Notes on Adiabatic Quantum Computing and Optimization

Aditya Morolia

December 29, 2021

Abstract

This is a report written for an Independent Study on "Adiabatic Quantum Computing and Optimization" I took under Prof. Shantanav Chakraborty in Monsoon 2021 at IIIT Hyderabad. The main references read are Albash and Lidar [2], Denchev et al. [7] and some of the papers and articles cited around them. I also study a couple of classical huristic optimization algorithms like Parallel Tempering, Tabu search, QMC, etc. and some experimental attempts to find quantum speedups on a quantum annealer.

Contents

1 Introduction

The most familiar model of quantum computation in terms of a universal set of gates was proposed by Deutsch and Penrose [8] following the ideas of Paul Benioff and Richard Feynman for a quantum Turing machine (Benioff [3], Benioff [4], Feynman [10]), and for simulating physical systems which are quantum in nature using a 'quantum' machine, to circumvent the problem with the exponential size of the Hilbert space. A programmable computer is a system based on some laws of physics, the initial state of which is programmable by us to represent the problem, into the evolution of which (the system) we can encode our solution procedure (an algorithm), and reading out the

final state of the system (measurement) gives us the solution to the problem. Quantum computing is based on the idea of a computational problem being encoded as a quantum system, evolving with a sequence of quantum gates (Unitaries, in accordance with the axiom of quantum mechanics), similar to that encoded as a classical system evolving with a sequence of classical logic gates.

Adiabatic quantum computing is another such framework where the initial state of the system is a Hamiltonian whose ground state is easy to prepare, which evolves according to the adiabatic theorem to a final Hamiltonian whose ground state encodes the solution to a computational problem. The adiabatic theorem guarantees that the system will remain in the ground state of the instantaneous hamiltonian, given that the evolution takes place sufficiently slowly. The idea to encode the solution to a computational problem into the ground state of a quantum Hamiltonian appeared in 1998 by Brooke et al. [5], in trying to solve classical combinatorial optimization problem. This was called Quantum Annealing. This was introduced as a classical 'quantum inspired' algorithm, akin to Simulated Annealing, and made use of simulated quantum fluctuations and tunneling (similar to thermal fluctuations simulated by SA.)

In 2007 in a paper titled "Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation," Aharonov et al. [1] showed that Adiabatic quantum computation with non-stoquastic Hamiltonians is polynomially equivalent to the circuit model ('Standard Quantum Computation'.)

Quantum annealing was proposed as one of the first adiabatic quantum algorithm, to solve combinatorial optimization problems. These problems can be notoriously hard, due to the vastness of the search space. The original proposal was a 'simulated quantum annealing,' in which the evolution of a quantum system was simulated on a classical computer, such that the final state contained the solution to a computational problem. [13] and Farhi et al. [9] showed that there is an advantage in using algorithms like these. It was thought that is large, controlled, programmable quantum systems could be built (quantum computers,) then such advantages would continue, ushering the era of quantum supremacy. (More on this in section 3.)

Companies like DWave have built physical devices, and considerable effort has gone into testing them. Although small speedups for curated problems have been seen, a conclusive picture does not exist.

2 Adiabatic Quantum Computing

A computation in this model is specified by two Hamiltonians H_i and H_f, such that the ground state of H_i is easy to prepare, and the ground state of H_f represents the solution to our computational problem. The Hamiltonians have to be *local*, i.e., they should only involve interactions between a constant number of particles. This requirement is equivalent, in the circuit model, to only allowing gates acting on a constant number of qubits at a time. The running time of the adiabatic computation is determined by the minimal spectral gap of all the Hamiltonians connecting H_i and H_f, given by

$$H(s) = (1-s)H_i + sH_f \tag{1}$$

where $s(t) : [0, t_f] \to [0, 1]$ is called the schedule, and t_f is the annealing time.

The following definition of adiabatic quantum computing has been taken from Aharonov et al. [1].

Definition 2.1 (k-local Hamiltonian)**.** A Hamiltonian H acting on n particles is called k-local H can be written as $\sum_A H_A$, where A runs over all the subsets of k out of n particles, and H_A acts trivially on all but the particles in A.

That is, for each A, H_A is a tensor product of a Hamiltonian on A, with identity on all the other particles. Notice that for any constant k, a k-local Hamiltonian on n qubits can be described in $2^{2k}n^k = poly(n)$ space, whereas describing an arbitrary Hamiltonian requires roughly 2^{2n} space.

Definition 2.2 (Adiabatic Quantum Computation)**.** A k-local adiabatic computation $AC(n, d, H_i, H_f, \epsilon)$ is specified by two k-local Hamiltonians, H_i and H_f acting on n d-dimensional particles, such that both Hamiltonians have unique ground states. The ground state of H_i is a tensor product state. The output is a state that is ϵ-close in l_2-norm to the ground state of H_f. Let t_f be the smallest time such that the final state of an adiabatic evolution according to Equation 1 for time t_f is ϵ-close in l_2-norm to the ground state of H_f. The running time of the adiabatic algorithm is defined to be

$$\text{cost} := t_f \cdot \max_s H(s) \tag{2}$$

Notice that the definition of running time as defined in Equation 2 is invariant under the scaling of the Hamiltonians by some overall factors.

Theorem 2.3 (Adiabatic Theorem)**.** Let H_i and H_f be two Hamiltonians acting on a n-qubit quantum systems, and consider the time dependent Hamiltonian as described in Equation 1. Assume that for all s, $H(s)$ has a unique ground state. Then for any fixed $\delta > 0$, if

$$t_f \geq \Omega\left(\frac{||H_f - H_i||^{1+\delta}}{\epsilon^{\delta} \min_{s\in[0,1]}\{\Delta^{2+\delta}(H(s))\}}\right) \tag{3}$$

then the final state of an adiabatic evolution according to $H(s)$ for time t_f is ϵ-close in the l_2-norm to the ground state of H_f.

The matrix norm is the spectral norm. $\Delta(H(s))$ is the spectral gap, defined as the minimum eigenvalue gap between the ground state and the first excited state of the instantaneous Hamiltonian.

Using the above definitions, Aharonov et al. [1] show the following theorem.

Theorem 2.4 (Equivalence of Adiabatic and Circuit Model Quantum Computation)**.** Given a quantum circuit on n qubits with L two qubit gates implementing a unitary U, and an $\epsilon > 0$, there exists a 5-local adiabatic quantum computation $AC(n + L, 2, H_i, H_f, \epsilon)$, whose running time is $poly(L, \frac{1}{\epsilon})$ and whose output after tracing out some ancilla qubits is ϵ-close in the trace distance to $U|0^{\otimes n}\rangle$. Moreover, H_i and H_f can be computed by a PTTM.

