as quantum annealing. What would be an appropriate way of testing or benchmarking special-purpose quantum computers such as quantum annealers, both in terms of their ability to achieve nontrivial quantum coherence and in terms of their ability to achieve a quantum speedup?

In this thesis, we propose that a kind of quantum Turing test, inspired by the famous Turing test in artificial intelligence, provides a particularly effective way of exploring these questions. Turing introduced his famous test to address the question "Can machines think?" By analogy a quantum Turing test seeks to address the question "Is the given machine quantum?" The main idea of the quantum Turing test is to compare the input-output behavior of the given machine to that of a suitable classical model on inputs sampled from a prescribed probability distribution, while treating both the machine and the classical model as black boxes in the same fashion as in the original Turing test. If the machine turns out to behave nearly identically to the classical model, we can conclude that the machine is not sufficiently quantum and that it fails the quantum Turing test. In that sense, the classical model in the quantum Turing test can be viewed as serving a role similar to that of the

null hypothesis in statistical testing. Such a test is particularly relevant in the context of a special-purpose quantum computer such as a quantum annealer, since such models do not have a general theory of fault-tolerant quantum computation that can provide a rigorous guarantee of quantum coherence.

Noting the fact that a special-purpose quantum computer is designed to solve (speed up) a restricted class of computational problems, the quantum Turing test can be defined as follows. Formally, a special-purpose quantum computer is a machine M together with a probability distribution D on inputs. Here, the probability distribution D can be thought of as defining the class of problems intended for the given special-purposed quantum computer M. Hence, if we interpret D as a distribution over all inputs to a universal quantum computer, the restriction is that M is designed to solve only those instances that are in the support of D. The resources of M are specified by a bound $P(n)$ on its degree of parallelism (or the amount of hardware in the machine) and a bound $T(n)$ on its running time, where n denotes the size of the input. A quantum Turing test treats M as a black box, and tests the claim that M has significantly better performance on random inputs drawn from the distribution D than any classical algorithm. To falsify the claim, we wish to find a classical model C that uses comparable resources ($O(P(n))$ parallelism and $O(T(n))$ time) and whose input-output behavior is nearly indistinguishable from that of M on randomly chosen inputs from the distribution D. If there exists such a classical model C, we conclude that the given machine M fails the quantum Turing test.

3.3 Classical models for quantum annealers

In applying the quantum Turing test to a given quantum device, the most important issue is that of the choice of the classical model. To see this, we point out that there is an inherent asymmetry in the quantum Turing test. Namely, while the existence of even a single classical model that reproduces the input-output behavior of the given machine immediately constitutes its failure on the quantum Turing test, exhaustion of *all* classical models is necessary to ensure with certainty that the machine passes the test. In practice, where such exhaustion is infeasible, the quantum Turing test should at least entail distinguishing the machine from some class of "reasonable" classical models. If the classical models used in the test are physically and algorithmically well-motivated, even such limited versions of the test can provide some evidence that the machine exhibits quantum effects or performs quantum computation.

In this section we propose a simple classical model for quantum annealers, which provides such a reasonable starting point for the quantum Turing test. In this model, qubits are modeled by classical magnets coupled through nearest-neighbor interaction and subject to an external magnetic field. The finite temperature of the device is modeled by applying the same kind of Metropolis rule as in simulated annealing to randomly "kick" each magnet at each step. The detailed design of the model is motivated as much by algorithmic considerations as physical ones, the goal being to preserve the algorithmic characteristics of

the quantum annealer as far as possible while also introducing the extra assumption that quantum coherence in the machine is destroyed at each instant. Viewed in the Bloch sphere formalism, our model can be understood as what physicists call a "mean-field approximation" of quantum annealing, which means that the system is assumed to remain in a product (i.e. unentangled) state at all times. In this sense, our classical model is a natural classical analogue of quantum annealing and provides a suitable foil against which to test that the given implementation of a quantum annealer achieves nontrivial quantum coherence. This addresses the first question raised in Section 3.1.3.3.

The classical model can also play a role in testing that the quantum annealer achieves a quantum speedup, i.e. testing a claim that there is a computational problem that it solves much faster than any classical algorithm. The key challenge in testing such a claim is, for a given choice of computational problem, how to identify a good (if not the best) classical algorithm to compare against. We argue that our classical model provides a natural starting point in the search for such a classical algorithm. Indeed, since our model is designed to explain the computational power of quantum annealing in the absence of quantum tunneling, there is even a sense in which the performance gap between our model and the given machine may be interpreted as corresponding to the contribution of quantum tunneling. Thus, our classical model addresses the second question from Section 3.1.3.3.

It is interesting to contrast our classical model with simulated annealing as a possible benchmark for a quantum annealer. Indeed, given quantum annealing's resemblance to simulated annealing, comparison tests based on simulated annealing have frequently been employed to argue that e.g. the D-Wave machine performs quantum computation [25, 24, 43]. An analysis of our findings sheds new light on the nature of quantum annealing and its comparison with simulated annealing. Our classical model can be viewed as a variant of simulated annealing which works with 2D vectors rather than bits (or classical spins) in the presence of an external bias in the x-direction. While neither feature by itself (2D vectors and external x-bias) changes the qualitative behavior vis-à-vis simulated annealing, surprisingly the combination of both features makes the model behave very differently from simulated annealing, producing a behavior that could naïvely be interpreted as a signature of quantum tunneling. The fact that a purely classical model like ours can exhibit such a behavior exposes a pitfall in using "quantum signatures" as evidence of quantum tunneling.

Before we introduce our classical model, we begin by introducing several related classical models that have been used in the literature to study quantum annealing.

3.3.1 Simulated annealing

In this section, we provide a brief introduction to simulated annealing, one of the most popular classical metaheuristics for optimization problems.

Practiced since prehistoric times, annealing is a highly effective metallurgical technique for obtaining high-quality metallic crystals. The method consists simply in first heating the target material to a high temperature and then letting it slowly cool back to room temperature. Annealed materials tend to possess various favorable characteristics such as

increased ductility, malleability, softness, etc. Interestingly, there is a surprising analogy between this metallurgical task of obtaining high-quality crystals and the computational optimization problems which have been the subject of our discussion. In fact, the task of obtaining high-quality crystals can directly be viewed as an optimization problem, because higher-quality crystals have particles that are more regularly spaced and therefore have a smaller physical energy. In this sense, annealing can be viewed as a technique for achieving lower-energy configurations of a given material.

Simulated annealing [77] is a metaheuristic obtained by directly simulating the metallurgical annealing process in the context of computational optimization problems. As an algorithm, it can be outlined as follows:

- Initialize the temperature T to be some high constant T_0. Generate a random solution x.
- Repeat the following for a suitable number of steps:
  - Sample a random neighbor x' of x by making a random local modification to x. For instance, if x is binary, x' can be generated by flipping a randomly selected bit of x.
  - Let $\Delta f = f(x') - f(x)$, where f is the function we seek to minimize. With probability $\min\{1, e^{-\Delta f/T}\}$, accept the new solution x' by updating $x \leftarrow x'$.
  - Decrease T by a small amount.

Note that the temperature T in this algorithm can be interpreted roughly as our aggressiveness in the exploration of the search space. For instance, when T is close to 0, moves that increase the cost function f even very slightly will have a vanishing acceptance probability $e^{-\Delta f/T}$ and will be virtually prohibited, yielding a rather conservative exploration of the search space. In contrast, when T is very large, the acceptance rule practically ignores the cost difference Δf and always accepts the new solution x', yielding a very aggressive exploration of the search space.

Therefore, simulated annealing can intuitively be thought of as an algorithm that initially attacks the search space completely randomly, but gradually increases the influence of the cost function f over time. The idea is that the higher temperature in earlier parts of the annealing process enables the algorithm to explore the search space without being stuck in local minima, enhancing its chances to find a region which is likely to contain the global minimum. Then, as the temperature approaches zero, the algorithm will eventually behave like a simple hill descending algorithm and quickly settle into a nearest local minimum. Note that this provides an interesting parallel to quantum annealing, in which also the randomness (which in the quantum case corresponds to the bias towards the uniform superposition) is decreased over time and the strength of the problem Hamiltonian is increased over time.

Clearly, the initial temperature T_0 and the amount by which T is decreased in each step are important parameters in simulated annealing. Together, these parameters constitute what is called the annealing schedule of the algorithm. We note that the latter parameter does not have to be constant. In fact, exponential update rules such as $T \leftarrow 0.99T$ are frequently used to implement the annealing schedule in practice. More generally, we can

think of the temperature as an arbitrary positive decreasing function $T(t)$ where t is the step number.

While the algorithm is relatively straightforward, the choice of the acceptance probability $\min\{1, e^{-\Delta f/T}\}$ merits more discussion. Widely known as the Metropolis acceptance rule, it was first proposed by Metropolis et al. in 1953 [82] in order to make the random walk implicit in the above algorithm converge to a physically meaningful distribution. Indeed, under the above choice of the acceptance rule, the stationary distribution of this random walk is shown to be the well-known Gibbs distribution $F(x) \propto e^{-f(x)/T}$,[3] which is widely used in statistical physics to describe the state of a given physical system at a fixed temperature T. In this interpretation, each x is thought of as corresponding to a physical configuration and $f(x)$ its energy. Hence, at fixed T, the above algorithm is merely a sampling algorithm for the Gibbs distribution, and this provides a justification for the claim that simulated annealing is a physical simulation of annealing.

While simulated annealing is guaranteed to converge to the global minimum after a sufficiently large number of steps [62], it is recognized mainly as a heuristic because the number of steps required for exact convergence is usually exponential. Nevertheless, simulated annealing is celebrated as one of the most successful metaheuristics for practical optimization problems.

3.3.2 Simulated quantum annealing

Another classical model for quantum annealing that is frequently considered in the literature is simulated quantum annealing, which is nothing but a direct simulation of quantum annealing itself. While a simulation of quantum systems is generally infeasible due to their exponential nature, a specific class of quantum Hamiltonians called stoquastic Hamiltonians (some details are provided in Section 4.2.5) are commonly believed to be "easy" to simulate using certain nontrivial methods [29]. The methods that are used to simulate such systems are known under the name of quantum Monte Carlo (e.g. [17]), and are based on a mapping that is known to exist between stoquastic quantum systems in d spatial dimensions and classical systems in $d+1$ spatial dimensions. It turns out that the quantum Hamiltonian for adiabatic quantum computing and quantum annealing described in Section 3.1.3.2 also satisfies the condition of stoquasticity, and therefore may admit accurate simulations via quantum Monte Carlo methods. Of course, if such simulations were actually accurate, then any potential for a quantum speedup would be immediately ruled out. However, the belief that quantum Monte Carlo simulations are effective for stoquastic Hamiltonians is of heuristic nature, and it is possible that there are features of quantum annealing which are not described by quantum Monte Carlo simulations.

[3] While the actual Gibbs distribution is defined as $F(x) \propto e^{-f(x)/kT}$, here we will think of the Boltzmann constant k as being subsumed into T.

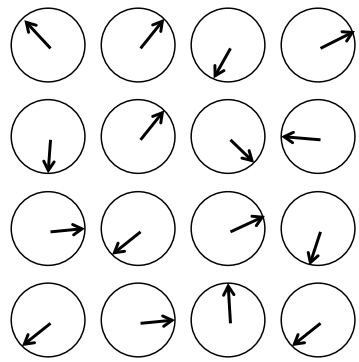

Figure 3.4: Classical 2D spins. Since the state of each individual qubit is fully described by the corresponding 2D vector, no entanglement is possible between these qubits.

3.3.3 Classical spin dynamics

In this section, we briefly introduce a classical model for quantum annealing proposed by Smolin and Smith [104], originally in the context of the D-Wave machine. Noting that qubits in the D-Wave machine are reported to decohere in time that is orders of magnitude shorter than the total annealing time of the machine, they propose that these qubits should be modeled as 2D classical spins, i.e. each qubit i should be represented by a 2D unit vector on the x-z plane making some angle θ_i with the $\hat{z}$ axis. Hence, any σ_i^x term in the original quantum Hamiltonian is now replaced by the x-component of the classical spin, which is given by $\sin\theta_i$, and any σ_i^z term is replaced by the z-component of the classical spin, given by $\cos\theta_i$. If $H(t)$ is the classical time-dependent Hamiltonian[4] thus obtained, the equations of motion of the system are simply defined by the following ordinary differential equations:

$$\frac{d}{dt}\theta_i = \dot{\theta}_i,$$
$$\frac{d}{dt}\dot{\theta}_i = \frac{dH}{d\theta_i}.$$

Then, it is straightforward to simulate the model by integrating this system of ordinary differential equations.

3.3.4 Our model

Finally, we describe our classical model for benchmarking quantum annealers. While our model shares certain features with some of the previous models, we argue that it provides a more appropriate benchmark for quantum annealers than any of those models.

[4]Note that a classical Hamiltonian is simply a function from the set of possible classical states (in this case $[0, 2\pi)^n$) to $\mathbb{R}$ which returns the energy of a given state.

In a typical implementation of quantum annealing for 2-local Hamiltonians, a quantum annealer consists of a set of qubits subject to a time-varying Hamiltonian of the form:

$$H(t) = -A(t)\sum_i \sigma_i^x - B(t)\left(\sum_i h_i\sigma_i^z + \sum_{1\le i<j\le n} J_{ij}\sigma_i^z\sigma_j^z\right).$$

Here $A(t)$ controls the magnitude of the transverse x-field (i.e. the initial Hamltonian), and $B(t)$ that of the local z-fields and z-z interactions (i.e. the final Hamiltonian). The z-z interactions are limited to neighboring pairs of qubits $\{i, j\}$ on an interaction graph which is dictated by the hardware architecture. For example, if the implemented interaction graph of a given quantum annealer were a 2D grid, J_{ij} would be allowed to have a nonzero value only if i and j are neighboring qubits on that grid. As before, we can assume that t goes from 0 to 1, and that the boundary conditions are $B(0) \approx 0$ and $A(1) \approx 0$, so that the ground state of the final Hamiltonian $H(1)$ is determined solely by the local z-fields and z-z interactions. The quantum annealer works by first preparing the system in the ground state of $H(0) \approx -A(0)\sum_i \sigma_i^x$ and then gradually evolving the Hamiltonian from $H(0)$ to $H(1)$ according to the schedule specified by $A(t)$ and $B(t)$. The computational problem that is native to the quantum annealer is therefore that of finding the ground state of a classical Ising spin glass, i.e. a spin configuration with each spin value $z_i \in \{-1, 1\}$ chosen to minimize the energy $H = -\sum_i h_i z_i - \sum_{1\le i<j\le n} J_{ij} z_i z_j$. We note that this problem is known to be **NP**-hard on 2D grids [16], but admits a polynomial-time approximation scheme on planar graphs [15].

To apply a quantum Turing test to quantum annealers, we introduce a classical 2D spin model that employs a Metropolis-like noise process. In this sense, the model can be viewed as incorporating ideas from both simulated annealing and classical spin dynamics. In our model, each spin i in the quantum annealer is modeled by a classical magnet pointing in some direction θ_i in the x-z plane. We further assume that

1. there is an external magnetic field of intensity $A(t)$ pointing in the $\hat{x}$-direction,
2. there is a local magnetic field of intensity $B(t)h_i$ pointing in the $\hat{z}$-direction at spin i, and
3. neighboring magnets are coupled via either ferromagnetic (i.e. we prefer their z-components to have the same sign) or antiferromagnetic (i.e. we prefer their z-components to have opposite signs) coupling, according to whether J_{ij} is positive or negative.

The resulting Hamiltonian mirrors the quantum Hamiltonian described earlier:

$$H(t) = -A(t)\sum_i \sin\theta_i - B(t)\left(\sum_i h_i\cos\theta_i + \sum_{1\le i<j\le n} J_{ij}\cos\theta_i\cos\theta_j\right).$$

Note that a natural update procedure for this model in the absence of noise is to have each spin i simply align with the net effective field at that location, which is $A(t)\hat{x} + B(t)\hat{z}(h_i +$

$\sum_{1\le i<j\le n} J_{ij}\cos\theta_j$), i.e., it should make an angle θ_i with the z-axis where

$$\tan\theta_i = \frac{A(t)}{B(t)\cdot\left(h_i + \sum_{1\le i<j\le n} J_{ij}\cos\theta_j\right)}.$$

However, since a quantum annealer is implemented at a finite temperature i.e. in a very noisy setting, neither the above update rule nor the ordinary differential equations from classical spin dynamics are likely to accurately describe the dynamics of the system. Instead, we perform a Metropolis-type update in order to simulate the effects of finite temperature T. That is, at each time step,

1. we pick a random angle $\theta_i' \in [0, 2\pi)$ for each spin i, and
2. update θ_i to θ_i' with probability $\max\{1, e^{-\Delta E_i/T}\}$ where

$$\begin{aligned}\Delta E_i &= H(t)|_{\theta_i=\theta_i'} - H(t)\\ &= -A(t)(\sin\theta_i' - \sin\theta_i) - B(t)h_i(\cos\theta_i' - \cos\theta_i)\\ &\quad -B(t)\sum_{1\le i<j\le n} J_{ij}\cos\theta_j(\cos\theta_i' - \cos\theta_i).\end{aligned}$$

Note that our update procedure can be regarded as the direct analogue of simulated annealing for 2D vectors instead of bits. It is clear by construction that the model can be simulated using n elementary processors, one for each spin, and the simulation time is linear, $O(T)$. Moreover, the model retains the geometric locality of the quantum annealer.

Further justification for the physical concept of our model can be obtained by viewing the model in terms of the Bloch sphere. Namely, one can interpret the 2D vectors in our model as Bloch sphere vectors for the corresponding qubits, in which case the expectations of observables σ_i^x and σ_i^z are given by

$$\begin{aligned}\langle\sigma_i^x\rangle &= \left(\cos\left(\frac{\theta_i}{2}\right)\langle 0| + \sin\left(\frac{\theta_i}{2}\right)\langle 1|\right)\sigma_i^x\left(\cos\left(\frac{\theta_i}{2}\right)|0\rangle + \sin\left(\frac{\theta_i}{2}\right)|1\rangle\right)\\ &= \left(\cos\left(\frac{\theta_i}{2}\right)\langle 0| + \sin\left(\frac{\theta_i}{2}\right)\langle 1|\right)\left(\cos\left(\frac{\theta_i}{2}\right)|1\rangle + \sin\left(\frac{\theta_i}{2}\right)|0\rangle\right)\\ &= 2\sin\left(\frac{\theta_i}{2}\right)\cos\left(\frac{\theta_i}{2}\right)\\ &= \sin(\theta_i),\\ \langle\sigma_i^z\rangle &= \left(\cos\left(\frac{\theta_i}{2}\right)\langle 0| + \sin\left(\frac{\theta_i}{2}\right)\langle 1|\right)\sigma_i^z\left(\cos\left(\frac{\theta_i}{2}\right)|0\rangle + \sin\left(\frac{\theta_i}{2}\right)|1\rangle\right)\\ &= \left(\cos\left(\frac{\theta_i}{2}\right)\langle 0| + \sin\left(\frac{\theta_i}{2}\right)\langle 1|\right)\left(\cos\left(\frac{\theta_i}{2}\right)|0\rangle - \sin\left(\frac{\theta_i}{2}\right)|1\rangle\right)\\ &= \cos^2\left(\frac{\theta_i}{2}\right) - \sin^2\left(\frac{\theta_i}{2}\right)\\ &= \cos(\theta_i).\end{aligned}$$

Hence, the classical Hamiltonian defined by our model can be written as

$$H(t) = -A(t)\sum_i \langle\sigma_i^x\rangle - B(t)\left(\sum_i h_i\langle\sigma_i^z\rangle + \sum_{1\le i<j\le n} J_{ij}\langle\sigma_i^z\rangle\langle\sigma_j^z\rangle\right),$$

which is nothing but the Hamiltonian obtained by replacing each Pauli matrix in the original quantum Hamiltonian by its expectation. This corresponds to a popular technique in quantum physics called mean-field approximation, which is widely used on systems in which entanglement is expected not to play a crucial role. This provides a sense in which our model is a natural classical analogue of quantum annealing, derived under the single assumption of the lack of entanglement. Moreover, this suggests a natural way to generalize our model to work for arbitrary quantum Hamiltonians; namely, under the assumption that the system is in some product state $|\psi\rangle$, any given quantum Hamiltonian $H = \sum_i H_i$ can be approximated by replacing H_i with $\langle H_i\rangle_\psi$. While in this chapter we are restricting the Bloch vectors to be on the XZ plane, we may need to allow them to use all three dimensions if we were to deal with more general Hamiltonians that also involve σ^y terms.