The runtime for this algorithm is $\mathcal{O}(\epsilon^{-(5+3\delta)}L^{5+2\delta})$ for any fixed $\delta > 0$.

Note that an adiabatic quantum algorithm can always be converted to a quantum circuit that can approximate the adiabatic computation by discretizing the time dependent Hamiltonian for some finite sequence of time steps and then using a standard Hamiltonian simulation algorithm to simulate the evolution.

3 Quantum Supremacy in the Adiabatic Setting and Computational Efficiency of Quantum Annealing

The term 'quantum supremacy' was coined by John Preskill in [18], where they described "the era of quantum supremacy [as] when we will be able to perform tasks with controlled quantum systems going beyond what can be achieved with ordinary digital computers." This was motivated by the unproven belief that classical systems (*ordinary digital computers*) can't simulate highly entangled quantum systems efficiently, but a programmable quantum system, by its nature, might be able to perform such tasks, surpassing what is possible with classical computation. They go on to discuss the possibilities of constructing such a controlled quantum device, the possibility of error correction, and what may or may not be possible with such a device, considering the vastness of the Hilbert space, and the fact that some states are more easy to prepare than others.

Another beautiful piece on quantum supremacy is Harrow and Montanaro [11], where they discuss various approaches to defining and demonstrating a quantum speedup from complexity theoretic point of view. They point out that quantum supremacy proposals are often based around sampling problems rather than decision problems, where the computational task is to output samples from a desired distribution, rather than outputting a boolean response (promise problems.) The strength of these proposals is that despite working with a restricted model of quantum computation, such as boson sampling, random circuit sampling, adiabatic optimization, etc., they do not need to assume that this specific model is hard to simulate classically. Rather, these can be based on well formed complexity theoretic assumptions, such as the "non-collapse of the polynomial hierarchy," or that "the exact counting of exponentially large sets is harder than their approximate counting (PostBPP $\neq$ PostBQP)." Here 'Post' indicates the respective complexity class with post-selection. They go on to discuss many other such, as well as some fine-grained complexity assumptions, and the verification of quantum supremacy experiments.

Once we have quantum computers, it becomes necessary to benchmark and compare the performances of these devices with other similar devices, as well as with various classical devices and algorithms. One common way in which this is done in the gate model of quantum computing is by comparing what is known as query complexity to the input, which is the number of times the circuit queries the input oracle. This makes the familiar theoretical study of complexity possible because we can define asymptotic behaviour of our quantum algorithms in this model, in terms of the size of input.

But this is not possible in cases where the hardware is specialized, analogue quantum computer capable of running specific algorithms as continuous time evolution of a state with some Hamiltonian, which is exactly what Quantum Annealers are. One solution to this issue is to compare time taken for the device to reach the solution or a target solution in terms of physical time (the target solution bit is important in some cases where we don't know the actual minina and can't be sure that we have indeed found the correct solution without exploring the whole (exponential) search space.) Another method is to compare the energy requirements of these devices to reach the same target (which I think is more reasonable that directly measuring time.)

3.1 Types of Quantum Speedups

Rønnow et al. [19] thoroughly studied this and used their measures to study the D-Wave two device. They define five different kinds of quantum speedups that might be possible.

1. *Provable quantum speedup*, where there exists a proof that no classical algorithm exists that can outperform a given quantum algorithm. An example of this is Grover's search algorithm, which is proven to have a quadratic speedup over the best possible classical algorithm.

2. *Strong quantum speedup*, where we compare a quantum algorithm against the best *possible* classical algorithm, which may or may not be explicitly known. This aims to capture the computational limitation of classical computers.

3. *Quantum speedup (without any adjectives)*, which compares a quantum algorithm against the best known (available) classical algorithm. This is a less ambitious version of *Strong quantum speedup*. An example of this is Shor's algorithm. The notion of 'best possible' is time and community dependent.

4. *Potential quantum speedup*, where we compare a quantum algorithm to a specific or a set of classical algorithms. An example of this is to simulate the time evolution of a quantum system, where the propagation of the wave function on a quantum computer would be exponentially faster than a direct integration of Schrodinger's equation on a classical computer.

5. *Limited quantum speedup*, where we compare the quantum algorithm to its equivalent classical algorithm. Such quantum algorithms follow the same algorithmic procedure as their classical counterpart, but on a classical hardware. This definition allows for the existence of other classical algorithms that are already better than the quantum algorithm. This is a weaker notion, useful when we have small machines capable of performing some specific computational tasks, which is the case for presently available quantum annealers, capable of performing come quantum adiabatic computing tasks. Most literature on comparing these machines to classical ones follow this definition of a speedup.

3.2 Weak Strong Clustering

Denchev et al. [7] crafted an Ising minimization problem for the DWave 2X qubit topology (the chimera topology. Refer to section 6 for more details of the hardware.) The problem is designed to have tall and thin barriers in the energy landscape, a feature that could put quantum tunneling to a huge advantage when compared to classical search techniques.

The problem description is as follows. A problem instance consists of a bunch of so called 'weak strong cluster pairs' connected together. Each weak strong cluster pair consists of two chimera unit, one of which (right) has a fixed local field of $h_2 = -1$, and all couplings (J) are set to 1. This makes the qubits ferromagnetically coupled. The local field on the left cluster is weaker and serves as a bifurcation parameter. Its values are $h_1 \in [0, 0.44]$.