Compared to other classical models for quantum annealing, our model possesses several advantages. First, unlike simulated annealing, our model is capable of modeling the transverse field $-A(t)\sum_i \sigma_i^x$ thanks to its adoption of 2D vector representation. In contrast, the state representation in simulated annealing allows only $+\hat{z}$ or $-\hat{z}$ for each spin and the introduction of a transverse field will not have any effect at all. Second, unlike classical spin dynamics, our model is capable of modeling the finite temperature T of the quantum annealer, using the Metropolis update rule. Third, unlike simulated quantum annealing, our model is simple and natural. We note that simulated quantum annealing introduces an additional spatial dimension that does not have an immediate physical interpretation, a factor that makes it difficult to understand and analyze.

Before we conclude the section, we remark that we have considered a particularly simple noise model in our classical model. We note that there are situations where it can be helpful to consider more general noise models. For example, in general the fluctuations of the local z-fields relative to those of the z-z interactions may be governed by an extra parameter. In certain regimes, this new parameter may be critically important in determining the qualitative behavior of the model. This has implications for the quantum Turing test: instead of testing against a single model, it must search in a small neighborhood around the core model. In other words, in such regimes, the classical benchmark should be regarded as a family of models.

3.4 Benchmarking the D-Wave machine

In this section, we demonstrate the effectiveness of the quantum Turing test by applying it to the D-Wave machine, which is the only commercially available, and therefore most accessible, quantum annealer to date.

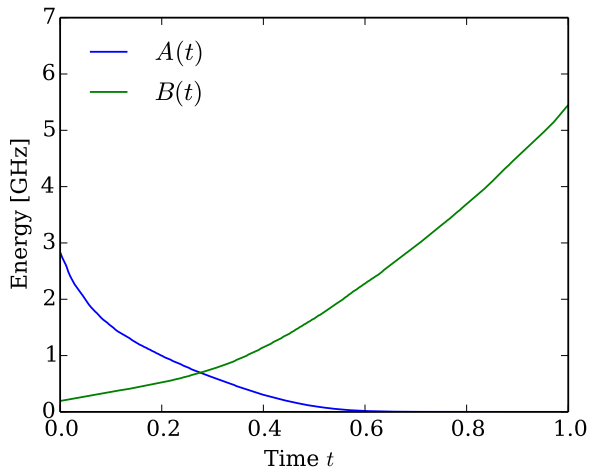

Figure 3.5: The annealing schedule of D-Wave One (figure adapted from the supplementary materials of [24]). In the actual implementation, the effective schedule may slightly vary from spin to spin (see the supplementary materials of [24] for details).

A leading figure in experimental quantum annealing, D-Wave Systems has been making headlines in various academic and mainstream media since 2011, when they released their first commercial product D-Wave One. A quantum annealer with 108 qubits, D-Wave One was soon reported to exhibit genuine quantum signatures in a few scientific papers [25, 74, 45, 24] and its 503-qubit successor D-Wave Two was even claimed to outperform conventional software solvers by a factor of 3600 [81]. After a few years of rapid scaling, their latest product D-Wave 2X, released in 2015, operates with as many as 1097 qubits, and reportedly outperforms simulated annealing by a factor of 10^8 on a specially designed class of test instances [43].

These D-Wave machines implement the quantum annealing model as introduced in the beginning of Section 3.3.4 with a particular choice of the annealing schedule and the interaction graph. For example, Figures 3.5 and 3.6 depict D-Wave One's annealing schedule and interaction graph. While the choice of its annealing schedule is straightforward in that it follows the usual boundary conditions $A(1) \approx 0$ and $B(0) \approx 0$, the choice of the interaction graph merits further comments. The interaction graph implemented by the D-Wave architecture is the so-called Chimera graph, which may be described as a 2D lattice with each lattice point replaced by a supernode of 8 vertices arranged as a complete bipartite graph $K_{4,4}$ (see Figure 3.6). The left four qubits in each supernode are coupled vertically in the 2D lattice and the right four qubits horizontally. More specifically, each left qubit is coupled with the corresponding left qubits in supernodes immediately above and below its own supernode, and each right qubit is coupled with the corresponding right qubits in supernodes immediately to the right and left of its own supernode. The number of qubits in a D-Wave processor is determined by the number of supernodes in each row and column of the original

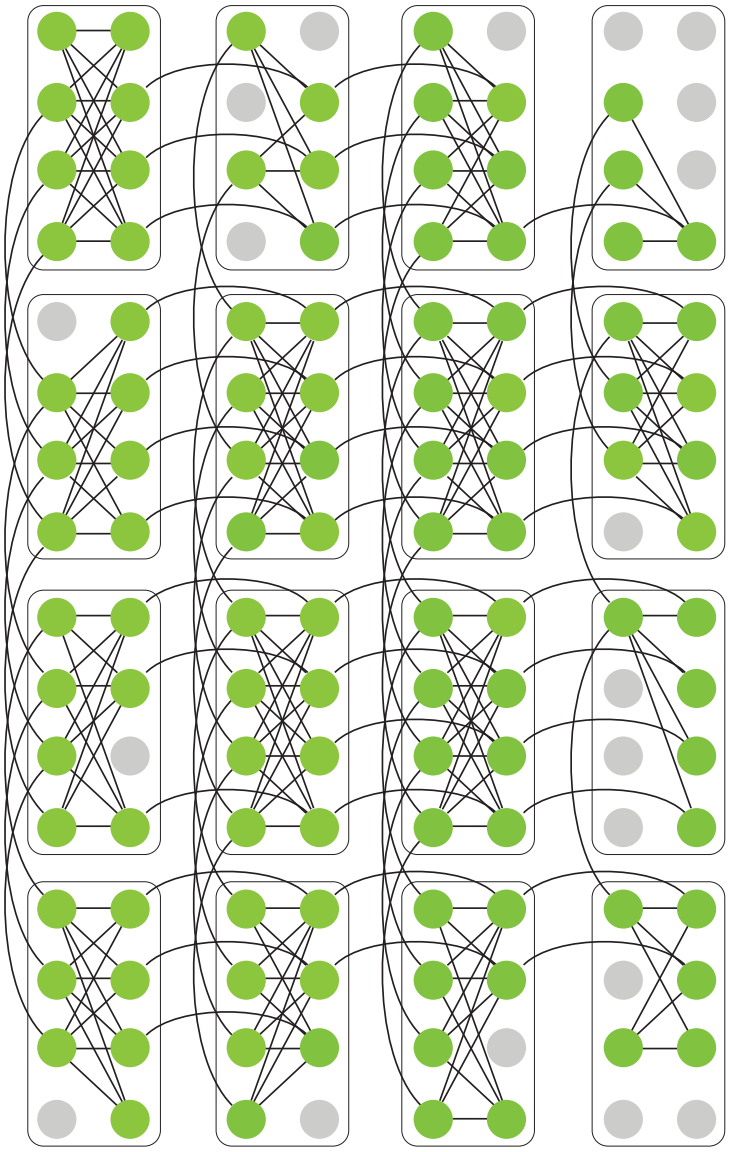

Figure 3.6: The "Chimera" interaction graph of D-Wave One (figure adapted from the supplementary materials of [24]). Due to issues in implementation, not all of the vertices on the Chimera graph represent working qubits on the device. The defective qubits are colored grey and are excluded from experiments.

2D lattice. For instance, D-Wave One has 16 supernodes arranged in a 4×4 lattice, whereas the latest D-Wave 2X has 144 supernodes arranged in a 12×12 lattice. We note that the Chimera graph is not planar, and therefore the problem of minimizing the energy on this graph cannot be handled by the polynomial-time approximation scheme of [15]. On the other hand, Saket [94] recently developed a new polynomial-time approximation scheme designed specifically for Chimera graphs, which means that computing approximate solutions to the class of problems solved by the D-Wave machine is easy even classically. Hence, the hope for achieving an exponential quantum speedup with D-Wave's current architecture can be thought of as mostly with respect to finding the *exact* minimum.

Since the release of D-Wave One, the findings and controversies surrounding the D-Wave machines have generated excited debates not only in the mainstream media but also in the computer industry and the academic community. Many of these debates arose from a number of early reports that seemed to contradict one another. For example, while McGeoch and Wang [81] found D-Wave Two to outperform conventional software solvers by a factor of 3600, Rønnow et al. [93] reported that they found no evidence of quantum speedup compared to simulated annealing, and Selby [99] even reported that there is a classical heuristic that could outperform D-Wave Two by a factor of up to 160. Similarly, while Boixo et al. [26] reported to have found a quantum signature in experiments with D-Wave One, Smolin and Smith [104] claimed that this same quantum signature could be described using entirely classical concepts. The debates about the potential power of the D-Wave machines were further fueled by fact that even among experts opinion is sharply divided about the nature of decoherence in a quantum annealer. On the one hand, the decoherence time (i.e. the time it takes for a qubit to decohere and effectively lose its quantum information) of D-Wave's qubits is reported to be on the order of nanoseconds [25], whereas the total annealing time of the D-Wave machine is typically on the order of microseconds [24]. One way to interpret this is that quantum information in the D-Wave machine keeps evaporating in time that is orders of magnitude shorter than the running time of the algorithm and therefore the algorithm should not be able to exhibit any quantum effects. On the other hand, there are studies [25, 45, 8] that suggest that decoherence is less detrimental to quantum annealers than to general-purpose quantum computers, rendering the short decoherence times of D-Wave's qubits less relevant.

Indeed, it was these controversies and confusions that provided the initial motivation for the research presented in this chapter, as they clearly manifested the impending need for a systematic approach to benchmarking quantum annealers. In this section, we demonstrate the effectiveness of the quantum Turing test and our classical model by applying the test to experimental data from D-Wave quantum annealers.

3.4.1 Random instances of the D-Wave native problem

In our first quantum Turing test, we compare the input-output behavior of the D-Wave machine to that of our classical model on the experimental data reported in [117, 24]. We note that this dataset was used in [24] to claim evidence of quantum tunneling, where the

main argument was based on the behavioral difference between the D-Wave machine and simulated annealing. The dataset records the input-output behavior of D-Wave One on a thousand randomly chosen inputs, noting its probability of finding the exact ground state for each instance. Hence, the input distribution of the quantum Turing test is defined by choosing J_{ij} for each edge $\{i, j\}$ on the Chimera graph to be either -1 or 1 independently at random. Note that this defines a probability distribution D that is uniform on its support and can be thought of as representing the random instances of the native problem of the D-Wave machine. While in practice the D-Wave machine can implement any final Hamiltonian of the form $H = -\sum_i h_i z_i - \sum_{1 \le i < j \le n} J_{ij} z_i z_j$ where $h_i, J_{ij} \in [-1, 1]$, this dataset focuses on the case where $h_i = 0$ and $J_{ij} \in \{-1, 1\}$ because it is suggested in [24] that this case captures the hardest instances of the problem. Indeed, nonzero values of h_i would add a bias to individual spins, which would typically make the problem easier to solve.

The results of this quantum Turing test are presented in Figures 3.7 and 3.8. First, we note that the simulations of our model yield a histogram with clear bimodal signature similar to that of the D-Wave machine, as opposed to the unimodal signature of simulated annealing (Figure 3.7). One interpretation of this histogram is that both the D-Wave machine and our classical model behave more deterministically than simulated annealing on this dataset, as it shows that on most instances they either succeed with high probability (the mode at 1.0) or fail with high probability (the mode at 0.0). Moreover, the success probabilities of our model achieve a remarkably high correlation with the success probabilities of D-Wave One (Figure 3.8), showing that the input-output behavior of D-Wave One on the above input distribution is accurately reproduced by our classical model. Since this implies that quantum tunneling is not necessary to explain the input-output behavior of D-Wave One on this input distribution, it can be formalized as the machine's failure to pass the quantum Turing test on random instances of the D-Wave native problem.

We note that the correlation of 0.91 between D-Wave One and our classical model is slightly higher than the correlation between D-Wave One and simulated quantum annealing reported in [24], which is also around 0.91. Interestingly, a direct comparison between our model and simulated quantum annealing reveals an extremely high correlation of $R \approx 0.99$ (Figure 3.9). This seems to confirm our previous observation that our model may be viewed as a mean-field approximation of quantum annealing, in which the system is assumed to be in a product state at every time step.

3.4.2 Eight-qubit motif problem of Vinci et al.

In this section, we apply the quantum Turing test to the eight-qubit motif problem of [23], which was proposed to distinguish our classical model from the D-Wave machine.

Figure 3.10 depicts the problem Hamiltonian used in the experiments of [116], which has 17 classical solutions with exactly the same energy -8. Indeed, it is easy to see that all of the sixteen classical states in which the four core spins are pointing in the positive direction are ground states of the Hamiltonian. In addition, there is one more ground state with all eight spins pointing in the negative direction. In [116], the first sixteen ground states are

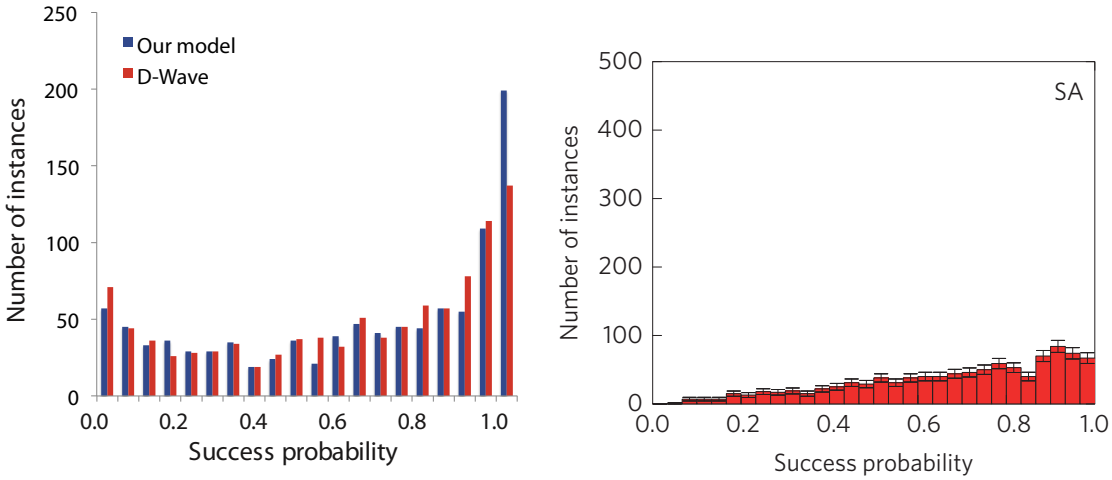

Figure 3.7: Histogram of success probabilities of D-Wave One, our classical model, and simulated annealing (SA). Unlike simulated annealing, the D-Wave machine and our model exhibit a clear bimodal distribution. The histogram for simulated annealing (right panel) was borrowed from [24].

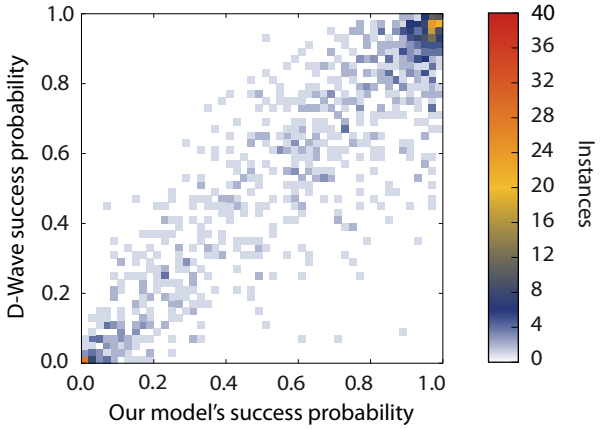

Figure 3.8: Correlation between D-Wave and our model. Each simulation of our model consisted of 150,000 steps, following the annealing schedule of D-Wave One from Figure 3.5. The system temperature of $T = 0.22\text{GHz} \approx 11\text{mK}$ was used. The (Pearson's) correlation coefficient R between the D-Wave One and our model is about 0.91.

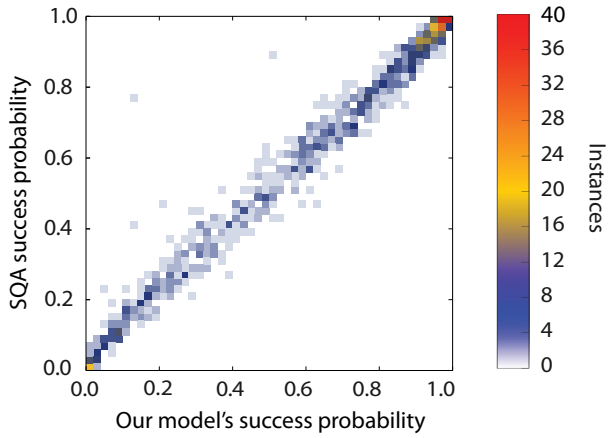

Figure 3.9: Correlation between simulated quantum annealing of [24] and our model. The correlation coefficient R is about 0.99.

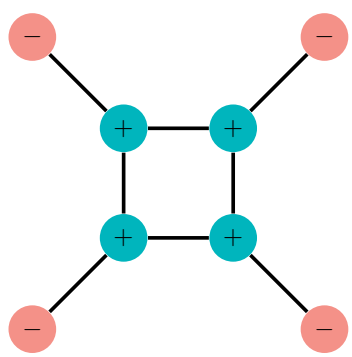

Figure 3.10: The eight-qubit motif problem proposed in [116], which can be mapped to a single supernode of the D-Wave machine. All couplings are ferromagnetic, whereas there is a local z-field applied in the positive direction for the four "core" spins, and in the negative direction for the four "peripheral" spins. Formally, the final Hamiltonian is defined as $H_f = -\sum_i h_i \sigma_i^z - \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z$. The local field h_i is set to be 1 if i is a core spin, and -1 otherwise. The coupling strength $J_{ij} = 1$ for every edge $\{i, j\}$. Figure is borrowed from [116].

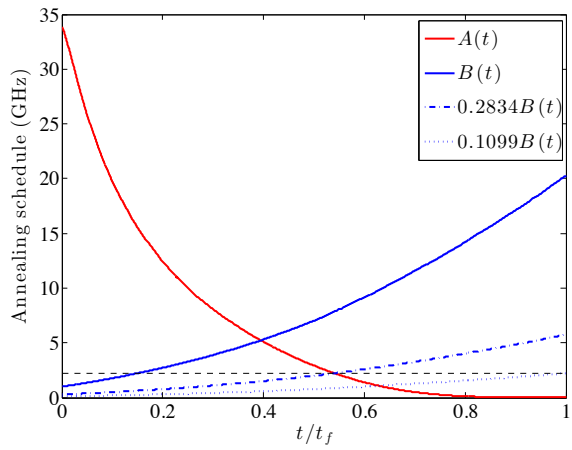

Figure 3.11: The solid curves represent the annealing schedule of D-Wave Two. Dotted blue curves represent the effective annealing schedule for cases $\alpha = 0.2834$ and $\alpha = 0.1099$. The dotted black line represents the system temperature. Figure is borrowed from [116].

called "clustered" ground states because they are all connected by single spin flips, whereas the last ground state is referred to as the "isolated" ground state.

We note that this Hamiltonian was previously used in [25] to distinguish the behavior of the D-Wave machine from that of simulated annealing. As the problem size is fairly small, both the D-Wave machine and simulated annealing almost always succeed in finding one of the 17 ground states. To distinguish between the two, [25] considers the quantity P_I/P_C, where P_I is the probability of seeing the isolated ground state at the end of the process and P_C is the probability of seeing a clustered ground state, divided by 16. They show that the D-Wave machine and the quantum simulation based on adiabatic Markovian master equation (which is exponential but feasible for very small problems) preferred the clustered ground state ($P_I/P_C < 1$), whereas simulated annealing preferred the isolated ground state ($P_I/P_C > 1$). A simple experiment confirms that our classical model also agrees with the behavior of the D-Wave machine and the quantum simulation ($P_I/P_C < 1$).