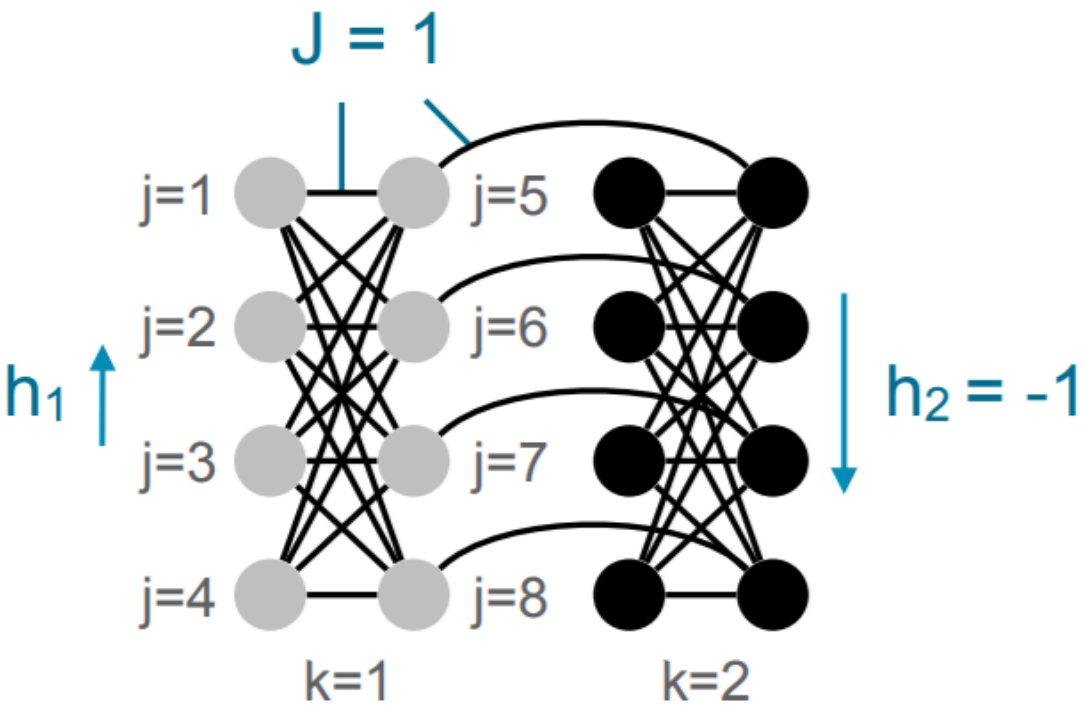

Figure 1: A weak strong cluster pair. (From Denchev et al. [7])

Using the weak strong cluster pairs as building blocks, larger problems are formed by connecting the strong clusters to one another in a glassy fashion. The four connections between two neighboring strong clusters are all set either to 1 (ferromagnetic) or -1 (antiferromagnetic), at random.

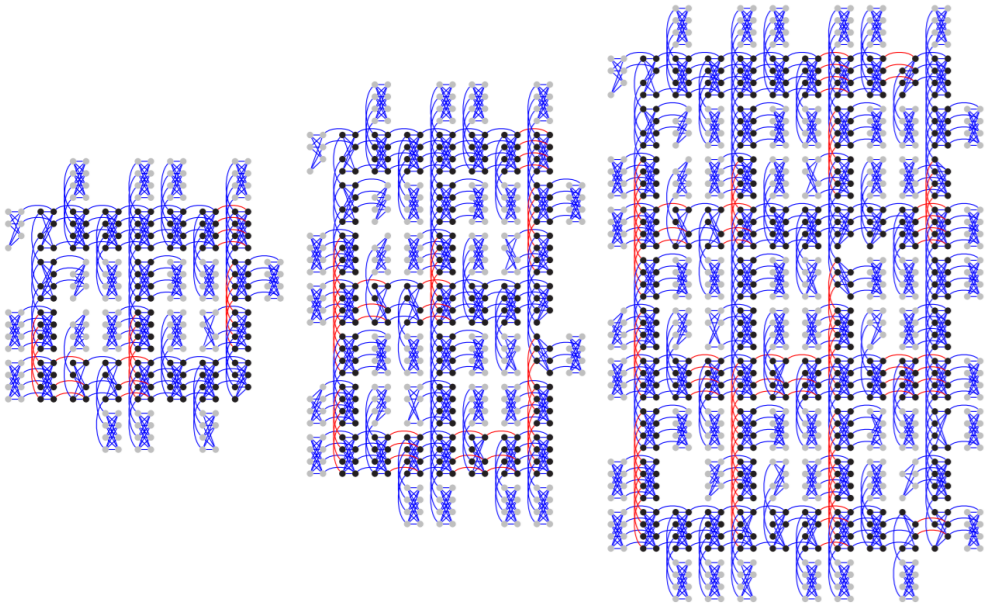

Figure 2: From Denchev et al. [7]. Layout of several instances of the weak strong cluster network problem on the DWave 2X processor. Shown are three different sizes with 296, 489 and 945 qubits. Each cluster consists of an eight qubit unit cell of the Chimera graph. Black dots depict qubits subject to a strong local field $h = -1$ while the gray dots represent qubits with the weak field $h = 0.44$. Blue lines correspond to strong ferromagnetic couplings ($J = 1$) and red lines to strong antiferromagnetic couplings ($J = -1$). Note that the graphs are somewhat irregular due to the fact that not all 1152 qubits are operational.

They ran the problem on the actual device, and compared the performance to SA and Quantum Monte Carlo. They saw a $\sim 10^8$ factor quantum speedup compared to both SA and QMC for 945 variable instances. The authors remark that although there are other classical heuristic algorithms that are comparable in performance to QA on these devices, they should be significantly outperformed by quantum annealers as the number of qubits in those devices increases and their quality improves.

4 Adiabatic Quantum Algorithms

4.1 Adiabatic Grover's Search

The adiabatic version of Grover's search exists with a provable quantum speedup. Assuming we know what Grover's search is in the circuit model and the problem is solves, following is a brief overview of adiabatic Grover's search and a proof of a quadratic quantum speedup. For proofs, see Albash and Lidar [2] section III-A.

For an n qubit system, The final Hamiltonian is defined as

$$H_1 := \mathbb{I} - |m\rangle \langle m| \tag{4}$$

where $|m\rangle$ is the market state corresponding to the marked item. Here the corresponding states are labelled as the binary representation of the index of the items. Node that H_1 is n-local. The initial Hamiltonian is defined as

$$H_0 := \mathbb{I} - |\phi\rangle \langle \phi| \tag{5}$$

where $|\phi\rangle$ is the uniform superposition state defined as

$$|\phi\rangle := \frac{1}{N} \sum_{i=0}^{N-1} |i\rangle = |+\rangle^{\otimes n} \tag{6}$$

Then, the time dependent Hamiltonian is given by

$$H(s) := (1 - A(s))H_0 + A(s)H_1 \tag{7}$$

where $s = \frac{t}{t_f} \in [0, 1]$ is the normalized time and $A(s)$ is an appropriate schedule that can be tweaked to get a reasonable adiabatic path that shows a quadratic speedup.

If the linear schedule $A(s) = s$ is considered, this adiabatic algorithm scales the same way as the classical algorithm, with minimum spectral gap

$$\Delta_{\min} = 2^{-n/2} \tag{8}$$

at $s = \frac{1}{2}$, and $t_f \gg \frac{3}{\Delta^2_{\min}}$.

Because the minimum gap only occurs at $s = \frac{1}{2}$, we can cleverly choose a schedule that slows down the evolution only around that minimum spectral gap point, which being faster otherwise. This is an interesting observation and becomes very important in AQC, because we can recover different complexities by altering the schedule appropriately. This makes the choice and study of schedules very important.