To distinguish between D-Wave and our model, Vinci et al. [116] perform a more elaborate version of this experiment with an additional control variable α which represents the strength of the final Hamiltonian. Namely, the machine is programmed to implement the time-dependent Hamiltonian $H(t) = A(t)H_0 + \alpha B(t)H_f$, where α is varied in the range $[0, 1]$ as in Figure 3.11. As shown in Figure 3.12, they find that the machine and the adiabatic quantum master equation prefer the isolated ground state ($P_I/P_C > 1$) when α is small, whereas the SSSV model always prefers the clustered ground state ($P_I/P_C < 1$) at all values of α.

It is illuminating to examine more closely the small α regime in which the behaviors of the D-Wave machine and our classical model differ. Since the final Hamiltonian H_f is being scaled by α, a small value of α means that the coupling strength between qubits

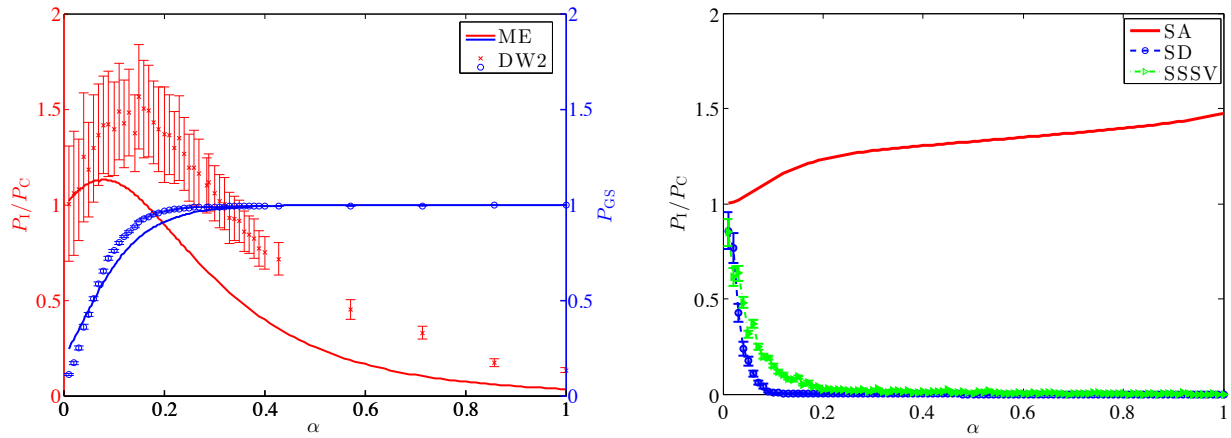

Figure 3.12: Experimental and numerical results from [116]. DW2, ME, SA, SD, and SSSV represent D-Wave Two, quantum adiabatic Markovian master equation (exponential quantum simulation), simulated anneailng, classical spin dynamics, and our classical model respectively. P_{GS} denotes the probability of finding one of the seventeen ground states. Figure is borrowed from [116].

is very small at any given time. This means that the system Hamiltonian is effectively 1-local for the most part, rendering the phenomenon of quantum entanglement rather unlikely. Therefore, the regime of small α may be thought of as a more "classical" regime in which the machine is expected to be driven mostly by thermal fluctuations rather than quantum effects. Furthermore, it can be seen in Figure 3.11 that this regime has the transverse field $A(t)$ nearly die out before the final Hamiltonian becomes strong enough to be able to overcome the system temperature, implying that the transverse field $A(t)$ does not participate in the computation in any meaningful way. However, without the transverse field the system Hamiltonian can be written entirely in the standard basis, which again makes the system classical.

In addition, we note that the smaller the value of α is, the larger the fluctuations in the control parameters of the algorithm would appear relative to the strength of the final Hamiltonian αH_f. Hence, we could predict that the effects of such fluctuations would become more pronounced in this regime. This means that in designing the quantum Turing test for this problem we need to be particularly careful of the fact that our classical model could be highly sensitive to such fluctuations and may change its qualitative behavior in response to them. In this sense, the results of Figure 3.12, which compares the D-Wave machine only to the unaltered version of our classical model, cannot be viewed as a final verdict of the quantum Turing test for the D-Wave machine on this problem. In fact we will presently demonstrate that a simple adoption of Gaussian local noise completely alters our model's behavior in the regime of small α, reproducing the same qualitative behavior observed in the machine and the quantum simulation in Figure 3.12.

Formally, the input distribution for the quantum Turing test on this problem corresponds

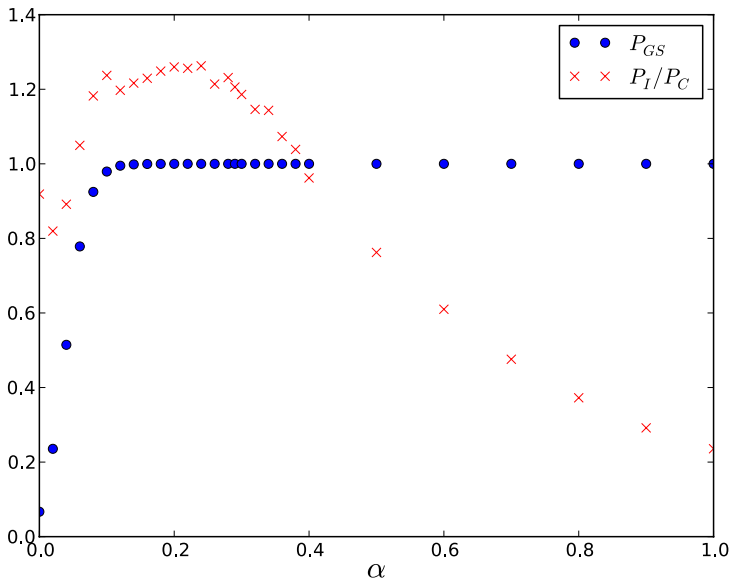

Figure 3.13: Simulations of our classical model with Gaussian local noise. The model produces a behavior similar to that of the D-Wave machine or quantum adiabatic master equation from Figure 3.12. The model was simulated for 1,500 steps at the system temperature of $T = 0.22$GHz. Ten thousand runs were performed for each value of α.

to sampling $\alpha \in [0, 1]$ uniformly at random and using as input the classical Hamiltonian of Figure 3.10 multiplied by α. Since the support of such a distribution is very small, it makes sense to perform the test on an ϵ-net of the support rather than on random samples. Figure 3.13 shows the simulation results of a modified version of our classical model, in which there is a small independent Gaussian noise in the calibration of each local z-field. To be more precise, we simulate the time-dependent Hamiltonian defined as

$$H(t) = A(t) \sum_i \sin\theta_i - \sum_i (B(t) \cdot \alpha \cdot h_i + \epsilon_i) \cos\theta_i - \sum_{i<j} B(t) \cdot \alpha \cdot J_{ij} \cos\theta_i \cos\theta_j,$$

where $\epsilon_i \sim N(0, 0.24)$.[5] Not only does the modified model readily reproduce the signature reported in Figure 3.12, it also reproduces various other signatures suggested in [116] (Figure 3.14).

These results stress the importance of the view that our classical model outlines a family of reasonable models rather than a single model. In the strict sense, establishing that a given phenomenon is truly "quantum" is extremely challenging, since it involves ruling out all possible classical explanations. While this is not practically feasible, it is difficult to

[5]We note that introducing similar Gaussian noise also on the z-z couplings does not seem to affect the simulation results.

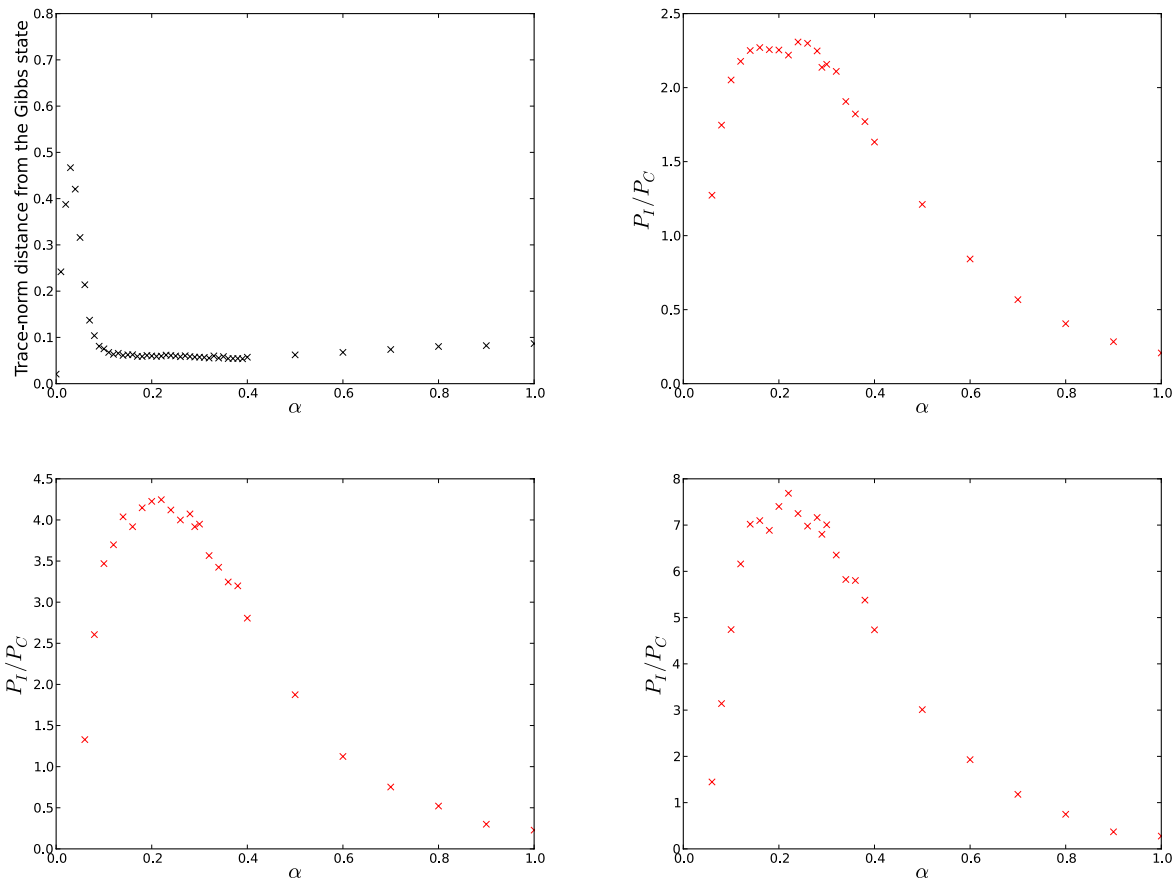

Figure 3.14: Further simulations of our modified classical model reproduce various other signatures suggested in [116]. The top-left panel, which plots the trace-norm distance [84] between the simulated state at the end of the annealing process and the Gibbs state for the final Hamiltonian, is a good qualitative fit to the experimental data presented in Figure 14 of [116]. The top-right, bottom-left, and bottom-right panels are simulation results on larger instances of the problem with 12, 16, and 20 spins respectively (for details about the construction of these instances, see [116]), and are consistent with the experimental results from Figure 10 of [116].

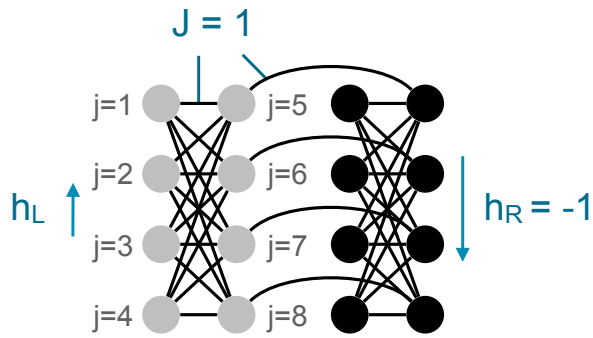

Figure 3.15: The sixteen-qubit motif problem proposed in [23], which can be mapped to two adjacent supernodes of the D-Wave machine. Figure is borrowed from [43].

overemphasize the importance of carefully ruling out a range of classical models contained in a small neighborhood of our core model.

We note that after the above results were posted, Vinci et al. [9] followed up with a more elaborate analysis of the same problem which showed some differences between our augmented model and the D-Wave machine. However, we do not attempt to further pursue this direction because such analyses do not seem to yield a direct computational implication towards the promise of a speedup.

3.4.3 Sixteen-qubit motif problem of Boixo et al.

In this section, we investigate the computational power of the newest D-Wave 2X machine on instances based on the sixteen-qubit motif problem of Boixo et al. This motif problem was initially proposed in [23] to provide evidence of quantum tunneling in the D-Wave machine and was subsequently used as a building block to construct much larger problems of size up to 945. It is reported that D-Wave 2X outperforms both simulated annealing and simulated quantum annealing by a factor of up to 10^8 on these instances [43].

Figure 3.15 depicts a single instance of this sixteen-qubit motif problem. The problem is constructed on two adjacent supernodes on the D-Wave chip by setting all interaction terms to be ferromagnetic ($J = 1$) while also applying local z-fields of size $0 < h_L < 0.5$ and $h_R = -1$ to the two supernodes. Hence, we will refer to these supernodes as the "weak cluster" and the "strong cluster" respectively. Note that the Hamiltonian strongly biases the spins in the strong cluster to point down, whereas the orientation of the weak cluster is determined by the competition between the z-z couplings to the strong cluster and its own local z-fields. This forms a kind of double-well potential in which there are two low-energy states separated by an energy barrier. As long as $h_L < 0.5$, the unique ground state of the Hamiltonian is the state in which every spin points down, whereas there also exists a prominent suboptimal state which has the spins in the weak cluster point up instead. The

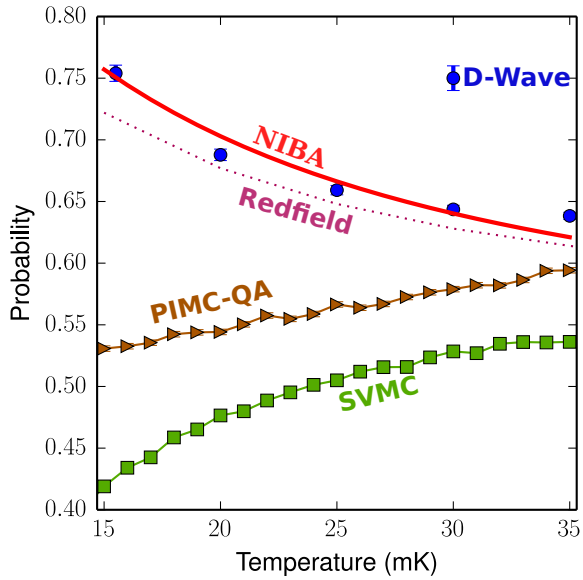

Figure 3.16: The success probabilities of various algorithms as the annealing temperature T is varied. The data points represent the D-Wave machine, quantum simulations based on master equations (NIBA and Redfield), simulated quantum annealing (PIMC-QA), and our classical model (SVMC). Figure is borrowed from [23].

energy gap between these two "wells" is $4 - 8h_L$ and can be tuned by varying the value of h_L. The main idea behind this construction is that quantum annealing may be able to tunnel through the energy barrier between these two wells and therefore find the true ground state of this Hamiltonian with higher probability than classical algorithms. With the choice $h_L = 0.44$, it is reported in [43] that while the D-Wave machine successfully finds the ground state of this Hamiltonian, our classical benchmark tends to find the above-mentioned suboptimal state. Moreover, it is reported that the success probability of the D-Wave machine decreases with annealing temperature T, whereas the success probability of our classical benchmark increases with T. In other words, the performance of the D-Wave machine improves as the rate of thermal effects is turned down, which may be interpreted as implying that the machine's success can mostly be attributed to quantum tunneling. On the other hand, a similar reasoning would render the two classical models to be performing computation mostly by thermal effects.

A perhaps even more interesting set of test instances is studied by Denchev et al. in [43], in which the above 16-qubit motif problem is scaled up by placing many copies of it on the D-Wave chip and then interconnecting them with ferromagnetic or antiferromagnetic couplings. An example problem instance of this type is shown in Fig. 3.17. We note that these instances are by far the largest test instances for the D-Wave machine that are publicly available, and that they have considerably more structure than the instances from [24]. In [43], it is reported

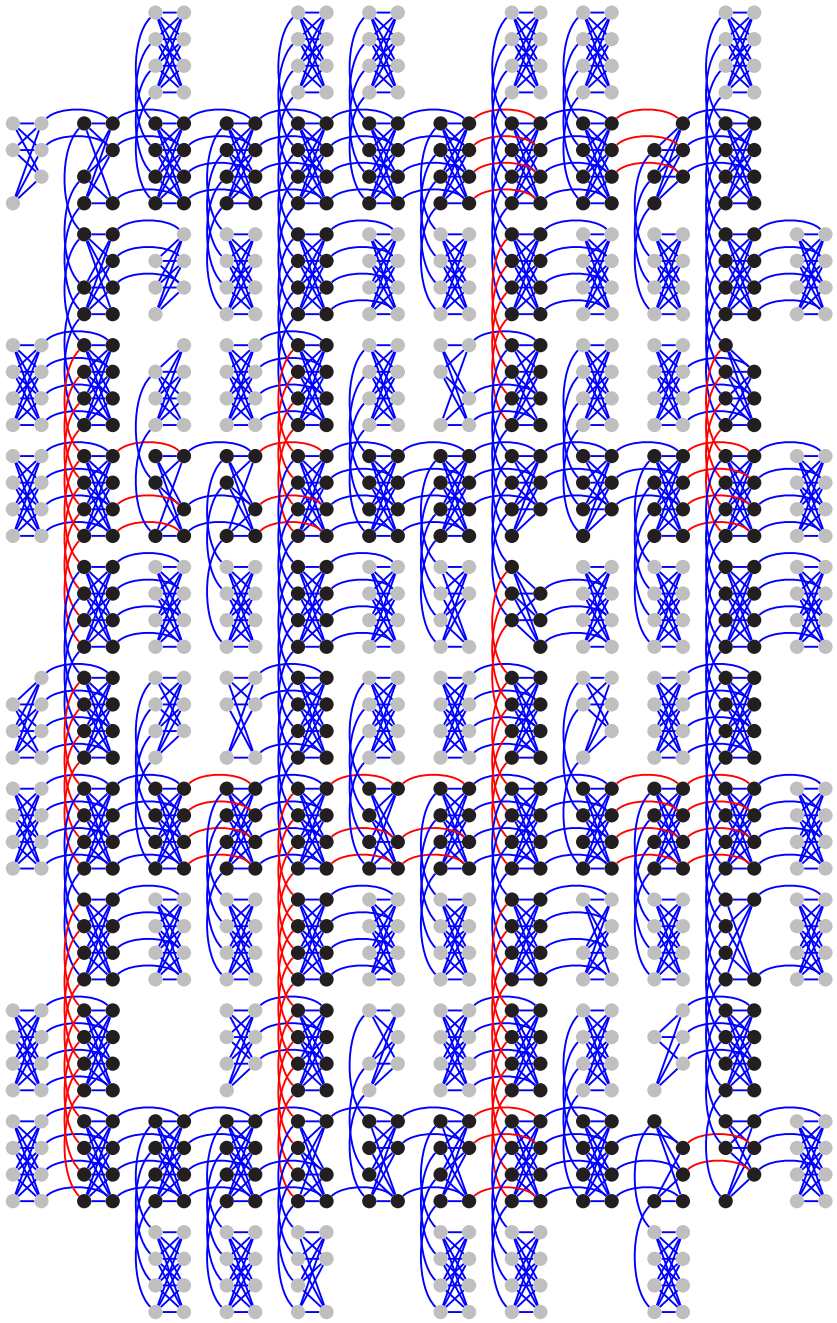

Figure 3.17: A 945-qubit instance constructed in [43] using Boixo et al.'s 16-qubit motif problem as a building block. Black indicates a strong cluster spin and grey indicates a weak cluster spin. Couplings between neighboring strong clusters are chosen to be either ferromagnetic (blue) or antiferromagnetic (red) at random. Figure is borrowed from [43].