In case of adiabatic Grover's, quadratic speedup is recovered by solving the following adiabatic condition

$$t_f \gg 2 \max_s \frac{||\partial_s H(s)||}{\Delta^2(s)} + \int_0^1 \left(\frac{||\partial_s^2 H(s)||}{\Delta^2} + \frac{||\partial_s H(s)||^2}{\Delta^3} \right) ds \tag{9}$$

where H and Δ are now schedule dependent. Solving for the schedule using an appropriate ansatz, one can show that the following schedule recovers a quadratic speedup.

$$A(s) = \frac{1}{2} + \frac{1}{2\sqrt{N-1}} \tan\left((2s-1) \tan^{-1} \sqrt{N-1} \right) \tag{10}$$

It can be shown that one can not optimize the schedule further and recover a speedup greater than a quadratic one. This algorithm can also be extended to account for multiple marked states by just replacing the one marked element $|m\rangle$ with the sum $\sum_{m \in \mathcal{M}} |m\rangle \langle m|$, where $\mathcal{M}$ is the index set of the marked items.

4.2 Adiabatic Quantum Linear Systems Solver

Lin and Tong [15] gave an adiabatic quantum algorithm to solve linear systems. It allows us to prepare a target eigenstate of a given Hamiltonian if we have an initial state with a non-trivial overlap with the target eigenstate and there is a reasonable lower bound on the spectral gap. Later they discretize the algorithm for $d-$sparse matrices using a Hamiltonial simulation subroutine. They also review a couple of other papers that they follow with near optimal query complexity in the circuit model.

Given oracle access to the entries of a matrix $A \in \mathbb{C}^{N \times N}$, and a quantum state $|b\rangle \in \mathbb{C}^N$, the problem asks to prepare the quantum state $|x\rangle = \frac{A^{-1}|b\rangle}{||A^{-1}|b\rangle||}$, where $|x\rangle$ is the quantum state proportional to the solution of the linear system $Ax = b$. This is called the quantum linear systems problem.

(Note: the notation of a quantum state belonging to the vector space over $\mathbb{C}$ is a bit loose. Oracle access to a quantum state means that there is a unitary U_b that can prepare the required quantum state as $U_b |0\rangle = |b\rangle$.) They also assume that the singular values of A are contained in $[\frac{1}{\kappa}, 1]$, where κ is the condition number of A. Following is a brief description of their algorithm.

Define

$$Q_b = \mathbb{I} - |b\rangle \langle b| \tag{11}$$

Then the initial Hamiltonian is

$$H_0 := \begin{pmatrix} 0 & Q_b \\ Q_b & 0 \end{pmatrix} = \sigma_x \otimes Q_b. \tag{12}$$

The final Hamiltonian is

$$H_1 := \begin{pmatrix} 0 & AQ_b \\ AQ_b & 0 \end{pmatrix} = |0\rangle \langle 1| \otimes AQ_b + |1\rangle \langle 0| \otimes A_b A. \tag{13}$$

Note that the null space of H_1 is spanned by $|0\rangle|x\rangle$ and $|1\rangle|b\rangle$. The rest of the spectrum is separated from 0 by a gap of $\frac{1}{\kappa}$. Then the adiabatic evolution is given by

$$H(f) = (1-f)H_0 + fH_1. \tag{14}$$

Using the vanilla (linear) schedule $f(s) = s$ gives a time complexity of $\mathcal{O}(\kappa^2/\epsilon)$. Using a method called AQC(p), where the schedule is

$$f(s) = \frac{\kappa}{\kappa - 1}\left(1 - \left(1 + s(\kappa^{p-1} - 1)\right)^{\frac{1}{1-p}}\right), \quad p \in (1, 2) \tag{15}$$

the time complexity can be reduced to $\mathcal{O}(\kappa/\epsilon)$, which is optimal in κ but not in ϵ. Using a method called AQC(exp), the the time complexity can be further reduced to $\mathcal{O}\left(\kappa \log^2(\kappa) \log^4\left(\frac{\log \kappa}{\epsilon}\right)\right)$, which is near-optimal in both κ and ϵ, although with a high constant overhead (numerically shown in their paper.)

They also describe eigenstate filtering, which they use after preparing the initial state using adiabatic evolution as described above, to filter out the required state and obtain the solution to the QLS.

5 Combinatorial Optimization

Optimization problems are concerned with the minimization or maximization of some objective function subject to some constraints, or a decision that no such solution exists. Combinatorics is the mathematics of discretely structured problems. Combinatorial optimization is an optimization that deals with discrete variables.

Definition 5.1. Combinatorial optimization is the process of searching for the maxima (or the minima) of an objective function F, whose domain is a discrete (or can be reduced to a discrete set), and often consists of a large configuration space (as opposed to an N-dimensional continuous space).

The space of possible solutions is typically too large to search exhaustively using pure brute force. In some cases, problems can be solved exactly using Branch and Bound techniques. Otherwise, in many such problems specialized algorithms (sometimes based on heuristics) that quickly rule out large parts of the search space or approximation algorithms must be resorted to instead.

Some common examples of such problems include:

- **Travelling Salesman Problem:** Given the coordinates of N different cities, find the shortest possible path that a salesman can take through all of them so as to visit each city exactly once.
- **Bin Packing:** Given a set of N objects each of a specified size s_i, fit them all into as few bins as possible (of a given fixed size B).

Other examples are integer linear programming and boolean satisfiability. Most of these problems (the decision versions of them) are NP-hard. Some classical algorithms used to solve them are Random restart hill-climbing, simulated annealing, genetic algorithms and Tabu search.

For the purpose of studying quantum adiabatic optimization, we are mostly interested in weighted max-2-SAT, the Ising minimization problem and QUBO. These three problems are equivalent, and the instances can be converted from one to another. Wikipedia tells more about this.

5.1 Weighted MAX-SAT

MAX-SAT extends the SAT problem to the problem of finding an assignment that maximizes the number of satisfied clauses. Weighted MAX-SAT extends this further by adding weights to each clause and requiring a solution that maximizes the sum of weights of the satisfied clause. The MAX-SAT problem is an instance of weighted MAX-SAT where all weights are 1. These problems can either be solved by incomplete local search methods or complete branch and bound methods.

These problems are NP-Hard and APX-Complete, with a PTAS implying P = NP.

5.2 QUBO

QUBO stands for Quadratic Unconstrained Binary Optimization. It refers to the problem of minimizing the following objective function. Let Q be a $n \times n$ matrix, $x \in \{0,1\}^n$.

$$\min y = x^T Q x = \sum_{i,j} Q_{i,j} x_i x_j. \tag{16}$$

Here x is the binary decision variable. This problem is also NP-Hard.

5.3 Ising Minimization Problem

Most of this section is from the Stat Mech II course web page of Chang [6].