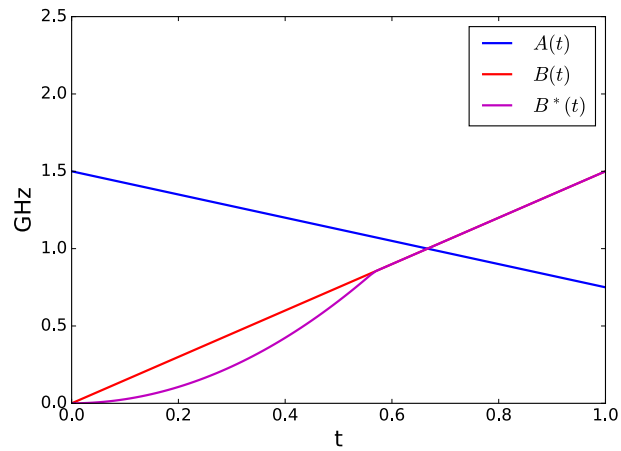

Figure 3.18: A modified annealing schedule used in our quantum Turing test.

that D-Wave 2X outperforms both simulated annealing and simulated quantum annealing by a factor of up to 10^8 times on these instances, which is then interpreted as evidence that quantum tunneling can provide considerable computational advantage.

Probing a small neighborhood of our core model, we find that this 16-qubit motif problem represents a regime in which the behavior of our classical benchmark is highly sensitive to small fluctuations in the control parameters, as suggested by the presence of local z-fields h_i's. Indeed, Figures 3.18 and 3.19 show how our classical model, augmented with a small variation in the annealing schedule parameter $B(t)$, instantly exhibits a performance that is comparable to that of D-Wave 2X. In this augmentation, we assume that the local z-fields follow a slightly different annealing schedule $B^*(t)$ from the z-z interactions, which yields the time-varying Hamiltonian

$$H(t) = -A(t)\sum_i \sin\theta_i - B^*(t)\sum_i h_i \cos\theta_i - B(t)\sum_{1\le i<j\le n} J_{ij}\cos\theta_i\cos\theta_j.$$

For optimization purposes, the annealing schedule that is used in these simulations (Figure 3.18) is different from the machine's physical schedule. However, we note that it is possible to achieve similar input-output behavior using the physical schedule if we alter the annealing schedule of the local z-fields using the same principle. While the running time of D-Wave 2X is on average still an order of magnitude smaller than the running time of our classical benchmark, this advantage is within the software simulation overhead for the classical benchmark. Hence, the results of our quantum Turing test appear to invalidate the interpretation of these test instances as evidence of a quantum speedup in the D-Wave machine.

In addition, we remark that the modified annealing schedule shown in Figure 3.18 can be interpreted as suggesting that the implemented strengths of the local z-fields are slightly weaker than the implemented strengths of the z-z interactions in the earlier half of the annealing process. Even physically, such an assumption is not entirely implausible given the

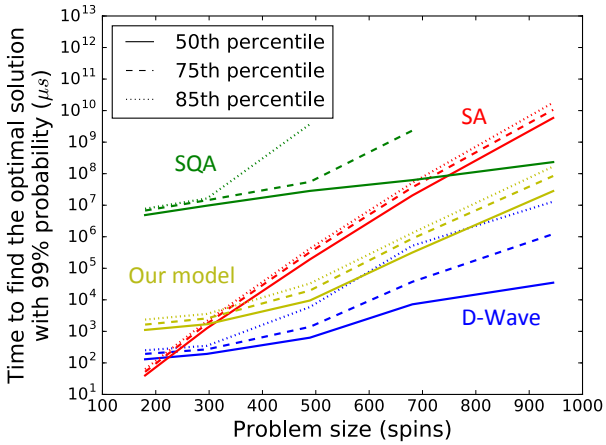

Figure 3.19: Performances of various algorithms on the instances of [43]. The data points for D-Wave 2X, simulated quantum annealing (SQA), and simulated annealing (SA) were taken from [43]. Each run of our classical model consisted of 2,000 steps and the system temperature of $T = 0.22\text{GHz} \approx 11\text{mK}$ was used. Once the success probability s is estimated for each instance, the time to find the optimal solution with 99% probability is calculated as (runtime for one run on a single core)/(# spins) $\cdot \frac{\log(1-0.99)}{\log(1-s)}$, where the factor 1/(# spins) accounts for the amount of parallelism inherent in the D-Wave machine.

difference in implementation of the local fields and of the interactions. However, we stress that such physical interpretations, while potentially of independent interest, are not directly relevant to the verdict of a quantum Turing test – the major thrust of a quantum Turing test is computational, and it treats the machine and the classical model as black boxes.

As a last note, we point out that it is possible to also reproduce the D-Wave machine's behavior on a single instance of the 16-qubit motif problem using the same modified annealing schedule. Under this modification, our model not only matches the success probability of the D-Wave machine on this problem but also reproduces its response to changes in annealing temperature (Figure 3.20). This sharp change in qualitative behavior again stresses the point that our classical model should be viewed as a family of models rather than a single model, especially when used to benchmark noisy devices.

In this section, we have investigated claims of speedup in the D-Wave machine by applying the quantum Turing test on the test instances of [43]. Our results suggest that there is no evidence of a significant speedup with respect to these test instances, and stand in contrast to previous comparisons of the machine to simulated annealing and simulated quantum annealing [43].

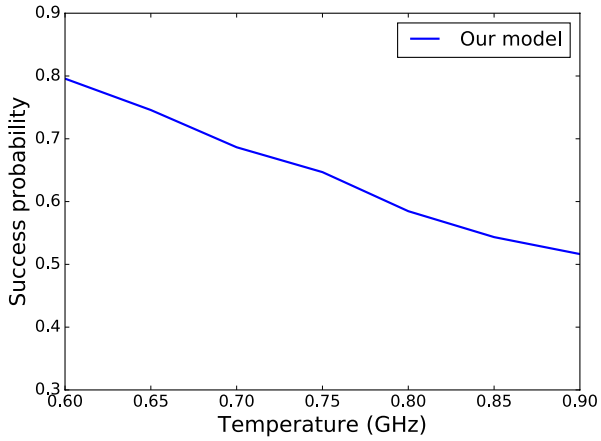

Figure 3.20: The performance of our model on a single instance of the 16-qubit motif problem. The success probability clearly decreases with annealing temperature T, a behavior which was interpreted in [23] as a signature of quantum tunneling. Compare to the D-Wave data in Figure 3.16.

3.5 Discussion

In the previous section, we have shown that the D-Wave machine fails the quantum Turing test with respect to the published test instances from [24] and [43]. In particular, our classical model reproduces both the input-output behavior of the D-Wave machine on random instances of its native problem [24] and its apparent speedup on a more specially designed set of instances [43]. The fact that no other classical model is able to reproduce both aspects of the machine highlights the effectiveness of our classical model in benchmarking quantum annealers.

3.5.1 Synchronized flipping of spins

In addition to being useful in the context of a quantum Turing test, our classical model also suggests interesting insights into the nature of quantum annealing itself. For instance, a closer examination of our simulations on the instances from [24] reveals that our model tends to flip a large cluster of spins simultaneously in the earlier parts of the annealing schedule. Naïvely, such a behavior may appear to resemble global-scale effects due to quantum tunneling and is therefore surprising. The source of this behavior can be traced to the gradation in the magnitude of z-components of spins in the presence of the transverse field (see Figure 3.21). Indeed, we note that in the absence of a transverse field each spin will simply tend to point completely up or completely down, for the energy will always be minimized at one of those two configurations. In the presence of a transverse field, the relative strengths of the net z-field at that location and the transverse field determine the angle at which the spin should

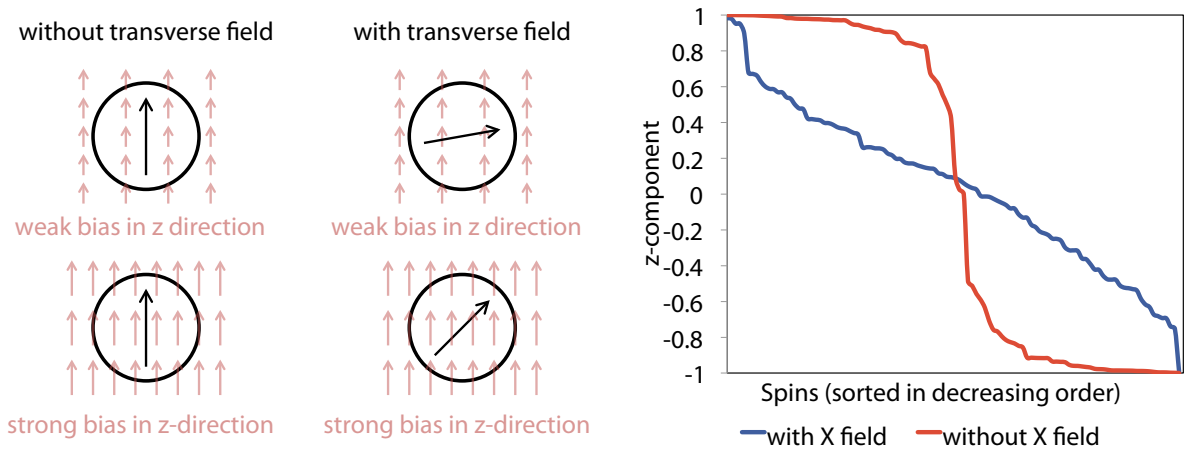

Figure 3.21: Role of transverse field. The right panel is a snapshot of a typical simulation run on the 108-qubit instances of [24], where the z-components of all 108 spins are plotted in decreasing order.

be oriented. To draw an analogy with spectral graph algorithms [37], the two situations are like cuts and eigenvectors of a graph. This is also reminiscent of certain features of belief propagation algorithms [27].

A case study on the instance 13-55-29 of [117] illustrates this phenomenon very clearly. Simulated on this instance, our model always settles into one of two fixed alternatives at time $t = 0.13$, up to the two-fold symmetry of flipping all spins (since the instances of [24] have $h_i = 0$ for every spin i, the energy of the system is preserved under the flipping of all spins). Figure 3.22 shows that the choice between these two alternatives corresponds to deciding the orientation of the "green" cluster with respect to that of the "blue" cluster, in the sense that moving from one alternative to the other corresponds to flipping all the spins in one of these two clusters. An examination of other instances of [24] and other values of t strengthens this observation: alternatives explored by our model invariably correspond to choosing different orientations between large clusters of spins.

How do we account for this similarity between the behavior of our classical model and that of quantum annealing, in particular the "tunneling-like" synchronized flipping exhibited by our model? Our simulations suggest that our classical model exploits the structure of the "Chimera" interaction graph in a very different way from simulated annealing, and possibly closer to the way quantum annealing treats it. For instance, the Chimera graph of D-Wave One consists of 16 supernodes of 8 vertices each, with edge density within a supernode much higher than between supernodes. This structure makes it likely that many supernodes have highly stable (i.e. low-energy) configurations determined mostly by interactions across their internal edges. The two-fold symmetry of flipping all the spins implies that these stable configurations come in pairs. To a first approximation, the major challenge for the algorithm consists in breaking this two-fold symmetry based on the energy contribution from

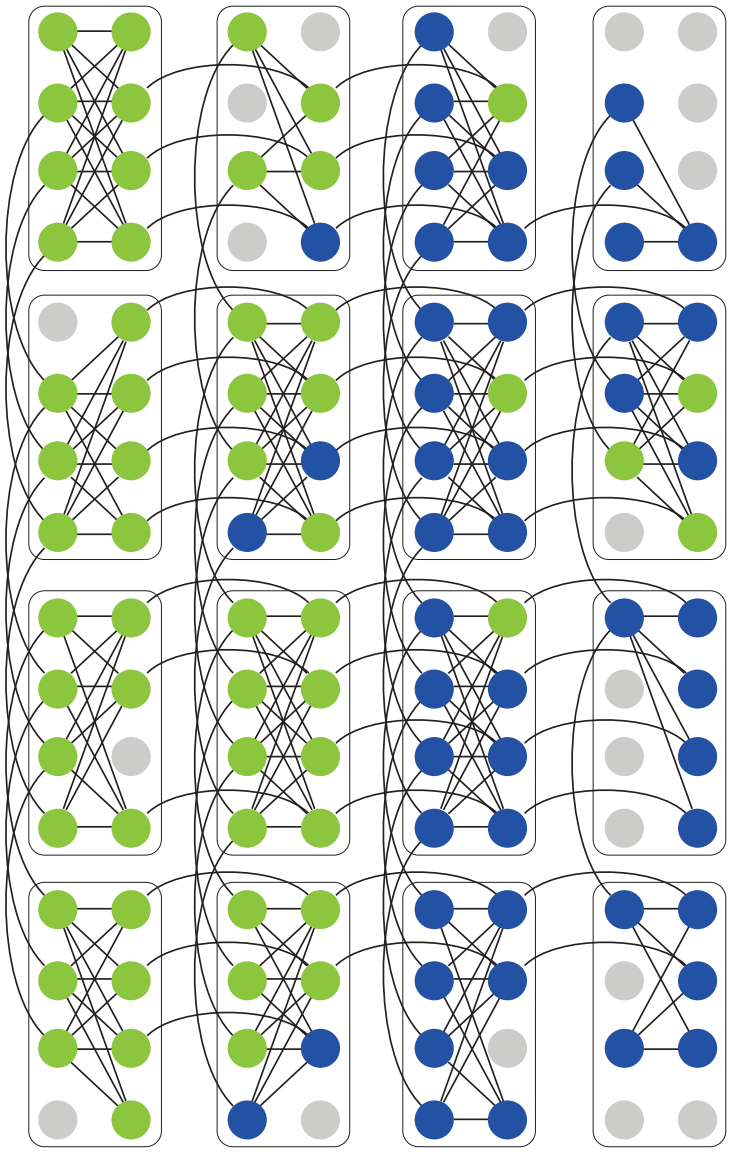

Figure 3.22: The first "branching point" of instance 13-55-29 at $t = 0.13$. There are two alternatives considered by the model at this point, i.e. $H(0.13)$ has two distinct local minima up to the two-fold symmetry of flipping all spins. Blue dots indicate the "difference" between these two alternatives, i.e. blue dots represent the spins on which the signs of z-components differ between the two alternatives. The Chimera graph figure was borrowed and modified from [24].

interactions with other supernodes. Indeed, in our simulations we see smaller and smaller clusters become involved in this type of symmetry breaking process as t increases. That is, we eventually see choices made about the orientation between one or two supernodes and the rest of the system.

This observation provides some insight into the nature of the energy minimization problem on Chimera graphs. Namely, for any algorithm that treats supernodes implicitly as building blocks, the effective problem size may be thought of as closer to the number of supernodes $m = n/8$ rather than the number of spins n. This appears to be the case for our classical model, and in this sense, the search space that the algorithm must explore to solve a 108-qubit problem is quite modest. Unfortunately this also suggests that as the number of qubits is increased, we might see a qualitative change in the difficulty of the problem. This intuition seems to be confirmed in [93], where it is reported that the success probability of the D-Wave machine drops off dramatically on 512-qubit instances.

3.5.2 Deterministic behavior

In this section, we discuss the source of the "deterministic" behavior of our classical model, which was manifested in the bimodal histogram of Figure 3.7. We recall that this was a feature of the D-Wave machine that simulated annealing was unable to reproduce.

To explore this issue, let us examine in more detail the search space explored by our model. First, we observe that our model simplifies to a 2-dimensional analogue of simulated annealing if the transverse field $A(t)$ is omitted. In this case, the time-dependent Hamiltonian of the model reduces to $H(t) = B(t)H_f$, so the Metropolis acceptance probability at time t is given by $\max\{e^{-\Delta H_f B(t)/T}, 1\}$. Since $B(t)$ is increasing in time and T is constant, we see that this exactly corresponds to a 2-dimensional version of simulated annealing that follows the schedule $T'(t) = T/B(t)$. This means that in the later parts of the annealing schedule, where $A(t)$ becomes negligibly small, our model should behave essentially like simulated annealing at low temperature, i.e. like a greedy local search. Indeed, there was no noticeable change to our simulation results on the instances of [24] even if we simply performed the 2-dimensional simulated annealing after $t = 0.31$.

Hence, the main difference in the ways our model and simulated annealing explore the search space lies in the regime where the transverse field $A(t)$ is large (say when $t < 0.31$). In simulated annealing, this is the part of the schedule where the system explores the state space in a manner resembling the uniform random walk. By contrast, the time-dependent Hamiltonian $H(t)$ of our classical model in this regime admits only a small number of local minima, and the system is forced to settle into one of those local minima. For example, it is easy to prove that it has only one local minimum when $t < 0.06$ (with respect to the D-Wave One annealing schedule of Figure 3.5), as the z-z interactions are still negligibly small up to this point. Moreover, it is empirically observed that our model reaches only a handful of distinct local minima even at $t = 0.31$. Combined with the previous observation that the model behaves like a local search afterwards, this provides an explanation of the apparent

determinism of our classical model and the reason that it produces a bimodal histogram rather than a unimodal histogram.

3.6 Conclusions

One of the central open questions about quantum annealing is whether it supports large-scale tunneling that can meaningfully contribute to the computation. On the one hand, toy examples have been constructed on which adiabatic quantum computers successfully tunnel through large energy barriers that simulated annealing is unable to surmount [48, 91]. On the other hand, evidence from simulations based on insights from Anderson localization suggests that such success may not extend to more general settings [11]. In any case, finding a signature of large-scale quantum tunneling seems to be a crucial milestone in the quest for a successful quantum annealer, for it is the very basis of the optimism that quantum annealers can achieve a speedup. Our results suggest that this task is challenging, since purely classical models like ours can exhibit behaviors that appear to be "quantum."

On the other hand, the quantum Turing test effectively avoids such complications by focusing more on the black box behavior of the machine and the classical model. In this paper, we have demonstrated the effectiveness of this approach both in terms of testing quantum coherence and testing for a quantum speedup. Since the posting of our preprint, our approach has helped to shape directions of much research in the field [9, 10, 90, 70, 23, 40, 43, 69], leading to interesting new discoveries about quantum annealing and its implementations. The quantum Turing test is expected to continue to play an important role in the pursuit of special-purpose quantum computation, as an effective null hypothesis against which any claim of quantumness can be tested.

Chapter 4

Tensor Network Nonzero Testing

To be, or not to be – that is the question.

— Hamlet, William Shakespeare

This chapter is based on joint work with Sevag Gharibian, Zeph Landau, and Guoming Wang [57].

4.1 Introduction

4.1.1 Tensor networks in quantum Hamiltonian complexity

In Chapter 2 we saw that tensor networks are a popular tool in condensed matter physics and quantum Hamiltonian complexity because of their ability to efficiently represent certain classes of entangled quantum states. In addition to matrix product states (MPS) [115, 88] we sketched in Section 2.5.2, physicists have developed more general classes of tensor networks such as projected entangled pair states (PEPS) [113] or multiscale entanglement renormalization ansatz (MERA) [114] to encode various quantum states that arise in nature. Among these, the study of the ground states of local Hamiltonians is considered particularly important, because it is closely related to many interesting physical phenomena that are observed at low temperature, such as superconductivity or superfluidity.

Unfortunately, not all such tensor networks can be as useful as the matrix product states because of computational efficiency issues. For example, we observed in Section 2.5.2 that the idea that enables efficient computation of various physical quantities on a matrix product state does not seem to generalize to tensor networks whose geometry has more than one spatial dimension. In view of the algorithm that we sketched in that section, contracting tensor networks defined on even a 2D lattice seems to require exponential time. Can we make this intuition rigorous by proving that this problem is hard for some complexity class,

e.g. **NP**? Since the usefulness of a given class of tensor networks is largely determined by their computational feasibility, it is natural to ask such complexity theoretic questions about various classes of tensor networks.