The Ising model is a mathematical model of an infinite lattice where each site can be in one of two states (spin up/down, or $\pm\frac{1}{2}$). The Hamiltonian of the model includes two terms. h, the external field term, can split the energies of the spin-down and spin-up state. The magnitude of h represents how strong the field is, so it tells you how much higher in energy one spin is than the other. The sign of h indicates which of the two spins is preferred. J is the interaction term. The sign of J tells us if the neighbouring spins want to align or anti-align. The magnitude of J tells us by how much do they want to align or anti align. In the Ising model, only the nearest neighbour sites interact with each other. This model can be of arbitrary dimensions. In a d-dimentional Ising model, every spin has $2d$ nearest neighbours. Thus, Hamiltonian of the Ising model is

$$H = -\sum_{\langle i,j \rangle} J_{i,j} \sigma_i^z \sigma_j^z - \sum_j h_j \sigma_j^z. \tag{17}$$

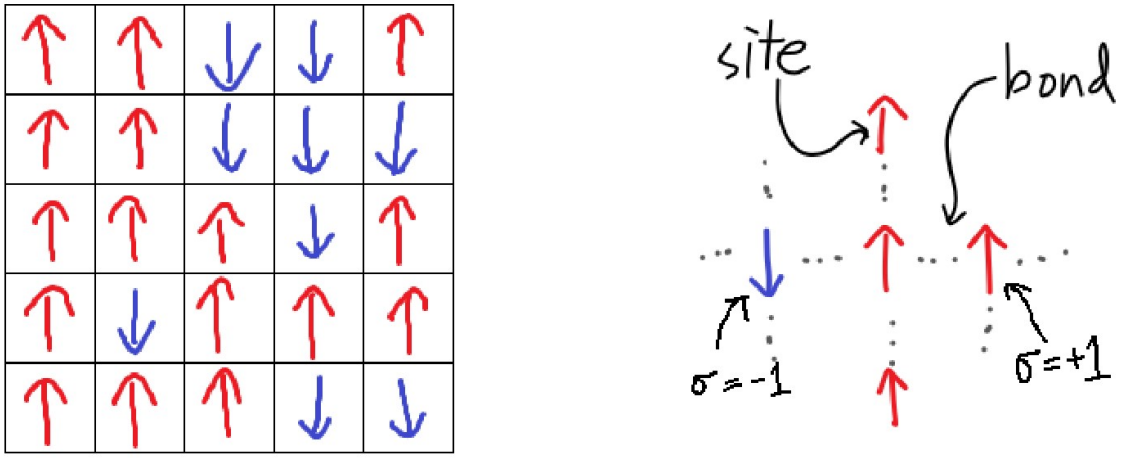

Figure 3: 2D Ising Model Image Source

The Ising minimization problem refers to the following problem.

Definition 5.2. Given the description of an Ising model as a Hamiltonian, find the spin configuration corresponding to the minimum energy of the system.

This problem is NP-hard. Sometimes a random instance of this is also called a spin-glass, referring to 'glassy' (*random*) nature of the molecules of glass.

An instance of this problem can easily be converted to a QUBO instance, by replacing the spins as $s_i = 2x_i - 1$.

A lot of NP-Hard problems can be mapped on to Ising minimization problem. Lucas [16] gives an overview of this.

5.4 Optimization Algorithms

5.4.1 Simulated Annealing

Introduced in Kirkpatrick et al. [14], Simulated Annealing is probably the most widely used heuristic algorithm for global optimization of pseudo-boolean functions with little known structure. (A pseudo-boolean function is a map $f : \{0, 1\} \to \mathbb{R}$.) The general problem of such class is given by

$$H(\vec{s}) = -\sum_{k=1}^{K} \sum_{j_1 \dots j_k}^{N} J_{j_1 \dots j_k} s_{j_1} \dots s_{j_k}, \tag{18}$$

where N is the number of spins (problem size), $s_{j_i} \in \{\pm 1\}$ are the spin variables, and each $J_{j_1 \dots j_k}$ is a real scalar called the coupling strength or coupling. $\vec{s}$ is the spin vector. $H(\vec{s})$ can thought of as a Hamiltonian of a K-spin system.

SA is a Monte Carlo algorithm, which was inspired by "a deep and useful connection between statistical mechanics (the behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature) and multivariate or combinatorial optimization (finding the minimum of a given function depending on many parameters)" (from Kirkpatrick et al. [14].) It is designed to mimic the thermalization dynamics of a system in contact with a slowly cooling reservoir. When the temperature of the system is high, the thermal fluctuations take charge and the system can pick up enough energy for its configuration to come out of a local minima. As it decreases, the system moves towards lower and lower energy states around its initial state. 'Annealing' is a 7000 year old neolithic era technology of slowly cooling metals to change its properties to desired ones. The Simulated Annealing algorithm uses the objective function of an optimization problem, which is to be minimized, instead of the energy of a material.

The algorithm works as follows. The system starts with some random configuration at a temperature T, and enters a loop. In each iteration, from the initial energy configuration E_1, the system makes a change in the spin configuration (called a *move*) to a configuration of energy E_2. If the move leads to a lower energy configuration ($E_2 < E_1$), the new configuration is accepted with probability 1. If it leads to a higher energy configuration ($E_2 > E_1$), it is accepted with some probability less than one. This probability decreases exponentially with the *badness* of the move, which is quantified to be proportional to the difference between the two energy levels, $\Delta E = E_2 - E_1$.

$$\text{Probability}(E_1 \to E_2) \ = 1 - \exp\left\{ \frac{\Delta E}{kT} \right\} \tag{19}$$

In a typical SA optimization, T starts from a high value and is gradually decreased according to a 'schedule'. At higher values of T, moves to higher energy configurations are more likely. As T tends to zero, they become more and more unlikely. The parameter k is a constant that relates temperature to energy (in nature it is Boltzmann's constant.) Algorithm 1 gives the pseudo code for SA.

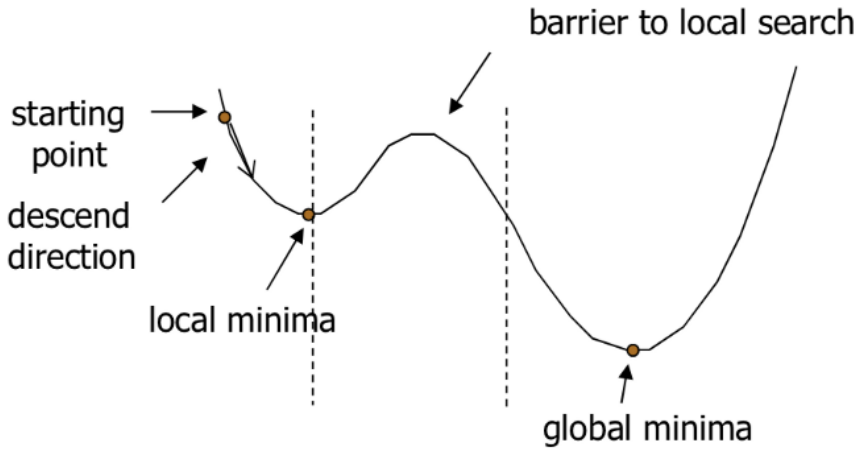

Figure 4: Landscape of a problem for Simulated Annealing. Image Source

The really cool part about this algorithm is that it can be used as a general framework to construct many variations, optimized towards a specific application. Isakov et al. [12] give highly optimized SA implementation for certain types of Ising minimization problems. Their source code is available, and table 2 of their paper summarizes the various optimizations in each implementation.