In fact, it turns out that the general problem of contracting a tensor network is computationally even much harder than **NP**. It is known to be complete for the class #**P** [98], which is the complexity class consisting of all the counting problems that are associated with the decision problems in **NP**. For example, the counting problem associated with the satisfiability problem would ask: how many assignments are there that satisfy the given boolean formula f? Of course, knowing the exact count of satisfying assignments implies knowing whether there exists at least one such assignment, so the counting problem is always at least as hard as its decision counterpart.

Given the widely believed hardness gap between the two complexity classes **NP** and #**P**, it is tempting to ask what is the decision problem corresponding to the problem of tensor network contraction, which may be viewed as a counting problem. It seems that one natural candidate is the problem of tensor network nonzero testing, in which we ask whether the contraction of a given tensor network T represents a nonzero tensor (i.e. whether there exists some input on which T outputs a nonzero value). Indeed, the standard reduction from #**P**-complete problems to tensor network contraction works by explicitly constructing a tensor network that counts the number of solutions to a given graph problem. On such constructions, tensor network nonzero testing would directly correspond to the problem of deciding whether the given graph problem has at least one solution. This raises some hope that tensor network nonzero testing could indeed be significantly easier than the #**P**-complete tensor network contraction and might give rise to novel approaches to problems in quantum Hamiltonian complexity.

In this chapter, we study the computational complexity of tensor network nonzero testing. We show that, contrary to what the above analogy seems to suggest, tensor network nonzero testing is not expected to be much easier than tensor network contraction in the most general case. On the other hand, this result does not entirely rule out the potential of this approach; by identifying easy special cases of tensor network nonzero testing, we are immediately able to make some nontrivial discoveries in quantum Hamiltonian complexity.

4.1.2 Counting problems vs. decision problems

In complexity theory, there is a standard way of converting decision problems to their corresponding counting versions. For instance, if we take any problem in **NP** and a corresponding polynomial-time verifier, the problem can be swiftly translated into a counting version by asking for the number of witnesses that will cause the verifier to accept. For most problems this can simply be understood as counting the number of solutions, e.g. the number of 3-colorings on a given graph, whereas their decision counterparts would ask whether there exists at least one solution, e.g. "is this graph 3-colorable?"

We note that such counting problems indeed have a distinguished history in the study of classical physics. For example, in statistical mechanics there is an object called "parti-

tion function," which is essential to computing probability distributions that arise in nature. The partition function, despite the slightly misleading name, can be thought of as the normalization constant to a probability density function defined over classical states of a given system. Since the number of possible classical states grows exponentially in the system size, computing the partition function generally corresponds to computing the weighted sum of an exponential number of terms. It is not difficult to see that the essence of this computation is similar to that of many counting problems [59]. Interestingly, computing the norm of an unnormalized quantum state, which may be expressed as a tensor network contraction problem, can be viewed as a quantum analogue of the computation of the partition function.

The computational difficulty of counting problems, physical or non-physical, has been extensively studied in the literature and is captured in the complexity class $\#\mathbf{P}$ [110]. While it is obvious that any counting problem is at least as hard as its decision counterpart, which implies that any $\#\mathbf{P}$-complete problem is at least also $\mathbf{NP}$-hard, it is unclear a priori exactly how much harder $\#\mathbf{P}$-complete problems may be compared to $\mathbf{NP}$-complete problems. An elegant answer to this question is provided by the celebrated theorem of Toda [107], which relates the computational complexity of counting problems to that of the so-called polynomial hierarchy ($\mathbf{PH}$). To define the polynomial hierarchy, let us imagine a hypothetical world in which we have access to an oracle with a mystical power to solve any problem in $\mathbf{NP}$ in an instant. For example, in that hypothetical world, all $\mathbf{NP}$-complete problems will be solvable in polynomial time by simply invoking the oracle once. If we denote by $\mathbf{P}^{\mathbf{NP}}$ the class of problems that can be solved in polynomial time with the help of such an "$\mathbf{NP}$ oracle," the above statement can be expressed as $\mathbf{NP} \subseteq \mathbf{P}^{\mathbf{NP}}$. Similarly, we can define $\mathbf{NP}^{\mathbf{NP}}$ to be the class of problems that can be solved in nondeterministic polynomial time with the help of an $\mathbf{NP}$ oracle. This then naturally defines an infinite hierarchy of such oracle complexity classes as follows:

$$\mathbf{NP} \subseteq \mathbf{NP}^{\mathbf{NP}} \subseteq \mathbf{NP}^{\mathbf{NP}^{\mathbf{NP}}} \subseteq \mathbf{NP}^{\mathbf{NP}^{\mathbf{NP}^{\mathbf{NP}}}} \subseteq \cdots$$

Then the complexity class $\mathbf{PH}$, named the polynomial hierarchy, is simply defined as the union of $\Sigma_i^{\mathbf{P}} := \overbrace{\mathbf{NP}^{\mathbf{NP}^{\mathbf{NP}^{\cdots\mathbf{NP}}}}}^{i}$ over all i (Figure 4.1). Toda's theorem relates the computational complexity of the polynomial hierarchy to that of counting problems by proving that counting problems are at least as hard as the polynomial hierarchy, i.e. $\mathbf{PH} \subseteq \mathbf{P}^{\#\mathbf{P}}$. Since a widely believed conjecture is that the polynomial hierarchy does not collapse, i.e. $\Sigma_i^{\mathbf{P}} \neq \Sigma_{i+1}^{\mathbf{P}}$ for every i, the theorem can be interpreted as implying a large hardness gap between $\mathbf{NP}$ and $\#\mathbf{P}$.

We also note that, somewhat surprisingly, a decision problem and its corresponding counting problem may not always have matching complexity. For instance, counting the number of perfect matchings is famously $\#\mathbf{P}$-complete [110], whereas deciding whether there exists at least one perfect matching is in $\mathbf{P}$. Such potential hardness gaps between a decision problem and its counting version provide further motivation to study tensor network nonzero testing, as they raise hope that tensor network nonzero testing could turn out to be much easier than tensor network contraction, at least in certain special cases.

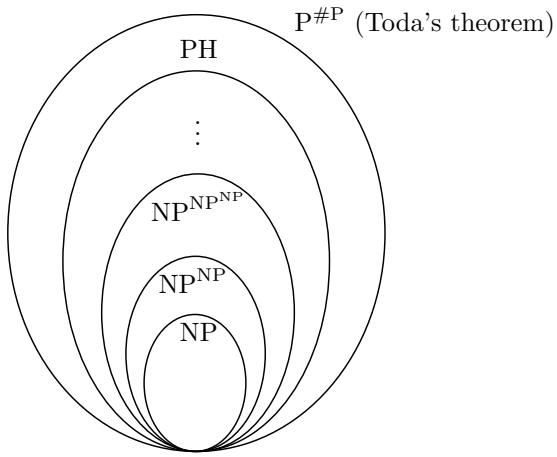

Figure 4.1: The polynomial hierarchy.

Our main result about the hardness of tensor network nonzero testing states that tensor network nonzero testing is not contained in the polynomial hierarchy unless the hierarchy collapses. Even though it is not known whether there is a gap between **PH** and $\mathbf{P}^{\#\mathbf{P}}$, our result may still be viewed as providing strong evidence that tensor network nonzero testing is computationally "very hard." On the other hand, certain special cases of tensor network nonzero testing can be identified which are complete for the much easier class of **NP**. By connecting these results to problems in quantum Hamiltonian complexity, we demonstrate the sense in which identifying "easy" special cases of tensor network nonzero testing could directly lead to discoveries in quantum Hamiltonian complexity.

4.1.3 Commuting local Hamiltonians

While our results do clearly demonstrate the hardness gap between the general problem of tensor network nonzero testing and some of its special cases, one might ask why such gaps should matter at all, given that the complexity classes that feature in our results are, "hard" or "easy," all unlikely to be solvable in polynomial time. If we intend to use certain special cases of tensor network nonzero testing to handle problems in quantum Hamiltonian complexity, and those special cases turn out to be **NP**-complete, is there any useful result that can still be derived from them?

We note that the witness-based definition of **NP** provides an interesting view of tensor network nonzero testing that makes this possible. To see this, we recall that the local Hamiltonian problem is known to be complete for the class **QMA** [78], the class of problems that admit an efficient quantum witness that is verifiable on a quantum computer. Assuming the widely believed conjecture $\mathbf{NP} \neq \mathbf{QMA}$, this implies that there is no efficiently verifiable classical witness that could attest to the fact that a given Hamiltonian has small ground energy. This seems to be related to the inherent exponential nature of quantum states, for

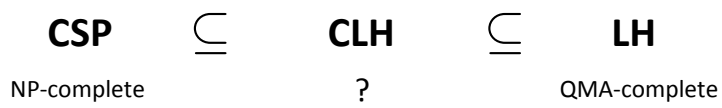

Figure 4.2: The computational complexity of the commuting local Hamiltonian problem is not known in the general case.

if there were a way to efficiently describe quantum states classically, then the ground state itself could serve as an effective classical witness. In stark contrast, we note that any **NP**-complete special case of the local Hamiltonian problem would by definition admit such an efficient classical witness. While finding this witness would still likely be intractable, the existence of such a witness is already significant in many ways. For example, it implies that there are interesting features (e.g. ground energy) of the given class of local Hamiltonians that can be efficiently described classically, which raises questions about how "quantum" such systems should really be considered to be. In fact, we have already seen in Section 2.3 that the classical constraint satisfaction problems can be viewed as an **NP**-complete special case of the local Hamiltonian problem.

An important open problem that can guide research in this direction is the commuting local Hamiltonian problem, originally introduced in [32]. A special case of the local Hamiltonian problem, the commuting local Hamiltonian problem is obtained by introducing the additional restriction that the local terms in the given Hamiltonian must pairwise commute. Equivalently, this means that there must be a basis that simultaneously diagonalizes all of the local terms. Hence, this problem represents an intriguing middle ground between the quantum local Hamiltonian problem and the classical constraint satisfaction problem, the latter of which differs from the commuting local Hamiltonian problem only in that the diagonalizing basis is further restricted to be the standard basis. In this sense, the commuting local Hamiltonian problem can be seen as a relaxation of the constraint satisfaction problem, and its computational complexity must lie somewhere between that of the constraint satisfaction problem and that of the local Hamiltonian problem.

Noting the hardness gap between these two problems, one would naturally wonder whether the computational complexity of the commuting local Hamiltonian problem is closer to **NP** or to **QMA**. Obviously this question is crucial to understanding whether such Hamiltonians should be thought of as being closer to classical Hamiltonians or to quantum Hamiltonians, and may even provide an important clue as to which features of quantum local Hamiltonians are responsible for the complexity jump from **NP** to **QMA**. While the fact that non-commutativity of quantum observables is the basis of many important quantum phenomena such as Heisenberg's uncertainty principle may be suggesting that commuting local Hamiltonians should essentially be classical, it is known that they can nonetheless exhibit highly nontrivial quantum phenomena such as long-range entanglement. A prominent example of this is the toric code [44].

Previous results on the subject place many special cases of the commuting local Hamiltonian problem in **NP**, showing that in those cases the problem can indeed be considered classical. Namely, it is known that commuting 2-local Hamiltonians are **NP**-complete [32], commuting 3-local Hamiltonians on qubits (i.e. the dimension of each particle is 2) are **NP**-complete [2], and even commuting 4-local Hamiltonians on 2D lattices of qubits are **NP**-complete [96]. There are also other results [68, 3, 120] whose conditions are more complicated than simply limiting the degree of locality or the particle dimension. However, the computational complexity of the commuting local Hamiltonian problem largely remains open, as none of these results seems to extend to the case where the degree of locality is greater than 4 and there is no progress towards proving that even the most general case of the commuting local Hamiltonian problem is **QMA**-complete.

Interestingly, the approach of Schuch [96] makes an implicit use of a tensor network representation of the ground space (i.e. the eigenspace corresponding to the lowest eigenvalue) of a given commuting local Hamiltonian, which can effectively serve as a classical witness to the fact that the Hamiltonian has small ground energy. Furthermore, it turns out that the verification of such witnesses involves solving an instance of tensor network nonzero testing. This explains how identifying **NP**-complete special cases of tensor network nonzero testing can automatically place certain classes of commuting local Hamiltonians in **NP**, and one such case will be explicitly demonstrated in Section 4.5 to argue for the usefulness of tensor network nonzero testing.

4.2 Preliminaries

In this section, we introduce the preliminaries and notations that will be used in the further exposition of the subject.

4.2.1 Polynomial hierarchy

To formally define the polynomial hierarchy, we first introduce the notion of oracle Turing machines.

Definition 4.1. Let A be any computational problem. An **oracle Turing machine for** A is a Turing machine with the additional ability to query the "oracle" about instances of A. The oracle returns the solution to the queried instance in a single step.

Similarly, if $\mathcal{C}$ is any complexity class, an **oracle Turing machine for** $\mathcal{C}$ is a Turing machine with the ability to query the oracle about any instance of any problem in $\mathcal{C}$.

Definition 4.2. If X is a computational problem or a complexity class, we define $\mathbf{P}^X$ to be the class of problems that can be solved in polynomial time by an oracle Turing machine for X. Similarly, $\mathbf{NP}^X$ is defined as the class of problems that can be solved in nondeterministic polynomial time by an oracle Turing machine for X.

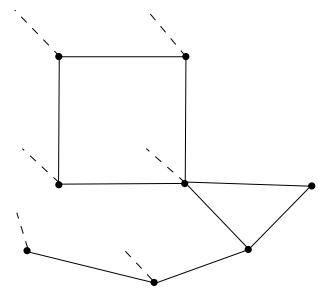

Figure 4.3: When a tensor network is viewed as a quantum state or operator, open edges (dashed lines) are interpreted as corresponding to the physical particles.

Then, the polynomial hierarchy is simply defined as follows:

Definition 4.3.

$$\Sigma_0^{\mathbf{P}} = \mathbf{P},$$
$$\Sigma_{i+1}^{\mathbf{P}} = \mathbf{NP}^{\Sigma_i^{\mathbf{P}}}.$$

Definition 4.4.

$$\mathbf{PH} = \cup_{i=0}^{\infty} \Sigma_i^{\mathbf{P}}.$$

4.2.2 Physical interpretations of tensor networks

We have noted that tensor networks are often used to represent quantum states and operators. In such cases, each open edge of the tensor network can be thought of as corresponding to a physical qubit of the system. For example, the quantum state $|\psi\rangle$ of an n-qubit system is represented by a tensor network $T_{|\psi\rangle}$ with n open edges of bond dimension 2 (Figure 4.3 is an example with $n = 6$), where j-th open edge corresponds to the j-th qubit. If we specify a labeling $i_1, \ldots, i_n \in \{0, 1\}$ on those open edges, the contraction of this tensor network, denoted $T^*_{|\psi\rangle}$, will output the coefficient of the basis vector $|i_1 \cdots i_n\rangle$ in $|\psi\rangle$:

$$T^*_{|\psi\rangle}(i_1, \ldots, i_n) = \langle i_1 \cdots i_n | \psi \rangle.$$

Similarly, a linear operator A for an n-qubit system (i.e. a $2^n \times 2^n$ matrix) is represented by a tensor network T_A with $2n$ open edges of bond dimension 2 (Figure 4.3 is an example with $n = 3$). In this case, the first n open edges are interpreted as representing the n inputs to the linear operator, whereas the last n open edges represent the n outputs of the linear operator. If we specify labels $i_1, \ldots, i_n, j_1, \ldots, j_n$ on these $2n$ open edges, the contraction of the tensor network outputs the corresponding entry of the matrix A:

$$T^*_A(i_1, \ldots, i_n, j_1, \ldots, j_n) = A_{j_1 \cdots j_n, i_1 \cdots i_n} = \langle j_1 \cdots j_n | A | i_1 \cdots i_n \rangle.$$

Hence, in both interpretations the open edges of the tensor network naturally correspond to the physical qubits of the quantum system at hand.

4.2.3 Commuting local Hamiltonians

In this section, we formally define the commuting local Hamiltonian problem.

Definition 4.5. A k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ is said to be **commuting** if for any $1 \leq i, j \leq m$ we have $H_i H_j = H_j H_i$.

Definition 4.6. The commuting k-local Hamiltonian problem is defined as the following promise problem:

- **Input**: A commuting k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ on n particles such that $m = \text{poly}(n)$ and $||H_i|| \leq 1$ (where $|| \cdot ||$ is the operator norm), numbers a and b such that $b - a > 1/\text{poly}(n)$.
- **Output**:
  - YES if the ground energy of H is at most a,
  - NO if the ground energy of H is at least b.

4.2.4 Quantum k-SAT

In this section, we introduce an important variant of the local Hamiltonian problem which is called quantum k-SAT. A special case of the local Hamiltonian problem, quantum k-SAT may be considered a more direct quantum analogue of classical k-SAT in that it asks whether the ground energy of a given local Hamiltonian is *exactly* zero.

Definition 4.7. The quantum k-SAT problem is defined as the following promise problem:

- **Input**: A k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ on n particles such that $m = \text{poly}(n)$ and each H_i is a projection, a number b such that $b > 1/\text{poly}(n)$.
- **Output**:
  - YES if the ground energy of H is zero,
  - NO if the ground energy of H is at least b.

Similarly to the classical case, quantum k-SAT is known to be in $\mathbf{P}$ for $k = 2$ [28] and $\mathbf{QMA_1}$-complete for $k > 2$ [28, 60], where $\mathbf{QMA_1}$ is a variant of $\mathbf{QMA}$ in which YES instances are accepted with probability 1 (perfect completeness). While it remains open whether $\mathbf{QMA_1} = \mathbf{QMA}$, it is plausible that the computational complexity of $\mathbf{QMA_1}$ is close to that of $\mathbf{QMA}$ (in the classical case, it is known that $\mathbf{MA_1} = \mathbf{MA}$ [122]). We note that the commuting version of quantum k-SAT can be naturally defined by adding the same commutativity constraint as in the commuting local Hamiltonian problem, while its computational complexity remains unknown.

4.2.5 Stoquastic Hamiltonians

In this section we introduce stoquastic Hamiltonians, another interesting class of local Hamiltonians we already briefly discussed in Section 3.3.2. The notion of stoquasticity was first formulated by Bravyi et al. [33] to identify those Hamiltonians that can be more efficiently handled on a classical computer. In that paper, for example, the local Hamiltonian problem for stoquastic Hamiltonians is shown to be contained in the complexity class **AM**, which means that there is a classical one-round interactive protocol for proving that a given stoquastic Hamiltonian has small ground energy. Moreover, as we mentioned in Section 3.3.2, it is commonly believed that stoquastic Hamiltonians can be efficiently simulated using a classical algorithm called quantum Monte Carlo [29].

In short, stoquastic Hamiltonians are defined as follows:

Definition 4.8. A k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ is called **stoquastic** if every H_i has real and nonpositive off-diagonal matrix elements.

A useful property of stoquastic Hamiltonians is that they admit Gibbs states whose matrix elements are nonnegative in the standard basis [31].[1] This further implies that the ground state of H has real and nonnegative coefficients in the standard basis [31], which allows one to view it essentially as a classical probability distribution.

We note that we can naturally define the stoquastic local Hamiltonian problem and the stoquastic quantum k-SAT problem by simply adding the stoquasticity constraint. As noted before, the stoquastic local Hamiltonian problem is known to be contained in **AM** [33] and complete for **StoqMA**, which lies between **MA** and **QMA** [30]. The stoquastic quantum k-SAT problem is known to be **MA**-complete. While the above results seem to suggest that stoquastic Hamiltonians could indeed be more classical than quantum, they nonetheless encompass a wide range of Hamiltonians with nontrivial quantum behavior. For instance, the transverse field Ising model, the Heisenberg model on bipartitate graphs, the bosonic Hubbard model, the toric code [44], and the quantum annealing Hamiltonians all fall into the class of stoquastic Hamiltonians. In particular, the ground states of the toric code Hamiltonian are known to manifest exotic quantum phases such as topological order (i.e. long-range entanglement).

4.2.6 Tensor network nonzero testing

Finally we introduce the main object of study of this chapter, which is the problem of tensor network nonzero testing.

Problem 4.1. Generalized tensor network nonzero testing (gTNZ) is defined as follows:

- **Input**: A tensor network T and two numbers a and b such that $b - a \geq 1$.