Algorithm 1 Simulated Annealing

Inputs: The cost function $f(x)$ to be minimized; A function $g(s)$ to generate random configurations from, given an arbitrary seed s; Initial temperature T_i
Outputs: The point of minima of the function x_m
Procedure:

$r \leftarrow random()$
$x_i \leftarrow g(r)$
$idx \leftarrow 0$ ▷ Sweeps Counter
while Stopping criteria not satisfied **do** ▷ This could be number of sweeps, a target energy, or the global minima
    $r \leftarrow random()$
    $x_j \leftarrow g(r)$
    if $f(x_j) < f(x_i)$ **then**
        $x_i \leftarrow x_j$
    else
        if Equation 19 is satisfied **then** ▷ Metropolis Criteria
            $x_i \leftarrow x_j$
        end if
    end if
    $idx \leftarrow idx + 1$ ▷ Update Sweeps Counter
    $T \leftarrow temperature(idx)$ ▷ Update Temperature Appropriately
end while
return x_i

5.4.2 Quantum Annealing

Quantum Annealing was proposed as a heuristic quantum enhanced optimization technique. The inspiration was that quantum mechanics allows 'tunneling' out of a local energy barrier, and therefore the system can go to a nearby

lower energy configuration even if it does not have enough energy to climb the energy barrier.

Quantum Annealing is an adiabatic quantum computation, where the initial Hamiltonian is the equal superposition of all possible spin configurations, obtained in the computational basis with pauli σ_x. The final Hamiltonian is of the form Equation 18, with the spin variables s_i replaced by the pauli σ_j^z, to account for the change to quantum Ising model. The time dependent Hamiltonian is given by

$$H(t) = -A(t)\sum_{j=1}^{N}\sigma_j^x + B(t)H_f; \quad t \in [0, t_f], \tag{20}$$

where H_f is the Ising hamiltonian defined in Equation 17, $H_f = -\sum_{\langle i,j\rangle} J_{i,j}\sigma_i^z\sigma_j^z - \sum_j h_j\sigma_j^z$ and t_f is the annealing time. Algorithm 2 gives the pseudo code for QA.

Considerable scientific and industrial effort has gone into showing a 'quantum advantage' of QA over classical algorithms. Denchev et al. [7] shows that for a curated problem called the 'weak-strong clustering pairs', QA can outperform SA by a factor of $\sim 10^8$ for a 945 variable instance. But in some other cases, no advantage, or a disadvantage has been demonstrated. A conclusive result remains elusive.

Figure 5 (taken from Denchev et al. [7]) summarizes the physics of QA very well.

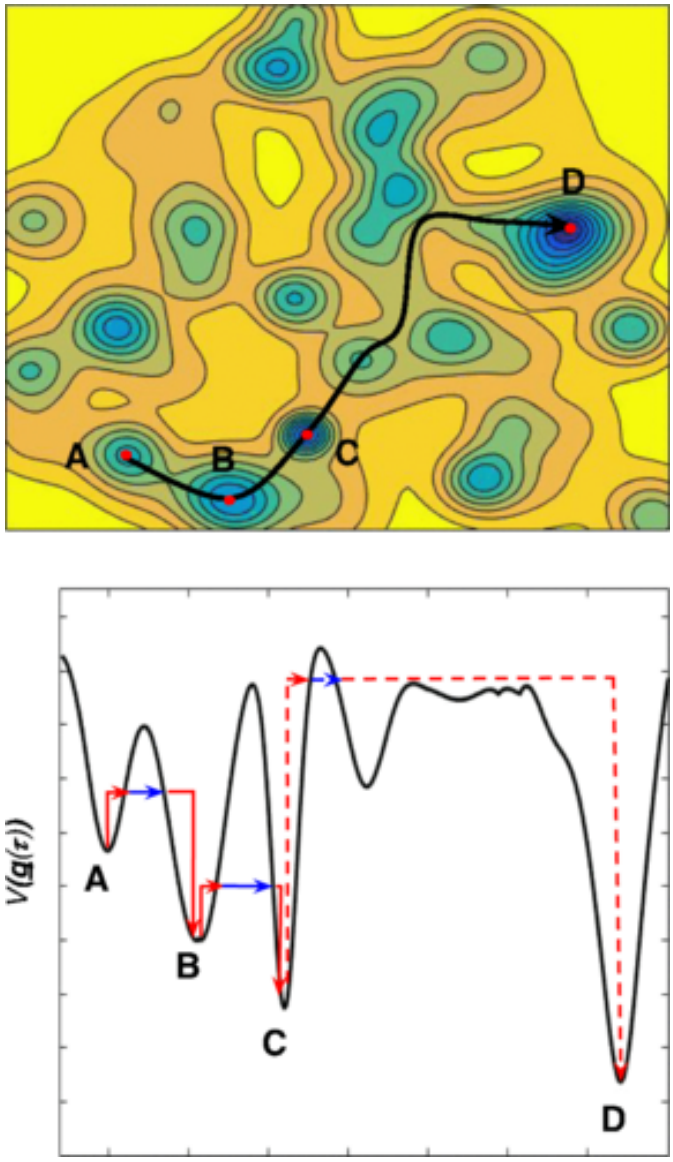

Figure 5: Top: Quantum annealer dynamics are dominated by paths through the mean-field energy landscape that have the highest transition probabilities. Shown is a path that connects local minimum A to local minimum D. Bottom: Mean-field energy $V(q(\tau))$ along the path from A to D, as defined by Eqs. (12) and (16). Finite-range, thermally assisted tunneling can be thought of as a transition consisting of three steps: (i) the system picks up thermal energy from the bath (red arrow up), (ii) the system performs a tunneling transition between the entry and exit points (blue arrow), and (iii) the system relaxes to a local minimum by dissipating energy back into the thermal bath (red arrow down). In transitions $A \to B$ or $B \to C$, finite-range tunneling considerably reduces the thermal activation energy needed to overcome the barrier. For long distance transitions in the lower part of the energy spectrum, such as $C \to D$, the transition rate is still dominated by the thermal activation energy, and the increase in transition rate brought about by tunneling is negligible.