[1] The Gibbs state for a quantum Hamiltonian H is defined as $\rho = e^{-H/T} / \operatorname{Tr}(e^{-H/T})$, and is widely used in statistical physics to describe the state of a given quantum system at a fixed temperature T.

- **Output**:
  - YES if there exists an input $\vec{x}$ such that $|T^*(\vec{x})| \geq b$ (where T^* denotes the contraction of T),
  - NO if for all $\vec{x}$, $|T^*(\vec{x})| \leq a$.

Problem 4.2. Tensor network nonzero testing (TNZ) is defined as follows:

- **Input**: A tensor network T.
- **Output**:
  - YES if there exists an input $\vec{x}$ such that $T^*(\vec{x}) \neq 0$,
  - NO if for all $\vec{x}$, $T^*(\vec{x}) = 0$.

We note that TNZ can be viewed as a special case of gTNZ in which $a = 0$ and $b = 1$, as we can amplify the gap $b - a$ by multiplying all the entries of the tensors in T by an appropriate scalar.

4.3 Hardness of tensor network nonzero testing

In this section, we show that tensor network nonzero testing is not expected to be much easier than tensor network contraction despite the intuition suggested by the analogy of decision problems vs. counting problems. Namely, we show that generalized tensor network nonzero testing is #**P**-hard, and that tensor network nonzero testing is not contained in **PH** unless the hierarchy collapses.

4.3.1 Generalized tensor network nonzero testing

Theorem 4.1. Generalized tensor network nonzero testing (gTNZ) is #**P**-hard.

Proof. We prove this by exhibiting a polynomial-time Turing reduction from #PERFECT-MATCHING to gTNZ. We note that the former problem is #**P**-complete even on 3-regular graphs [109].

Suppose we are given an instance of #PERFECT-MATCHING, namely an arbitrary 3-regular graph $G = (V, E)$. We will construct a tensor network T on this graph such that its contraction encodes the number of perfect matchings in G. To achieve this, we set the bond dimension of every edge to be 2, and make each vertex $v \in V$ correspond to the tensor A defined as follows:

$$A(i, j, k) = \begin{cases} 1 & \text{if exactly one of } i,\ j, \text{ and } k \text{ is } 1, \\ 0 & \text{otherwise.} \end{cases}$$

Now, T is a closed tensor network without any open edge, so its contraction is simply a scalar. By the definition of contraction, it is computed as

$$\sum_{i_1,i_2,\ldots,i_{|E|}\in\{0,1\}} T(i_1,\ldots,i_{|E|}) = \sum_{i_1,i_2,\ldots,i_{|E|}\in\{0,1\}} \prod_{\substack{v\in V \\ i_{v_1},i_{v_2},i_{v_3}\in E(v)}} A(i_{v_1},i_{v_2},i_{v_3}).$$

Noting that each summand is 1 if and only if $i_1,\ldots,i_{|E|}$ corresponds to a perfect matching, we see that the contraction of T indeed encodes the number of perfect matchings in the graph G. We remark that a similar construction based on 3-COLORING was used in [13] to sketch #**P**-hardness of tensor network contraction.

To complete the Turing reduction, it suffices to show that we can efficiently compute the contraction of T with an oracle Turing machine for gTNZ. Indeed, it is obvious that the oracle Turing machine can achieve this using binary search. □

4.3.2 Tensor network nonzero testing

An immediate corollary of the construction used in Theorem 4.1 is that TNZ is at least **NP**-hard. In fact, the construction suggests an intriguing analogy between tensor network contraction and counting problems, in which tensor network nonzero testing would correspond to the decision problems associated with those counting problems. This raises a natural question: could TNZ also be in **NP**? Unfortunately, the following theorem seems to answer it in the negative.

Theorem 4.2. If TNZ is in $\Sigma_i^{\mathbf{P}}$, then $\mathbf{PH} \subseteq \Sigma_{i+2}^{\mathbf{P}}$, i.e., the polynomial hierarchy collapses to the $(i+2)$-nd level.

Proof. To prove the theorem, let A denote the following computational problem.

- **Input**: A 3-regular graph $G = (V, E)$ and a nonnegative integer k.
- **Output**:
  - YES if the number of perfect matchings in G is at least k,
  - NO otherwise.

As a first step, we show that $A \in \mathbf{NP}^{\mathrm{TNZ}}$ by constructing an oracle Turing machine for TNZ that decides A in nondeterministic polynomial time. This oracle Turing machine can be described as follows:

1. Nondeterministically guess $k \le k' \le 3^n$ where $n = |V|$. (Note that the number of perfect matchings in a 3-regular graph never exceeds 3^n, for each vertex is allowed to choose only one of its three incident edges.)
2. As in the proof of Theorem 4.1, construct a closed tensor network T whose contraction encodes k^*, the number of perfect matchings in the given graph G.

3. Manipulate T to obtain another closed tensor network T' whose contraction encodes $k^* - k'$. While there are multiple ways to achieve this, one way may be summarized as follows. First, we increase the bond dimension of every edge in T by one. In our case, every edge of T has bond dimension of 2, so edges of T' will have bond dimension of 3. Moreover, if we denote by A_v the tensor associated with vertex v in T, the corresponding tensor A'_v in T' is defined as

$$A'_v(i,j,k) = \begin{cases} A_v(i,j,k), & \text{if } i,j,k \in \{0,1\}, \\ 1, & \text{if } i=j=k=2, \\ 0, & \text{otherwise.} \end{cases}$$

   except at one vertex v_1. On v_1, A'_{v_1} is defined similarly except that it outputs $-k'$, instead of 1, when $i = j = k = 2$. It is straightforward to verify that the contraction of T' equals $k^* - k'$.

4. Invoke the oracle for TNZ on T'. If the oracle answers YES, we output NO. Otherwise, we output YES.

To see that the above algorithm works, suppose we are given a YES instance of A, i.e. a graph G along with a number k which is greater than or equal to k^*, the true number of perfect matchings in G. Then, in step 1, there will be some nondeterministic path on which $k' = k^*$. On this path, T' constructed in step 3 will contract to zero, and therefore the oracle will output NO in step 4. As desired, we end up answering YES. On the other hand, suppose we are given a NO instance of A, i.e. a graph G along with a number k which is smaller than k^*. In this case there will be no nondeterministic path on which $k' = k^*$, and therefore T' will always contract to a nonzero value. Hence, the oracle will always output YES and we will end up answering NO, as desired. This shows that $A \in \mathbf{NP}^{\mathrm{TNZ}}$.

However, we note that an oracle Turing machine for A can easily count the number of perfect matchings on a given graph using binary search. This means that

$$\mathbf{P}^{\#\text{PERFECT-MATCHING}} \subseteq \mathbf{P}^A \subseteq \mathbf{P}^{\mathbf{NP}^{\mathrm{TNZ}}} \subseteq \mathbf{NP}^{\mathbf{NP}^{\mathrm{TNZ}}}.$$

Using our assumption that TNZ is in $\Sigma_i^{\mathbf{P}}$, we conclude

$$\mathbf{P}^{\#\text{PERFECT-MATCHING}} \subseteq \mathbf{NP}^{\mathbf{NP}^{\Sigma_i^{\mathbf{P}}}} = \Sigma_{i+2}^{\mathbf{P}}.$$

On the other hand, Toda's theorem and the #**P**-completeness of #PERFECT-MATCHING imply

$$\mathbf{PH} \subseteq \mathbf{P}^{\#\mathrm{P}} = \mathbf{P}^{\#\text{PERFECT-MATCHING}}.$$

Combining the two inclusions, we obtain $\mathbf{PH} \subseteq \Sigma_{i+2}^{\mathbf{P}}$, as desired. □

Since a widely believed conjecture is that the polynomial hierarchy should not collapse, this theorem can be interpreted as strong evidence that tensor network nonzero testing is not likely to be contained in the polynomial hierarchy.

4.4 Special cases of tensor network nonzero testing

Unfortunately, the results of the previous section can be interpreted as showing that tensor network nonzero testing is highly unlikely to be in **NP**. This implies that the most general case of tensor network nonzero testing cannot be used to provide an efficient classical witness for local Hamiltonians (as we will show in Section 4.5, TNZ $\in$ **NP** would imply that the commuting local Hamiltonian problem is also in **NP**). In this section, we identify two special cases of tensor network nonzero testing which can be placed in **NP**, in the hope that such special cases can be translated to produce relevant results in quantum Hamiltonian complexity.

4.4.1 Nonnegative tensor networks

The first special case we consider is the case of nonnegative tensor networks, which are defined as follows.

Definition 4.9. A tensor network is called **nonnegative** if its tensors only uses nonnegative real entries.

Theorem 4.3. Tensor network nonzero testing for nonnegative tensor networks is contained in **NP**.

Proof. Suppose we are given a nonnegative tensor network T. By definition, T is nonzero if and only if there exists an input $\vec{x}$ such that $T^*(\vec{x}) > 0$ (where T^* denotes the contraction of T). Furthermore, since each summand in the computation of $T^*(\vec{x})$ is nonnegative, $T^*(\vec{x}) > 0$ if and only if there exists at least one summand which is nonzero. Hence, an **NP** witness for this problem would specify a labeling $\vec{x}$ on the open edges of T and labeling $\vec{y}$ on the closed edges of T such that $T(\vec{x}, \vec{y}) > 0$. This is easy to verify in polynomial time, and implies that $T^*(\vec{x}) > 0$. Hence, tensor network nonzero testing for nonnegative tensors is contained in **NP**. □

Theorem 4.4. Tensor network nonzero testing for nonnegative tensor networks is **NP**-hard.

Proof. We prove the theorem by reducing 3-EDGE-COLORING to tensor network nonzero testing. The former problem, which asks whether a given graph is edge-colorable with 3 colors, is known to be **NP**-complete even on simple 3-regular graphs [72].

Suppose we are given an instance of 3-EDGE-COLORING, i.e. a simple 3-regular graph G. As in the proof of Theorem 4.1, we construct a nonnegative tensor network T on this graph such that its contraction encodes the number of edge-colorings of G. To achieve this, we set the bond dimension of every edge to be 3, and make each vertex $v \in V$ correspond to the tensor A defined as follows:

$$A(i,j,k) = \begin{cases} 1 & \text{if } i,\ j, \text{ and } k \text{ are distinct,} \\ 0 & \text{otherwise.} \end{cases}$$

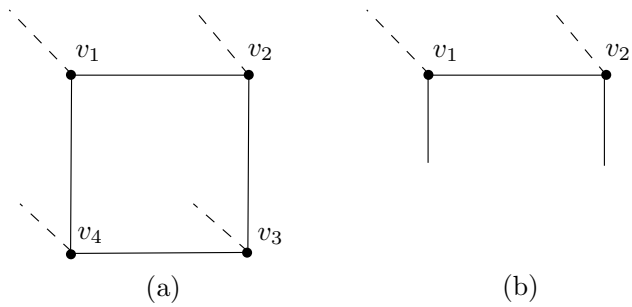

Figure 4.4: (a) An example tensor network T. (b) The subnetwork of T induced by vertices $\{v_1, v_2\}$. Note that the edges (v_1, v_4) and (v_2, v_3) are treated as open edges in the subnetwork. Namely, $E_T^{\text{phys}}(S)$ contains the dashed open edges in (b), whereas $E_T^{\text{vir}}(S)$ contains the solid open edges.

Since T is a closed tensor network without any open edge, its contraction is simply a scalar. By definition, it is computed as

$$\sum_{i_1,i_2,\ldots,i_{|E|}\in\{0,1\}} T(i_1,\ldots,i_{|E|}) = \sum_{i_1,i_2,\ldots,i_{|E|}\in\{0,1\}} \prod_{\substack{v\in V \\ i_{v_1},i_{v_2},i_{v_3}\in E(v)}} A(i_{v_1}, i_{v_2}, i_{v_3}).$$

Noting that each summand is 1 if and only if $i_1, \ldots, i_{|E|}$ corresponds to a valid edge-coloring, we see that the contraction of T indeed encodes the number of edge-colorings of the graph G. Hence, deciding whether the contraction of T is nonzero corresponds to deciding whether there exists at least one valid edge-coloring of G, which shows that tensor network nonzero testing for nonnegative tensor networks is **NP**-hard. □

Together, the above two theorems show that tensor network nonzero testing for nonnegative tensor networks is indeed **NP**-complete. The result holds even if we restrict to nonnegative tensor networks on 3-regular graphs. On the other hand, it is well known that tensor networks on 2-regular graphs are efficiently contractible, because such graphs have the same kind of 1D structure which made possible e.g. the contraction of matrix product states (see Section 2.5.2).

4.4.2 Injective tensor networks

Next, we consider the case of the so-called injective tensor networks, which are defined as follows:

Definition 4.10. Let T be a tensor network defined on graph $G = (V, E)$ and $S \subseteq V$. Also, let $E_T^{\text{phys}}(S)$ denote the open edges in T that are incident to vertices in S. The **subnetwork of T induced by** S is the tensor network T_S consisting of all vertices in S along with all

the edges that are incident to them. Edges between S and $V \setminus S$ are treated as open edges in T_S and are denoted by $E_T^{\text{vir}}(S)$.

Definition 4.11. A tensor network T defined on graph $G = (V, E)$ is called k**-injective** if V can be partitioned into sets $S_1, \ldots, S_k$ such that for every $1 \leq i \leq k$,

1. T_{S_i} is connected,
2. $E_T^{\text{phys}}(S_i)$ is nonempty, and
3. the operator A_i obtained by viewing T_{S_i} as a linear map from $E_T^{\text{vir}}(S_i)$ to $E_T^{\text{phys}}(S_i)$ is injective.

Thus, an intuitive understanding of injectivity is that the given tensor network can be partitioned into k parts such that the physical state of each part (i.e. output from $E_T^{\text{phys}}(S_i)$) is uniquely determined by its boundary conditions (i.e. input to $E_T^{\text{vir}}(S_i)$). We note that this definition of injectivity was inspired from the study of [89], in which a similar notion was studied in the context of translationally invariant tensor networks, and was observed to be a generic condition.

Interestingly, the following two theorems demonstrate a way in which injectivity might be related to the problem of tensor network nonzero testing.

Theorem 4.5. The following problem, which we call injectivity testing, is in **NP**.

- **Input**: A tensor network T.
- **Output**:
  - YES if T is k-injective for some k, with the size of each S_i being $O(\log n)$.
  - NO otherwise.

Proof. The prover simply specifies the partition $S_1, \ldots, S_k$ that witnesses the k-injectivity of T. Since each S_i is of logarithmic size, T_{S_i} can be contracted in polynomial time. Then it is trivial to check that the conditions of k-injectivity hold. □

Theorem 4.6. If a tensor network T is k-injective for some k, then its contraction T^* is nonzero.

Proof. Suppose $S_1, \ldots, S_k$ is the partition that witnesses the k-injectivity of T. Then, A_i, as defined in Definition 4.11, is injective for every i. This means that the adjoint map A_i^* from $E_T^{\text{phys}}(S_i)$ to $E_T^{\text{vir}}(S_i)$ is surjective, so for every i there exists some $|\psi_i\rangle$ such that

$$A_i^* |\psi_i\rangle = |\overbrace{00 \cdots 0}^{|E_T^{\text{vir}}(S_i)|}\rangle.$$

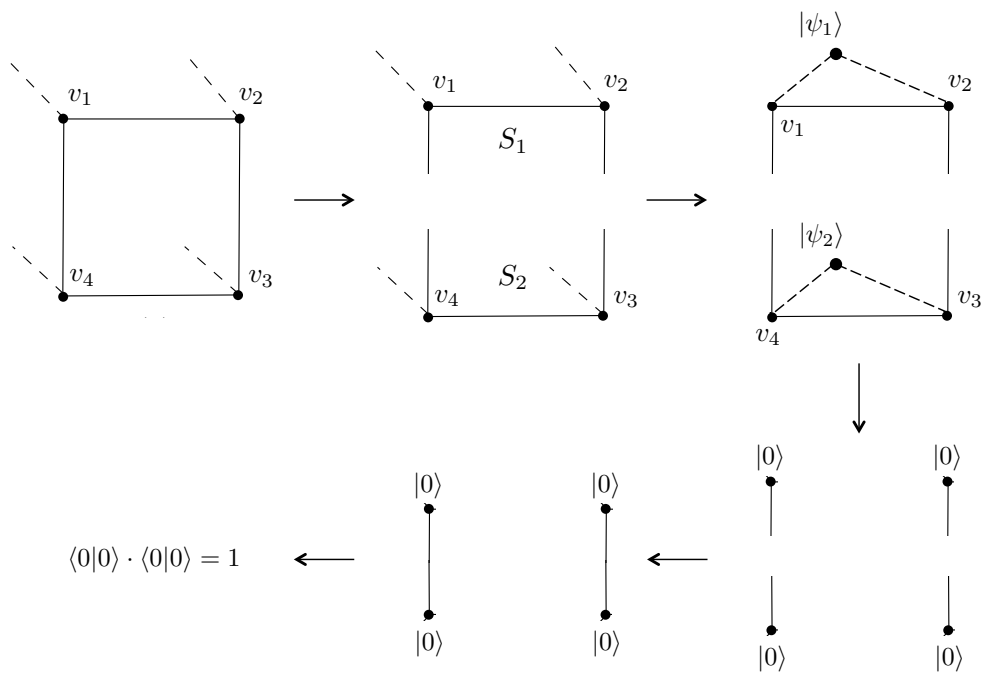

Figure 4.5: Illustrating the proof of Theorem 4.6.

Hence, if we join each T_{S_i} with a tensor network representing $|\psi_i\rangle$ so as to represent $A_i^*|\psi_i\rangle$, the contraction of the resulting tensor network must equal the tensor product state $|0\rangle \otimes \cdots \otimes |0\rangle$. Recombining them back into the shape of T, we see that the contraction of the resulting closed tensor network is a product of many copies of $\langle 0|0\rangle$, which is 1. Thus, we have shown that if A is the linear map interpretation of T from $\cup_{i=1}^k E_T^{\text{phys}}(S_i)$ to $\mathbb{C}$, $A(|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_k\rangle) = 1$ (see Figure 4.5). It follows that the contraction of T is nonzero. $\square$

Note that the above two theorems provide a sense in which injectivity testing can be regarded as a weaker version of tensor network nonzero testing. In fact, the notion of injectivity provides an efficiently verifiable certificate of the fact that a given tensor network is nonzero. A natural question therefore is whether the converse might also hold: can we show e.g. that every nonzero tensor network has an injective tensor network representation in which S_i's are logarithmically large? The question is interesting because an affirmative answer to this question could potentially lead to a novel approach to proving that the commuting local Hamiltonian problem is in **NP**.

To make progress on this question, we can try to formulate it as follows: given a nonzero tensor network T, is it always possible to find a geometrically equivalent k-injective tensor network T' such that the size of each S_i is $O(\log n)$? Here, T and T' are said to be geometrically equivalent if their underlying graphs are identical. We note that this notion of geometric equivalence is often natural in quantum Hamiltonian complexity, where tensor

networks of fixed geometry (e.g. matrix product states) are often used to represent ground states of local Hamiltonians.

In the following theorem, we answer this question in the negative.

Theorem 4.7. For all $k > 2$, there exists a nonzero tensor network T which does not have a geometrically equivalent k-injective representation.

Proof. We prove the theorem by explicitly constructing a tensor network with the desired property. Let T be a matrix product state of length n with bond dimension 2 (Figure 4.6). Then, we can make T represent the quantum state $|\psi\rangle = |00\cdots00\rangle + |10\cdots01\rangle$ by defining the tensors of T as follows:

1. If $i \in \{1, n\}$, A_{v_i} is the tensor that outputs 1 if and only if the labels on the two incident edges match. Otherwise, A_{v_i} outputs 0.

2. If $i \in \{2, \ldots, n-1\}$, A_{v_i} is the tensor that outputs 1 if and only if the labels on the two incident closed edges match and the label on the incident open edge is 0. Otherwise, A_{v_i} outputs 0.