5.4.3 Quantum Monte Carlo

5.4.4 Parallel Tempering

Marinari and Parisi [17] introduced parallel tempering, also called replica exchange monte carlo markov chain sampling. It is a variant of SA, where instead of running a single copy of the system's evolution, multiple *replicas*

Algorithm 2 Quantum Annealing

1: **Inputs:** H_i, H_f, t_f ▷ Initial and Final Hamiltonian, Annealing time
2: **Outputs:** $\vec{s}$, an n-dimensional vector containing boolean entries
3: **Procedure:**

4: Prepare the state $|\psi_i\rangle$, the ground state of H_i.
5: Fix a schedule $s(t) : [0, t_f] \to [0, 1]$ ▷ Could be taken as an input
6: $H(s) \leftarrow (1-s)H_i + sH_f$
7: On a quantum annealer, adiabatically evolve $|\psi_i\rangle$ to the final state $|\psi_f\rangle$ according to the time dependent Hamiltonian $H(s)$. ▷ Theorem 2.3 guarantees that the final state will be in the ground state of H_f if the evolution (schedule) is slow enough.
8: Measure $|\psi_f\rangle$ in the computational basis and return the output as $\vec{s}$.

that are randomly initialized at different temperatures are run in parallel. Then, on the basis of a Metropolis criterion, configurations at different temperatures are exchanged. This allows configurations at high temperatures to be accessible to systems at lower temperatures, and thus systems previously stuck in a local minima are bumped out to a higher energy configuration. This method works well for optimization problems with rugged energy landscapes. Parallel tempering is shown to work much better for such situations.

In a sense, this brings SA closer to QA, as the system does not have to gain enough eneegy to get out of a local minima. Although this comes with severe computational overhead.

5.4.5 Tabu Search

Tabu search is another meta-heuristic used to solve optimization problems. It combines local search strategies with memory structures that are used to prohibit previously visited configuration points, and drive the search in a more desirable direction. Although the algorithm also allows non-improving moves in cases where the search is stuck in a valley or local minima.

Three types of memory structures used in Tabu search, short term, intermediate term, and long term. Short term memory keeps track of recently visited configurations and can also be used to return to a specific configuration to localize and intensify search, a technique known as intensification. Sometimes intensification can be used with an intermediate term memory. The long term memory is used to drive the search to new directions, and explore previously unexplored areas of the search space.

The general procedure followed for this procedure is:

1. Start with an arbitrary candidate solution.
2. Generate a neighbourhood set of the candidate solution.
3. Remove the Tabu solutions from the neighbourhood set.
4. Choose the best solution in the neighbourhood set and update the best solution.
5. Update the Tabu states and remove the expired states (states past the Tabu tenure) from it.
6. Repeat from step 2 until the stopping criteria is met.

Simulated annealing can be thought of as a special case of Tabu search, where the Tabu is determined by the metropolis criteria (the new solution is accepted based on a probability distribution.

6 Experimental Implementations of Quantum Annealers

DWave builds specialized quantum hardware that is used to solve optimization problems of the type Equation 17. The programmer specifies the h_i and $J_{i,j}$ values, and the annealing time (typically $\in [5, 2000]\mu s$ due to machine constraints.) Other parameters such as the schedule can also be controlled.

The DWave quantum processing unit (QPU) is a lattice of interconnected qubits. The qubits they use are superconducting qubits, where the ground states of a superconducting loop form the $|0\rangle$ and $|1\rangle$ states. The qubits

are connected together via couplers in specific patterns. The building block of these devices are specific qubit connectivity graphs (qubit topologies) joined together to form bigger devices. The initial devices had a chimera topology, where the eight qubits formed a complete $K_{4,4}$ graph connected together to form bigger structures. The buildin block is shown in Figure 8. The connectivity is shown in Figure 7.

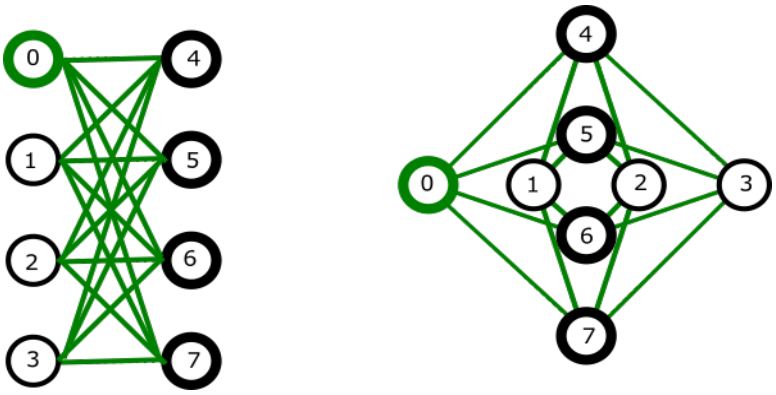

Figure 6: A chimera unit cell

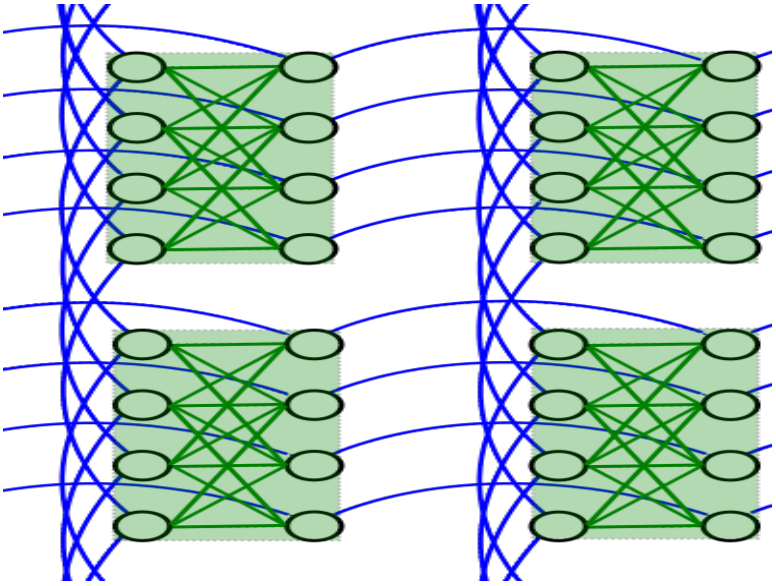

Figure 7: Chimera network.

They have recuntly moved to the pegasus structure, which has a unit cell of P_4 graph. More on these topologies can be found here. Most experiments on quantum speedups for quantum annealing, and to study physics of them, including all of the ones described above has been done on these devices.