Now, assume towards a contradiction that T admits a geometrically equivalent k-injective representation T' for some $k > 2$. Since $k > 2$, there exists some S_i that contains neither v_1 nor v_n. Moreover, since S_i must be connected, $S_i = \{v_j, v_{j+1} \ldots, v_{l-1}, v_l\}$ for some j and l. Let $L = \{v_1, \ldots, v_{j-1}\}$ and $R = \{v_{l+1}, \ldots, v_n\}$, and denote by e_L and e_R the edges (v_{j-1}, v_j) and (v_l, v_{l+1}) respectively, as in Figure 4.6. Now, if we join to each of the $j-1$ open edges in L a tensor network representing $|0\rangle$, L can be seen as a tensor network with one open edge e_L. Moreover, the vector $|\psi_L\rangle$ represented by this tensor network is nonzero, because otherwise the coefficient of any standard basis vector that begins with $j-1$ zeroes must vanish in $|\psi\rangle$, which is a contradiction. Similarly, if we join to each of the first $n-l-1$ open edges in R a tensor network representing $|0\rangle$ and to the last open edge a tensor network representing $|1\rangle$, R can be seen as a nonzero tensor network with one open edge e_R. Let $|\psi_R\rangle$ be the vector represented by this tensor network.

Finally, we observe that the operator A_i obtained by viewing T_{S_i} as a linear map from $E_T^{\text{phys}}(S_i)$ to $E_T^{\text{vir}}(S_i) = \{e_L, e_R\}$ is surjective by definition, so there must be some input $|\phi\rangle$ such that $A_i|\phi\rangle = |\psi_L\rangle \otimes |\psi_R\rangle$. This means that if we now join to the open edges of S_i a tensor network representing $|\phi\rangle$, the resulting tensor network can be viewed as an inner product between $A_i|\phi\rangle$ and $|\psi_L\rangle \otimes |\psi_R\rangle$, which is clearly nonzero. However, another way to interpret this tensor network is as an inner product between $|\psi\rangle$ and $|00\cdots0\rangle \otimes |\phi\rangle \otimes |00\cdots01\rangle$, which is zero. This is a contradiction, and we conclude that T does not have a geometrically equivalent k-injective representation.

□

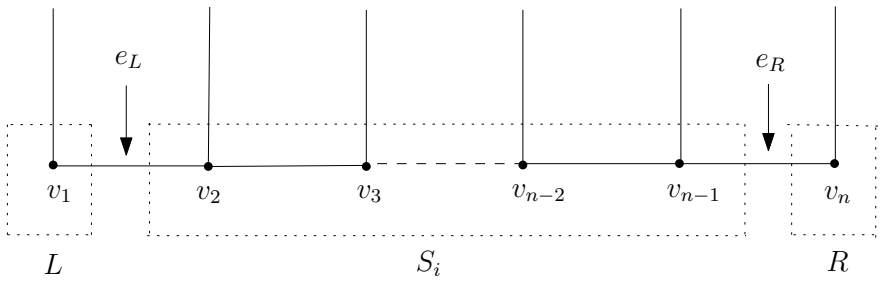

Figure 4.6: The tensor network T' in the proof of Theorem 4.7. In this example, $L = \{v_1\}$, $S_i = \{v_2, \ldots, v_{n-1}\}$, and $R = \{v_n\}$.

4.5 Connections to quantum Hamiltonian complexity

In this section, we rigorously establish the connection between tensor network nonzero testing and the commuting local Hamiltonian problem. In particular, it turns out that the nonnegative special case of tensor network nonzero testing directly corresponds to the commuting version of stoquastic quantum k-SAT, allowing us to immediately place the problem in **NP**.

Theorem 4.8. If tensor network nonzero testing is in **NP**, then the commuting k-local Hamiltonian problem is also in **NP** for any $k = O(\log n)$.

Proof. To prove this theorem, we use a setup similar to that of Schuch [96]. Suppose we are given an instance of the commuting local Hamiltonian problem, i.e., a commuting local Hamiltonian $H = \sum_i H_i$ and two numbers a and b such that $b - a > 1/\text{poly}(n)$. If H is a YES instance, its ground state $|\psi\rangle$ will have energy at most a. We seek to provide an efficiently verifiable witness to this fact.

First, we note that since H_i's commute with each other, there is a basis of H that simultaneously diagonalizes every H_i. This implies that H has a ground state $|\psi\rangle$ which is a simultaneous eigenvector of H_i's. Moreover, if we denote by Π_i the projection operator onto the eigenspace of H_i occupied by $|\psi\rangle$, $\Pi := \prod_i \Pi_i$ is the projection onto a subspace of H consisting only of ground states. In particular, $\langle\psi|\Pi|\psi\rangle = 1$, so Π is nonzero.

Now, suppose that tensor network nonzero testing is in **NP**. Then, the prover for the commuting local Hamiltonian problem specifies for each H_i which eigenspace Π_i is occupied by the ground state $|\psi\rangle$, and also provides the **NP** witness of the fact that the tensor network $\Pi = \prod_i \Pi_i$ is nonzero (since each Π_i is of polynomial size, Π can be efficiently described as a tensor network by joining Π_i's together as in Figure 4.7). The verifier does the following:

1. Check that the sum of eigenvalues corresponding to eigenspaces Π_i's is at most a.

2. Invoke the verifier for tensor network nonzero testing to check that Π is nonzero.

If H indeed has small ground energy, it is obvious that an honest prover will pass the above test. Conversely, if the prover passes the above test, Π is a nonempty subspace of H and any

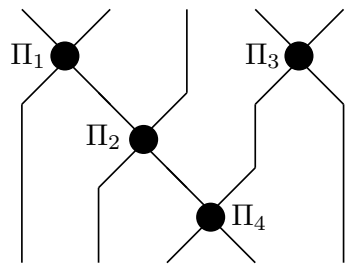

Figure 4.7: An example tensor network representation of Π.

state in Π has energy smaller than or equal to a. Hence, we conclude that the commuting k-local Hamiltonian is in **NP** for any $k = O(\log n)$. □

Theorem 4.9. If tensor network nonzero testing for nonnegative tensor networks is in **NP**, then the commuting stoquastic quantum k-SAT problem is also in **NP** for any $k = O(\log n)$.

Proof. Suppose $H = \sum_i H_i$ is a YES instance of commuting stoquastic quantum k-SAT and $|\psi\rangle$ is the ground state of H. Since $|\psi\rangle$ has zero energy and H_i's are projections, $|\psi\rangle$ must occupy the smallest eigenspace of each H_i. Moreover, since H_i has real and nonpositive off-diagonal matrix elements, the projection Π_i onto the smallest eigenspace of H_i has nonnegative real entries (Proposition 4.1 of [31]). This implies that the tensor network representation of $\Pi := \Pi_i$ is a nonnegative tensor network. The theorem now follows by the proof of Theorem 4.8. □

Corollary 4.10. The commuting stoquastic quantum k-SAT problem is in **NP** for any $k = O(\log n)$.

Proof. This is an immediate consequence of Theorem 4.3 and Theorem 4.9. □

We remark that while commutativity and stoquasticity constraints may appear to make the problem more "classical," this result is still surprising given that highly nontrivial quantum Hamiltonians such as the toric code [44] satisfy all the constraints of this problem. Stoquastic quantum k-SAT without the commutativity constraint is known to be **MA**-complete by a result of Bravyi et al. [30]

4.6 Conclusions

In this section, we have studied the computational complexity of tensor network nonzero testing, a fundamental problem in quantum Hamiltonian complexity. Despite the analogies made between this problem and the decision versions of counting problems, our results show that the problem is unlikely to be much easier than tensor network contraction: namely, it

is shown that the problem is not contained in the polynomial hierarchy unless the hierarchy collapses.

On the other hand, we were able to identify two "easy" special cases of tensor network nonzero testing – nonzero testing of nonnegative tensors and injectivity testing – which may be useful in certain contexts. For example, the fact that nonzero testing of tensor networks with nonnegative real entries is **NP**-complete has a direct implication on an important open question in quantum Hamiltonian complexity: namely, it places commuting stoquastic quantum k-SAT in **NP** for any $k \in O(\log n)$. To the best of our knowledge, this represents the first approach to commuting local Hamiltonian problems which does not rely on the techniques of Bravyi and Vyalyi [32]. Thus our result is free of any constraint on the locality parameter k, which seems to be imposed by the use of those techniques.

Naturally, our results open up a number of directions for future research. Firstly, we note that the construction used in the proof of Theorem 4.8 imposes a very specific structure on tensor network Π. Hence, in fact we only need to show that the special case of tensor network nonzero testing restricted to this structure of Π is in **NP** in order to show that the commuting k-local Hamiltonian problem is also in **NP**. This possibility is intriguing and seems to merit further research. Secondly, it also seems worthwhile to explore whether our techniques can be extended to place the commuting stoquastic k-local Hamiltonian problem in **NP**. The proof of Theorem 4.9 does not go through in this case because the local terms H_i's are not necessarily projections.

But most importantly, the framework of tensor network nonzero testing seems to have a potential to apply to other problems in quantum Hamiltonian complexity as well, beyond the commuting local Hamiltonian problems which were explored in this chapter. Tensor networks are ubiquitous in quantum Hamiltonian complexity, and the problem of nonzero testing seems too fundamental to be not important. We hope that further research on this topic will beget many more approaches to understanding high complexity quantum systems.

Chapter 5

Conclusions

For one who is interested in both computer science and fundamental sciences, the present time is full of excitement and opportunities. As scientific theories across all disciplines become more complex and required computations more demanding, we are entering an era in which in order to do any science one has to know a bit of computer science. Moreover, this is not only in the sense of knowing how to write simple programs, but also in the sense of having a more fundamental insight into what computation is. In this thesis, we have surveyed how two seemingly unrelated fields of computer science and quantum physics are coming together on the common ground of quantum computing. The emergent field of quantum Hamiltonian complexity, which lies at the heart of this intersection, makes a particularly strong case of the benefit bestowed by such interdisciplinary exchanges of expertise.

The main contribution of this thesis is two promising classical approaches to understanding high complexity quantum systems. Firstly, we have introduced the notion of a quantum Turing test, which compares the black-box behavior of a quantum device to that of a suitable classical model in order to test the "quantumness" of the machine. Our classical model for the quantum annealer shows a remarkable correlation with experimental data from the D-Wave quantum annealer and closely reproduces its performance on test instances specially designed to measure speedup. This can be formulated as the machine's failure on a quantum Turing test defined on those input distributions. Our methodology is expected to continue to provide an effective guideline to future work on experimental quantum annealing.

Secondly, we have studied the computational problem of tensor network nonzero testing, a fundamental problem which can be thought of as a decision version of the #**P**-complete tensor network contraction problem. Under reasonable complexity assumptions, we prove that the problem is not contained in the polynomial hierarchy and therefore, in its most general form, unlikely to yield an efficient classical description of properties of quantum systems. On the other hand, we identify certain special cases of tensor network nonzero testing which are contained in **NP**. In particular, the fact that nonzero testing of nonnegative tensor networks is in **NP** turns out to have a direct implication on an open problem in quantum Hamiltonian complexity: it shows that commuting stoquastic quantum k-SAT, a variant of the more general commuting local Hamiltonian problem, is in **NP**.

A chief benefit of working in an interdisciplinary field such as quantum Hamiltonian complexity is that it is full of important discoveries which are to be made just by establishing simple connections between concepts from different fields. This compels us to free ourselves from the established formalism and mannerism specific to a field and revisit familiar concepts afresh, which often leads to a more intuitive and deeper understanding of those concepts. We hope that the work presented in this thesis succeeds in demonstrating this point: in fact, our work mostly consisted in recombining in interesting ways elementary concepts from computer science and physics. Most importantly, we hope that it sends out a clear invitation to the reader to jump into the field of these interesting connections between computer science and other sciences, which exist and must be further clarified.

Bibliography

[1] Dorit Aharonov, Michael Ben-Or, and Elad Eban. "Interactive proofs for quantum computation". In: *Proceedings of Innovations of Computer Science*. 2010, pp. 453–469.

[2] Dorit Aharonov and Lior Eldar. "On the Complexity of Commuting Local Hamiltonians, and Tight Conditions for Topological Order in Such Systems". In: *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*. FOCS '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 334–343. ISBN: 978-0-7695-4571-4. DOI: 10.1109/FOCS.2011.58. URL: http://dx.doi.org/10.1109/FOCS.2011.58.

[3] Dorit Aharonov and Lior Eldar. "The Commuting Local Hamiltonian Problem on Locally Expanding Graphs is Approximable in NP". In: *Quantum Information Processing* 14.1 (Jan. 2015), pp. 83–101. ISSN: 1570-0755. DOI: 10.1007/s11128-014-0877-9. URL: http://dx.doi.org/10.1007/s11128-014-0877-9.

[4] Dorit Aharonov and Tomer Naveh. *Quantum NP - A survey*. E-print arXiv:0210077. http://arxiv.org/abs/quant-ph/0210077. 2002.

[5] Dorit Aharonov et al. "Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation". In: *SIAM J. Comput.* 37.1 (Apr. 2007), pp. 166–194. ISSN: 0097-5397. DOI: 10.1137/S0097539705447323. URL: http://dx.doi.org/10.1137/S0097539705447323.

[6] Dorit Aharonov et al. "The Power of Quantum Systems on a Line". In: *Communications in Mathematical Physics* 287.1 (2009), pp. 41–65. ISSN: 1432-0916. DOI: 10.1007/s00220-008-0710-3. URL: http://dx.doi.org/10.1007/s00220-008-0710-3.

[7] Armin Alaghi and John P. Hayes. "Survey of Stochastic Computing". In: *ACM Trans. Embed. Comput. Syst.* 12.2s (May 2013), 92:1–92:19. ISSN: 1539-9087. DOI: 10.1145/2465787.2465794. URL: http://doi.acm.org/10.1145/2465787.2465794.

[8] Tameem Albash and Daniel A. Lidar. "Decoherence in adiabatic quantum computation". In: *Phys. Rev. A* 91 (6 June 2015), p. 062320. DOI: 10.1103/PhysRevA.91.062320. URL: http://link.aps.org/doi/10.1103/PhysRevA.91.062320.

[9] Tameem Albash et al. "Consistency tests of classical and quantum models for a quantum annealer". In: *Phys. Rev. A* 91 (2015), p. 042314.

[10] Tameem Albash et al. "Reexamining classical and quantum models for the D-Wave One Processor". In: *Eur. Phys. J. Special Topics* 224 (2015), pp. 111–129.

[11] Boris Altshuler, Hari Krovi, and Jeremie Roland. "Anderson localization casts clouds over adiabatic quantum optimization". In: *Proceedings of the National Academy of Sciences of the United States of America* 107.28 (2010), pp. 12446–12450.

[12] M. H. S. Amin, C. J. S. Truncik, and D. V. Averin. "Role of single-qubit decoherence time in adiabatic quantum computation". In: *Phys. Rev. A* 80 (2 Aug. 2009), p. 022303. DOI: 10.1103/PhysRevA.80.022303. URL: http://link.aps.org/doi/10.1103/PhysRevA.80.022303.

[13] Itai Arad and Zeph Landau. "Quantum Computation and the Evaluation of Tensor Networks". In: *SIAM Journal on Computing* 39.7 (2010), pp. 3089–3121. DOI: 10.1137/080739379. eprint: http://dx.doi.org/10.1137/080739379. URL: http://dx.doi.org/10.1137/080739379.

[14] Itai Arad, Zeph Landau, and Umesh Vazirani. "Improved one-dimensional area law for frustration-free systems". In: *Phys. Rev. B* 85 (19 May 2012), p. 195145. DOI: 10.1103/PhysRevB.85.195145. URL: http://link.aps.org/doi/10.1103/PhysRevB.85.195145.

[15] Nikhil Bansal, Sergey Bravyi, and Barbara Terhal. "Classical approximation schemes for the ground-state energy of quantum and classical Ising spin Hamiltonians on planar graphs". In: *Quant. Inf. Comp.* 9.8 (2009), p. 0701.

[16] F Barahona. "On the computational complexity of Ising spin glass models". In: *Journal of Physics A: Mathematical and General* 15.10 (1982), p. 3241. URL: http://stacks.iop.org/0305-4470/15/i=10/a=028.

[17] J. A. Barker. "A quantum statistical Monte Carlo method; path integrals with boundary conditions". In: *The Journal of Chemical Physics* 70.6 (1979), pp. 2914–2918. DOI: http://dx.doi.org/10.1063/1.437829. URL: http://scitation.aip.org/content/aip/journal/jcp/70/6/10.1063/1.437829.

[18] Stefanie Barz et al. "Demonstration of Blind Quantum Computing". In: *Science* 335.6066 (2012), pp. 303–308.

[19] Stefanie Barz et al. "Experimental verification of quantum computation". In: *Nature Physics* 9 (2013), pp. 727–731.

[20] Charles H. Bennett. "The thermodynamics of computation—a review". In: *International Journal of Theoretical Physics* 21.12 (), pp. 905–940. ISSN: 1572-9575. DOI: 10.1007/BF02084158. URL: http://dx.doi.org/10.1007/BF02084158.

[21] Ethan Bernstein and Umesh Vazirani. "Quantum Complexity Theory". In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1411–1473.

[22] Jacob D. Biamonte and Peter J. Love. "Realizable Hamiltonians for universal adiabatic quantum computers". In: *Phys. Rev. A* 78 (1 July 2008), p. 012352. DOI: 10.1103/PhysRevA.78.012352. URL: http://link.aps.org/doi/10.1103/PhysRevA.78.012352.

[23] Sergio Boixo et al. "Computational multiqubit tunnelling in programmable quantum annealers". In: *Nature Communications* 7 (2016).

[24] Sergio Boixo et al. "Evidence for quantum annealing with more than one hundred qubits". In: *Nature Physics* 10 (2014), pp. 218–224.

[25] Sergio Boixo et al. "Experimental signature of programmable quantum annealing". In: *Nature Communications* 4 (2012), p. 2067.

[26] Sergio Boixo et al. *Quantum annealing with more than one hundred qubits*. E-print arXiv:1304.4595. http://arxiv.org/pdf/1304.4595v2.pdf (An updated version appears in [24]). 2013.

[27] A. Braunstein, M. Mézard, and R. Zecchina. "Survey propagation: An algorithm for satisfiability". In: *Random Structures and Algorithms* 27.2 (2005), pp. 201–226.

[28] Sergey Bravyi. *Efficient algorithm for a quantum analogue of 2-SAT*. E-print arXiv: 0602108. http://arxiv.org/abs/quant-ph/0602108. 2006.

[29] Sergey Bravyi. "Monte Carlo Simulation of Stoquastic Hamiltonians". In: *Quantum Info. Comput.* 15.13-14 (Oct. 2015), pp. 1122–1140. ISSN: 1533-7146. URL: http://dl.acm.org/citation.cfm?id=2871363.2871366.

[30] Sergey Bravyi, Arvid J. Bessen, and Barbara M. Terhal. *Merlin-Arthur Games and Stoquastic Complexity*. E-print arXiv:0611021. http://arxiv.org/abs/quant-ph/0611021. 2006.

[31] Sergey Bravyi and Barbara Terhal. "Complexity of Stoquastic Frustration-Free Hamiltonians". In: *SIAM J. Comput.* 39.4 (Nov. 2009), pp. 1462–1485. ISSN: 0097-5397. DOI: 10.1137/08072689X. URL: http://dx.doi.org/10.1137/08072689X.

[32] Sergey Bravyi and Mikhail Vyalyi. "Commutative Version of the Local Hamiltonian Problem and Common Eigenspace Problem". In: *Quantum Info. Comput.* 5.3 (May 2005), pp. 187–215. ISSN: 1533-7146. URL: http://dl.acm.org/citation.cfm?id=2011637.2011639.

[33] Sergey Bravyi et al. "The Complexity of Stoquastic Local Hamiltonian Problems". In: *Quantum Info. Comput.* 8.5 (May 2008), pp. 361–385. ISSN: 1533-7146. URL: http://dl.acm.org/citation.cfm?id=2011772.2011773.

[34] Anne Broadbent, Joseph F. Fitzsimons, and Elham Kashefi. "Universal blind quantum computation". In: *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*. 2009, pp. 517–526.