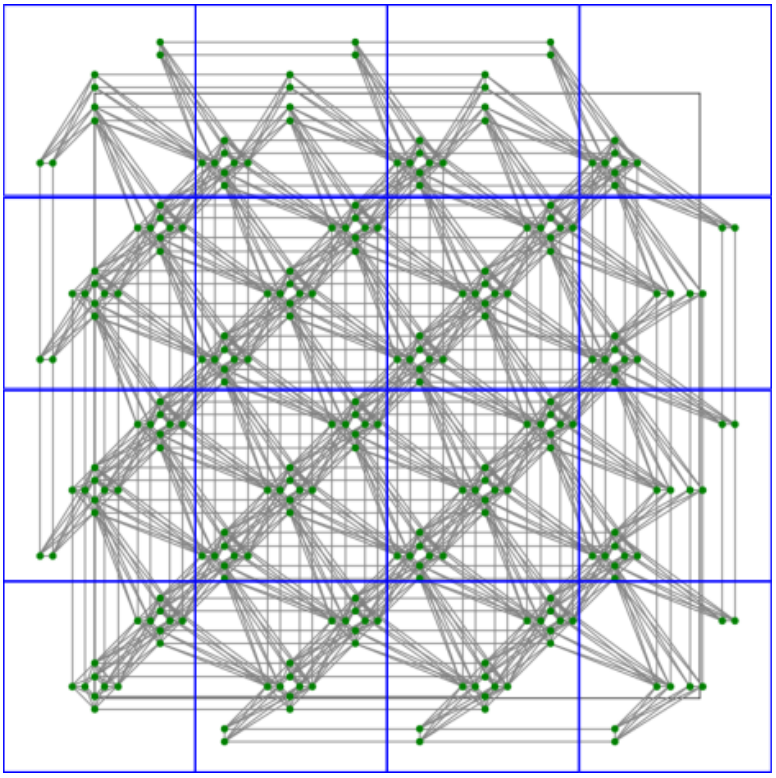

Figure 8: A pegasus unit cell network

It should be noted that these annealers don't solve the exact problem specified in the problem Hamiltonian. A DWave machine can only represent a discrete subset numbers between $[-1, 1]$ using some finite precision (the value of the local fields h_i and the couplers J_{ij} as in Equation 17). Let

$$\Gamma = \{-1, -1 + \gamma^{-1}, -1 + 2\gamma^{-1}, \ldots, -\gamma^{-1}, 0, \gamma^{-1}, 2\gamma^{-1}, \ldots, 1 - \gamma^{-1}, 1\} \tag{21}$$

for some integer $\gamma \geq 1$ (for DWave $\gamma \sim 30$). Whenever a minimization problem is sent to a DWave machine, it normalizes the matrix entries and converts each entry of the matrix to the closest number in this range. That is the problem which is then solved by the annealer.

References

[1] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Journal on Computing*, 37(1):166–194, 2007. doi: 10.1137/S0097539705447323. URL `https://doi.org/10.1137/S0097539705447323`.

[2] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90:015002, Jan 2018. doi: 10.1103/RevModPhys.90.015002. URL `https://link.aps.org/doi/10.1103/RevModPhys.90.015002`.

[3] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):563–591, May 1980. ISSN 1572-9613. doi: 10.1007/BF01011339. URL `https://doi.org/10.1007/BF01011339`.

[4] Paul Benioff. Quantum mechanical models of turing machines that dissipate no energy. *Phys. Rev. Lett.*, 48:1581–1585, Jun 1982. doi: 10.1103/PhysRevLett.48.1581. URL `https://link.aps.org/doi/10.1103/PhysRevLett.48.1581`.

[5] J. Brooke, D. Bitko, T. F., null Rosenbaum, and G. Aeppli. Quantum annealing of a disordered magnet. *Science*, 284(5415):779–781, 1999. doi: 10.1126/science.284.5415.779. URL `https://www.science.org/doi/abs/10.1126/science.284.5415.779`.

[6] Jeffrey Chang. The ising model. https://web.stanford.edu/~jeffjar/statmech/intro4.html.

[7] Vasil S. Denchev, Sergio Boixo, Sergei V. Isakov, Nan Ding, Ryan Babbush, Vadim Smelyanskiy, John Martinis, and Hartmut Neven. What is the computational value of finite-range tunneling? *Phys. Rev. X*, 6:031015, Aug 2016. doi: 10.1103/PhysRevX.6.031015. URL https://link.aps.org/doi/10.1103/PhysRevX.6.031015.

[8] David Elieser Deutsch and Roger Penrose. Quantum computational networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 425(1868):73–90, 1989. doi: 10.1098/rspa.1989.0099. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1989.0099.

[9] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292 (5516):472–475, 2001. doi: 10.1126/science.1057726. URL https://www.science.org/doi/abs/10.1126/science.1057726.

[10] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6): 467–488, Jun 1982. ISSN 1572-9575. doi: 10.1007/BF02650179. URL https://doi.org/10.1007/BF02650179.

[11] Aram W. Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, Sep 2017. ISSN 1476-4687. doi: 10.1038/nature23458. URL http://dx.doi.org/10.1038/nature23458.

[12] S.V. Isakov, I.N. Zintchenko, T.F. Rønnow, and M. Troyer. Optimised simulated annealing for ising spin glasses. *Computer Physics Communications*, 192:265–271, Jul 2015. ISSN 0010-4655. doi: 10.1016/j.cpc.2015.02.015. URL http://dx.doi.org/10.1016/j.cpc.2015.02.015.

[13] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Phys. Rev. E*, 58:5355–5363, Nov 1998. doi: 10.1103/PhysRevE.58.5355. URL https://link.aps.org/doi/10.1103/PhysRevE.58.5355.

[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi: 10.1126/science.220.4598.671. URL https://www.science.org/doi/abs/10.1126/science.220.4598.671.

[15] Lin Lin and Yu Tong. Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum*, 4:361, November 2020. ISSN 2521-327X. doi: 10.22331/q-2020-11-11-361. URL https://doi.org/10.22331/q-2020-11-11-361.

[16] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2, 2014. ISSN 2296-424X. doi: 10.3389/fphy.2014.00005. URL http://dx.doi.org/10.3389/fphy.2014.00005.

[17] E Marinari and G Parisi. Simulated tempering: A new monte carlo scheme. *Europhysics Letters (EPL)*, 19(6): 451–458, jul 1992. doi: 10.1209/0295-5075/19/6/002. URL https://doi.org/10.1209/0295-5075/19/6/002.

[18] John Preskill. Quantum computing and the entanglement frontier, 2012. URL https://arxiv.org/abs/1203.5813v3.

[19] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer. Defining and detecting quantum speedup. *Science*, 345(6195):420–424, 2014. doi: 10.1126/science.1252319. URL https://www.science.org/doi/abs/10.1126/science.1252319.