[35] H. John Caulfield and Shlomi Dolev. "Why future supercomputing requires optics". In: *Nat Photon* 4.5 (May 2010), pp. 261–263. URL: http://dx.doi.org/10.1038/nphoton.2010.94.

[36] Andrew M. Childs, Edward Farhi, and John Preskill. "Robustness of adiabatic quantum computation". In: *Phys. Rev. A* 65 (1 Dec. 2001), p. 012322. DOI: 10.1103/PhysRevA.65.012322. URL: http://link.aps.org/doi/10.1103/PhysRevA.65.012322.

[37] Fan Chung. *Spectral Graph Theory*. American Mathematical Society, 1992.

[38] B. Jack Copeland, Carl J. Posy, and Oron Shagrir. "Is Quantum Mechanics Falsifiable? A Computational Perspective on the Foundations of Quantum Mechanics". In: *Computability:Turing, Gödel, Church, and Beyond*. MIT Press, 2013, pp. 376–. ISBN: 9780262312677. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6555798.

[39] D. G. Cory et al. "Experimental Quantum Error Correction". In: *Phys. Rev. Lett.* 81 (10 Sept. 1998), pp. 2152–2155. DOI: 10.1103/PhysRevLett.81.2152. URL: http://link.aps.org/doi/10.1103/PhysRevLett.81.2152.

[40] Philip J. D. Crowley and A. G. Green. *An Anisotropic Landau-Lifschitz-Gilbert model of dissipation in qubits*. E-print arXiv:1503.00651. http://arxiv.org/pdf/1503.00651v1. 2015.

[41] Wim van Dam, Michele Mosca, and Umesh Vazirani. *How Powerful is Adiabatic Quantum Computation?* E-print arXiv:0206003. http://arxiv.org/pdf/quant-ph/0206003v1. 2002.

[42] Wim van Dam and Umesh Vazirani. *Limits on Quantum Adiabatic Optimization*. Manuscript. http://www.cs.berkeley.edu/~vazirani/pubs/qao.pdf. 2003.

[43] Vasil S. Denchev et al. *What is the Computational Value of Finite Range Tunneling?* E-print arXiv:1512.02206. http://arxiv.org/abs/1512.02206. 2014.

[44] Eric Dennis et al. "Topological quantum memory". In: *Journal of Mathematical Physics* 43 (9 2002), pp. 4452–4505. ISSN: 0022-2488.

[45] N G Dickson et al. "Thermally assisted quantum annealing of a 16-qubit problem". In: *Nature Communications* 4 (2013), p. 1903.

[46] P. A. M. Dirac. "The Lagrangian in quantum mechanics". In: *Phys. Z. der Sowjetunion* 3 (1933), pp. 64–71.

[47] William L Ditto, K Murali, and Sudeshna Sinha. "Chaos computing: ideas and implementations". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 366.1865 (2008), pp. 653–664.

[48] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *Quantum Adiabatic Evolution Algorithms versus Simulated Annealing*. E-print arXiv:quant-ph/0201031. http://arxiv.org/pdf/quant-ph/0201031v1.pdf. 2002.

[49] Edward Farhi et al. *Quantum Computation by Adiabatic Evolution*. E-print arXiv: quant-ph/0001106. http://arxiv.org/pdf/quant-ph/0001106v1.pdf. 2000.

[50] Richard P. Feynman. "A new approach to quantum theory". PhD thesis. Princeton University, 1942.

[51] Richard P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6 (1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179. URL: http://dx.doi.org/10.1007/BF02650179.

[52] Joseph F. Fitzsimons and Elham Kashefi. *Unconditionally verifiable blind computation*. E-print arXiv:1203.5217. http://arxiv.org/pdf/1203.5217v2. 2012.

[53] Austin G. Fowler et al. "Surface codes: Towards practical large-scale quantum computation". In: *Phys. Rev. A* 86 (3 Sept. 2012), p. 032324. DOI: 10.1103/PhysRevA.86.032324. URL: http://link.aps.org/doi/10.1103/PhysRevA.86.032324.

[54] Stuart J. Freedman and John F. Clauser. "Experimental Test of Local Hidden-Variable Theories". In: *Phys. Rev. Lett.* 28 (14 Apr. 1972), pp. 938–941. DOI: 10.1103/PhysRevLett.28.938. URL: http://link.aps.org/doi/10.1103/PhysRevLett.28.938.

[55] FRANK GAITAN. "SIMULATION OF QUANTUM ADIABATIC SEARCH IN THE PRESENCE OF NOISE". In: *International Journal of Quantum Information* 04.05 (2006), pp. 843–870. DOI: 10.1142/S0219749906002213. eprint: http://www.worldscientific.com/doi/pdf/10.1142/S0219749906002213. URL: http://www.worldscientific.com/doi/abs/10.1142/S0219749906002213.

[56] Sevag Gharibian et al. "Quantum Hamiltonian Complexity". In: *Foundations and Trends in Theoretical Computer Science* 10.3 (2015), pp. 159–282. ISSN: 1551-305X. DOI: 10.1561/0400000066. URL: http://dx.doi.org/10.1561/0400000066.

[57] Sevag Gharibian et al. "Tensor Network Non-zero Testing". In: *Quantum Info. Comput.* 15.9-10 (July 2015), pp. 885–889. ISSN: 1533-7146. URL: http://dl.acm.org/citation.cfm?id=2871422.2871429.

[58] Adrian Giordani. "A quantum leap in processors?" In: *Scientific Computing World* (Dec. 2015). URL: http://www.scientific-computing.com/features/feature.php?feature_id=490.

[59] Leslie Ann Goldberg et al. "A Complexity Dichotomy for Partition Functions with Mixed Signs". In: *SIAM J. Comput.* 39.7 (Aug. 2010), pp. 3336–3402. ISSN: 0097-5397. DOI: 10.1137/090757496. URL: http://dx.doi.org/10.1137/090757496.

[60] David Gosset and Daniel Nagaj. "Quantum 3-SAT Is QMA1-Complete". In: *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. FOCS '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 756–765. ISBN: 978-0-7695-5135-7. DOI: 10.1109/FOCS.2013.86. URL: http://dx.doi.org/10.1109/FOCS.2013.86.

[61] Daniel Gottesman and Sandy Irani. "The Quantum and Classical Complexity of Translationally Invariant Tiling and Hamiltonian Problems". In: *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 95–104. ISBN: 978-0-7695-3850-1. DOI: 10.1109/FOCS.2009.22. URL: http://dx.doi.org/10.1109/FOCS.2009.22.

[62] V. Granville, M. Krivanek, and J. P. Rasson. "Simulated annealing: a proof of convergence". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.6 (June 1994), pp. 652–656. ISSN: 0162-8828. DOI: 10.1109/34.295910.

[63] D.J. Griffiths. *Introduction to Quantum Mechanics*. Pearson Prentice Hall, 2005.

[64] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: ACM, 1996, pp. 212–219. ISBN: 0-89791-785-5. DOI: 10.1145/237814.237866. URL: http://doi.acm.org/10.1145/237814.237866.

[65] Sean Hallgren, Daniel Nagaj, and Sandeep Narayanaswami. "The Local Hamiltonian Problem on a Line with Eight States is QMA-complete". In: *Quantum Info. Comput.* 13.9-10 (Sept. 2013), pp. 721–750. ISSN: 1533-7146. URL: http://dl.acm.org/citation.cfm?id=2535680.2535681.

[66] T. P. Harty et al. "High-Fidelity Preparation, Gates, Memory, and Readout of a Trapped-Ion Quantum Bit". In: *Phys. Rev. Lett.* 113 (22 Nov. 2014), p. 220501. DOI: 10.1103/PhysRevLett.113.220501. URL: http://link.aps.org/doi/10.1103/PhysRevLett.113.220501.

[67] M B Hastings. "An area law for one-dimensional quantum systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2007.08 (2007), P08024. URL: http://stacks.iop.org/1742-5468/2007/i=08/a=P08024.

[68] Matthew B. Hastings. "Trivial Low Energy States for Commuting Hamiltonians, and the Quantum PCP Conjecture". In: *Quantum Info. Comput.* 13.5-6 (May 2013), pp. 393–429. ISSN: 1533-7146. URL: http://dl.acm.org/citation.cfm?id=2481614.2481617.

[69] Bettina Heim et al. "Quantum versus classical annealing of Ising spin glasses". In: *Science* 348.6231 (2015), pp. 215–217.

[70] Itay Hen et al. *Probing for quantum speedup in spin glass problems with planted solutions*. E-print arXiv:1502.01663. http://arxiv.org/pdf/1502.01663v2.pdf. 2015.

[71] A. S. Holevo. "Bounds for the Quantity of Information Transmitted by a Quantum Communication Channel". In: *Probl. Peredachi Inf.* 9.3 (1973), pp. 3–11.

[72] Ian Holyer. "The NP-Completeness of Edge-Coloring". In: *SIAM Journal on Computing* 10.4 (1981), pp. 718–720. DOI: 10.1137/0210055. eprint: http://dx.doi.org/10.1137/0210055. URL: http://dx.doi.org/10.1137/0210055.

[73] Sabine Jansen, Ruedi Seiler, and Mary-Beth Ruskai. "Bounds for the adiabatic approximation with applications to quantum computation". In: *J. Math. Phys.* 48 (2007), p. 102111.

[74] M.W. Johnson et al. "Quantum annealing with manufactured spins". In: *Nature* 473 (2011), pp. 194–198.

[75] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum annealing in the transverse Ising model". In: *Phys. Rev. E* 58 (5 Nov. 1998), pp. 5355–5363. DOI: 10.1103/PhysRevE.58.5355. URL: http://link.aps.org/doi/10.1103/PhysRevE.58.5355.

[76] Alastair Kay. "Quantum-Merlin-Arthur-complete translationally invariant Hamiltonian problem and the complexity of finding ground-state energies in physical systems". In: *Phys. Rev. A* 76 (3 Sept. 2007), p. 030307. DOI: 10.1103/PhysRevA.76.030307. URL: http://link.aps.org/doi/10.1103/PhysRevA.76.030307.

[77] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by Simulated Annealing". In: *Science* 220.4598 (1983), pp. 671–680. ISSN: 0036-8075. DOI: 10.1126/science.220.4598.671. eprint: http://science.sciencemag.org/content/220/4598/671.full.pdf. URL: http://science.sciencemag.org/content/220/4598/671.

[78] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. Boston, MA, USA: American Mathematical Society, 2002. ISBN: 0821832298.

[79] Zeph Landau, Umesh Vazirani, and Thomas Vidick. "A polynomial time algorithm for the ground state of one-dimensional gapped local Hamiltonians". In: *Nat Phys* 11.7 (July 2015), pp. 566–569. URL: http://dx.doi.org/10.1038/nphys3345.

[80] Robert McConnell et al. "Entanglement with negative Wigner function of almost 3,000 atoms heralded by one photon". In: *Nature* 519.7544 (Mar. 26, 2015), pp. 439–442. URL: http://dx.doi.org/10.1038/nature14293.

[81] Catherine C. McGeoch and Cong Wang. "Experimental Evaluation of an Adiabiatic Quantum System for Combinatorial Optimization". In: *Proceedings of the 2013 ACM Conference on Computing Frontiers*. 2013.

[82] Nicholas Metropolis et al. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. DOI: http://dx.doi.org/10.1063/1.1699114. URL: http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114.

[83] Daniel Nagaj. "Local Hamiltonians in Quantum Computation". PhD thesis. Massachusetts Institute of Technology, 2008.

[84] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. New York, NY, USA: Cambridge University Press, 2011. ISBN: 1107002176, 9781107002173.

[85] Roberto Oliveira and Barbara M. Terhal. "The Complexity of Quantum Spin Systems on a Two-dimensional Square Lattice". In: *Quantum Info. Comput.* 8.10 (Nov. 2008), pp. 900–924. ISSN: 1533-7146. URL: `http://dl.acm.org/citation.cfm?id=2016985.2016987`.

[86] Gheorghe Păun. "Applications of Membrane Computing". In: ed. by Gabriel Ciobanu, Gheorghe Păun, and Mario J. Pérez-Jiménez. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. Chap. Introduction to Membrane Computing, pp. 1–42. ISBN: 978-3-540-29937-0. DOI: `10.1007/3-540-29937-8_1`. URL: `http://dx.doi.org/10.1007/3-540-29937-8_1`.

[87] Roger Penrose. *The Emperor's New Mind: Concerning Computers, Minds and The Laws of Physics*. Oxford University Press, 1989.

[88] D. Perez-Garcia et al. "Matrix Product State Representations". In: *Quantum Info. Comput.* 7.5 (July 2007), pp. 401–430. ISSN: 1533-7146. URL: `http://dl.acm.org/citation.cfm?id=2011832.2011833`.

[89] D. Pérez-García et al. "Characterizing symmetries in a projected entangled pair state". In: *New Journal of Physics* 12 (2010), p. 025010.

[90] K.P. Pudenz, T. Albash, and Daniel Lidar. "Quantum annealing correction for random Ising problems". In: *Physical Review A* 91 (2015), p. 042302.

[91] Ben Reichardt. "The quantum adiabatic optimization algorithm and local minima". In: *ACM Symposium on Theory of Computing*. 2004.

[92] Ben W. Reichardt, Falk Unger, and Umesh Vazirani. "Classical command of quantum systems". In: *Nature* 496 (2013), pp. 456–460.

[93] Troels F. Rønnow et al. "Defining and detecting quantum speedup". In: *Science Online* (2014), p. 1252319.

[94] Rishi Saket. *A PTAS for the Classical Ising Spin Glass Problem on the Chimera Graph Structure*. E-print arXiv:1306.6943. `http://arxiv.org/pdf/1306.6943v2`. 2013.

[95] Jun John Sakurai. *Advanced Quantum Mechanics*. Addison-Wesley, 1967, pp. 1–334.

[96] Norbert Schuch. "Complexity of Commuting Hamiltonians on a Square Lattice of Qubits". In: *Quantum Info. Comput.* 11.11-12 (Nov. 2011), pp. 901–912. ISSN: 1533-7146. URL: `http://dl.acm.org/citation.cfm?id=2230956.2230957`.

[97] Norbert Schuch and Frank Verstraete. "Computational complexity of interacting electrons and fundamental limitations of density functional theory". In: *Nat Phys* 5.10 (Oct. 2009), pp. 732–735. URL: `http://dx.doi.org/10.1038/nphys1370`.

[98] Norbert Schuch et al. "Computational Complexity of Projected Entangled Pair States". In: *Phys. Rev. Lett.* 98 (14 Apr. 2007), p. 140506. DOI: 10.1103/PhysRevLett.98.140506. URL: http://link.aps.org/doi/10.1103/PhysRevLett.98.140506.

[99] Alex Selby. *D-Wave: comment on comparison with classical computers.* http://www.archduke.org/stuff/d-wave-comment-on-comparison/. 2013.

[100] Seung Woo Shin et al. *Comment on "Distinguishing Classical and Quantum Models for the D-Wave Device".* E-print arXiv:1404.6499. http://arxiv.org/pdf/1404.6499v2.pdf. 2014.

[101] Seung Woo Shin et al. *How "Quantum" is the D-Wave Machine?* E-print arXiv: 1401.7087. http://arxiv.org/pdf/1401.7087v2.pdf. 2014.

[102] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Review* 41.2 (1999), pp. 303–332. DOI: 10.1137/S0036144598347011. eprint: http://dx.doi.org/10.1137/S0036144598347011. URL: http://dx.doi.org/10.1137/S0036144598347011.

[103] Daniel R. Simon. "On the Power of Quantum Computation". In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1474–1483. ISSN: 0097-5397. DOI: 10.1137/S0097539796298637. URL: http://dx.doi.org/10.1137/S0097539796298637.

[104] John A. Smolin and Graeme Smith. *Classical Signature of Quantum Annealing.* E-print arXiv:1305.4904. http://arxiv.org/pdf/1305.4904v1.pdf. 2013.

[105] W. Richard Stark. "Amorphous computing: examples, mathematics and theory". In: *Natural Computing* 12.3 (2013), pp. 377–392.

[106] Markus Tiersch and Ralf Schützhold. "Non-Markovian decoherence in the adiabatic quantum search algorithm". In: *Phys. Rev. A* 75 (6 June 2007), p. 062313. DOI: 10.1103/PhysRevA.75.062313. URL: http://link.aps.org/doi/10.1103/PhysRevA.75.062313.

[107] Seinosuke Toda. "PP is As Hard As the Polynomial-time Hierarchy". In: *SIAM J. Comput.* 20.5 (Oct. 1991), pp. 865–877. ISSN: 0097-5397. DOI: 10.1137/0220053. URL: http://dx.doi.org/10.1137/0220053.

[108] A. M. Turing. "Computing Machinery and Intelligence". In: *Mind* LIX.236 (1950), pp. 433–460.

[109] Salil P. Vadhan. "The Complexity of Counting in Sparse, Regular, and Planar Graphs". In: *SIAM J. Comput.* 31.2 (Feb. 2002), pp. 398–427. ISSN: 0097-5397. DOI: 10.1137/S0097539797321602. URL: http://dx.doi.org/10.1137/S0097539797321602.

[110] Leslie G. Valiant. "The complexity of computing the permanent". In: *Theoretical Computer Science* 8.2 (1979), pp. 189–201.

[111] Umesh Vazirani. *edX Berkeley CS191x: Quantum Mechanics and Quantum Computation.* https://www.edx.org/course/quantum-mechanics-quantum-computation-uc-berkeleyx-cs-191x.

[112] I. de Vega, M. C. Banuls, and Perez A. "Effects of dissipation in an adiabatic quantum search algorithm". In: *New J. Phys.* 12 (2010), p. 123010.

[113] F. Verstraete and J. I. Cirac. *Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions*. E-print arXiv:cond-mat/0407066. `https://arxiv.org/abs/cond-mat/0407066`. 2004.

[114] G. Vidal. "Class of Quantum Many-Body States That Can Be Efficiently Simulated". In: *Phys. Rev. Lett.* 101 (11 Sept. 2008), p. 110501. DOI: `10.1103/PhysRevLett.101.110501`. URL: `http://link.aps.org/doi/10.1103/PhysRevLett.101.110501`.

[115] Guifré Vidal. "Efficient Classical Simulation of Slightly Entangled Quantum Computations". In: *Phys. Rev. Lett.* 91 (14 Oct. 2003), p. 147902.

[116] Walter Vinci et al. *Distinguishing Classical and Quantum Models for the D-Wave Device*. E-print arXiv:1403.4228. `http://arxiv.org/pdf/1403.4228v1.pdf`. 2014.

[117] Lei Wang et al. *Comment on: "Classical signature of quantum annealing"*. E-print arXiv:1305.5837. `http://arxiv.org/pdf/1305.5837v1.pdf`. 2013.

[118] Steven R. White. "Density matrix formulation for quantum renormalization groups". In: *Phys. Rev. Lett.* 69 (19 Nov. 1992), pp. 2863–2866. DOI: `10.1103/PhysRevLett.69.2863`. URL: `http://link.aps.org/doi/10.1103/PhysRevLett.69.2863`.

[119] Erik Winfree. "Algorithmic Self-Assembly of DNA". PhD thesis. California Institute of Technology, 1998.

[120] Jijiang Yan and Dave Bacon. *The k-local Pauli Commuting Hamiltonians Problem is in P*. E-print arXiv:1203.3906. `http://arxiv.org/abs/1203.3906`. 2012.

[121] Kevin C. Young, Mohan Sarovar, and Robin Blume-Kohout. "Error Suppression and Error Correction in Adiabatic Quantum Computation: Techniques and Challenges". In: *Phys. Rev. X* 3 (4 Nov. 2013), p. 041013. DOI: `10.1103/PhysRevX.3.041013`. URL: `http://link.aps.org/doi/10.1103/PhysRevX.3.041013`.

[122] Stathis Zachos and Martin Furer. "Probabilistic Quantifiers vs. Distrustful Adversaries". In: *Proc. Of the Seventh Conference on Foundations of Software Technology and Theoretical Computer Science*. Pune, India: Springer-Verlag, 1987, pp. 443–455. ISBN: 0-387-18625-5. URL: `http://dl.acm.org/citation.cfm?id=36624.36655`.